Link Discovery Tutorial

Introduction

Mohamed Ahmed Sherif[1], Irini Fundulaki[2], Axel-Cyrille Ngonga Ngomo[1]

(1) Institute for Applied Informatics, Germany
(2) FORTH, Greece

May 28th, 2017
Portoroz, Slovenia.

Disclaimer

- No pursuit of completeness [Nen+15]
- Focus on
  - Basic ideas and principles
  - Principles
  - Evaluation
  - Open questions and challenges

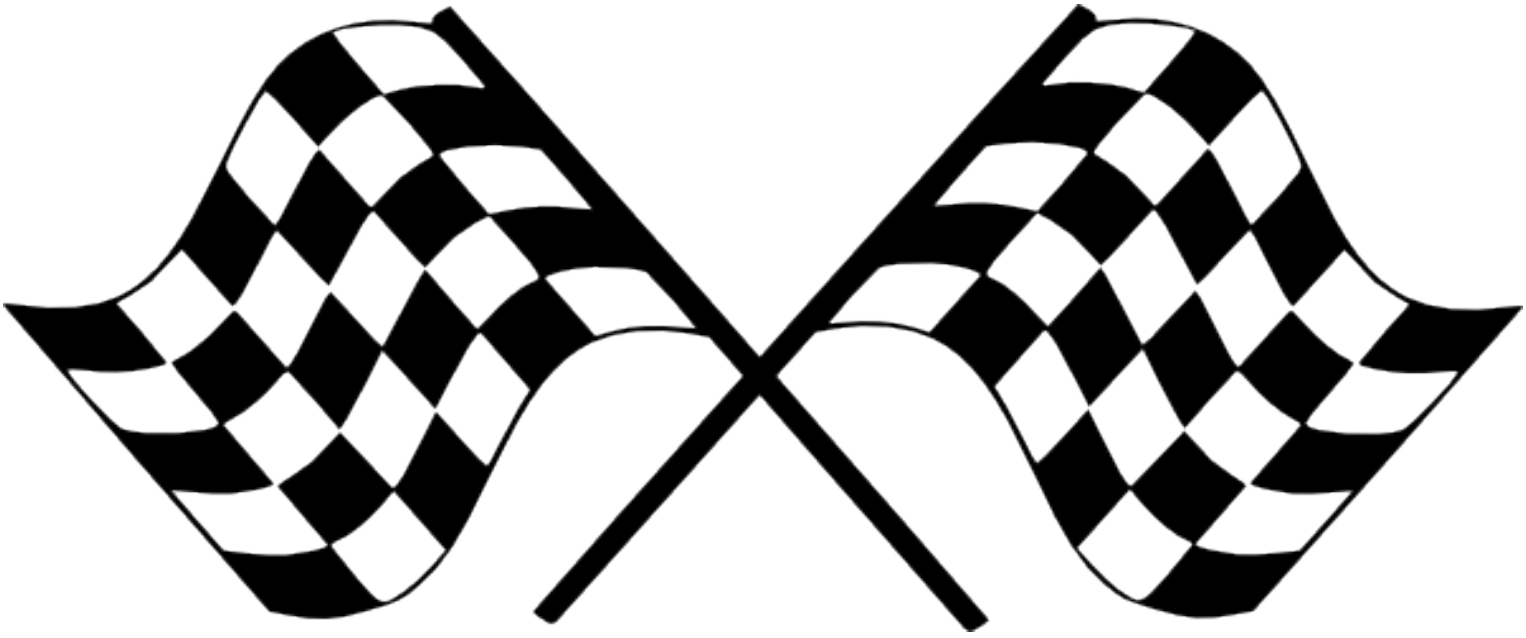
Disclaimer

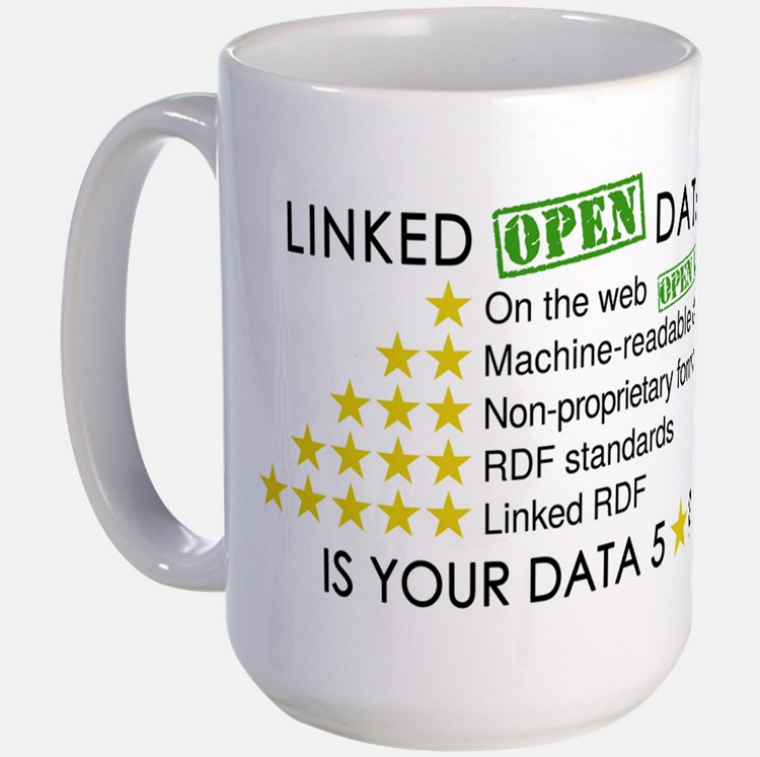
- No pursuit of completeness [Nen+15]
- Focus on
  - Basic ideas and principles
  - Principles
  - Evaluation
  - Open questions and challenges

Guidelines

- Question? Just ask!
- Comment? Go for it!
- Be kind :)

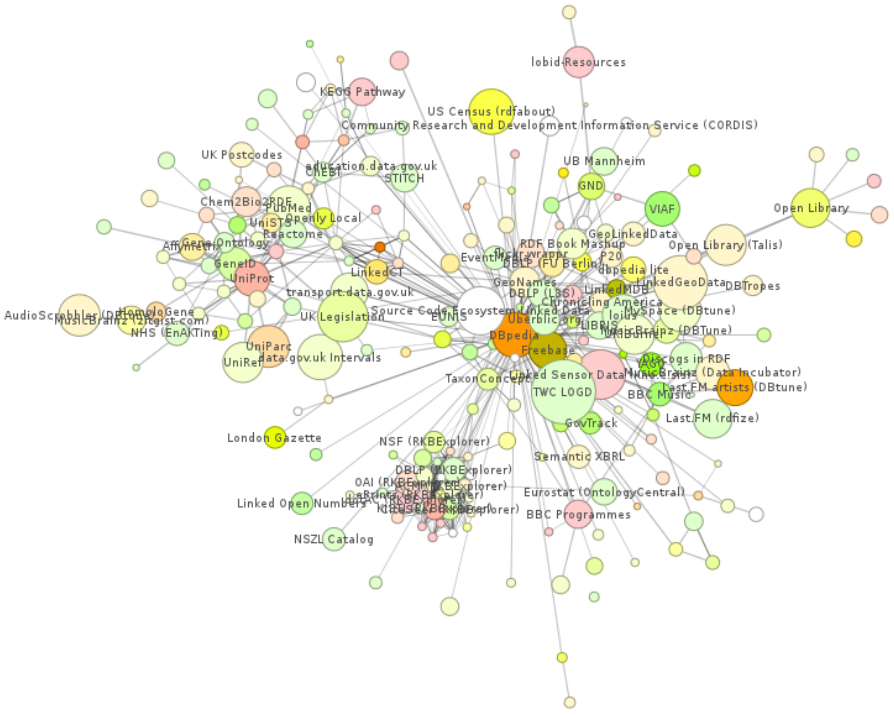
Let's Go!




LINKED OPEN DA
On the web
Machine-readable
Non-proprietary for
RDF standards
Linked RDF
IS YOUR DATA 5

Why Link Discovery?

1. Linked Open Data Cloud
   - 130+ billion triples
   - ≈ 0.5 billion links
   - Mostly `owl:sameAs`
2. Decentralized dataset creation
3. Complex information needs ⇒ Need to consume data across knowledge bases
4. Links are central for
   - Cross-ontology QA
   - Data Integration
   - Reasoning
   - Federated Queries
   - ...



Example

Give me the name and description of all drugs that cure their side-effect. [SNA13]

1. Need information from
   - Drugbank (Drug description)
   - Sider (Side-effects)
   - DBpedia (Description)
2. Gathering information via SPARQL query using links

Example

Give me the name and description of all drugs that cure their side-effect.

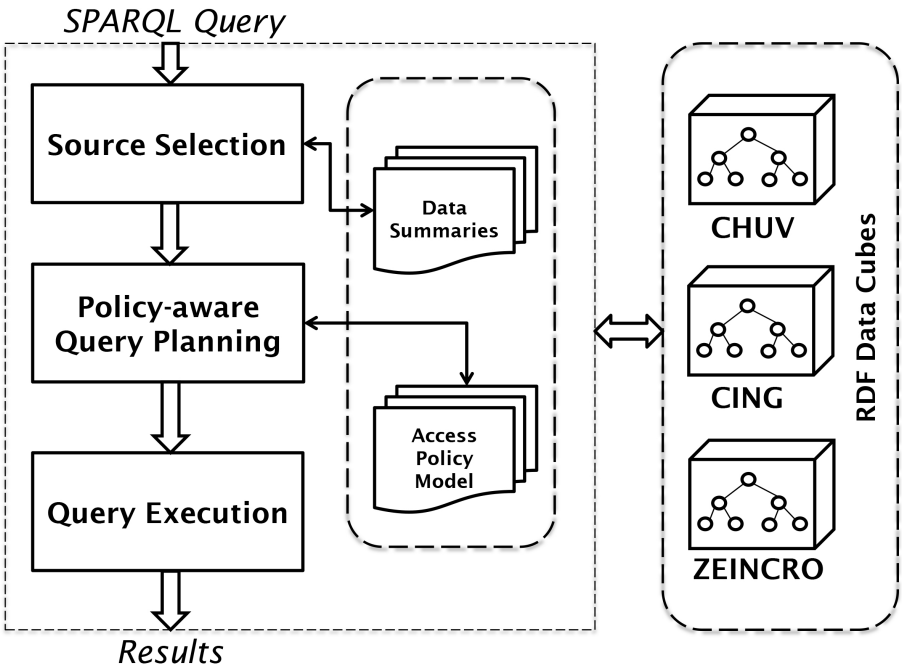
```
SELECT ?drug ?name ?desc WHERE
{
  ?drug a drugbank:Drug .
  ?drug rdfs:label ?name .
  ?drug drugbank:cures ?disease .
  ?drug owl:sameAs ?drug2 .
  ?drug owl:sameAs ?drug3 .
  ?drug2 sider:hasSideEffect ?effect .
  ?effect owl:sameAs ?disease .
  ?drug3 dbo:hasWikiPage ?desc .
}
```

Example (DEQA)

Give me flats near kindergartens in Kobe. [Leh+12]

```
SELECT ?flat WHERE
{
  ?flat a deqa:Flat .
  ?flat deqa:near ?school .
  ?school a lgdo:School .
  ?school lgdo:city lgdo:Kobe .
}
```

- Federated Queries on Patient Data [Kha+14]



Example (FedBench CD2)

Return Barack Obama's party membership and news pages. [Sal+15]

```
SELECT ?party ?page WHERE
{
  dbr:Barack_Obama dbo:party ?party .
  ?x nytimes:topicPage ?page .
  ?x owl:sameAs dbr:Barack_Obama .
}
```

Definition (Link Discovery, informal)

- Given two knowledge bases S and T, find links of type $\mathcal{R}$ between S and T
- Here, declarative link discovery

Definition

HOBBIT AKSW

Definition (Link Discovery, informal)

- Given two knowledge bases S and T, find links of type $\mathcal{R}$ between S and T
- Here, declarative link discovery

Definition (Declarative Link Discovery, formal, similarities)

- Given sets S and T of resources and relation $\mathcal{R}$
- Find $M = \{(s,t) \in S \times T : \mathcal{R}(s,t)\}$
- Common approach: Find $M' = \{(s,t) \in S \times T : \sigma(s,t) \geq \theta\}$

Definition

Definition (Link Discovery, informal)

- Given two knowledge bases S and T, find links of type $\mathcal{R}$ between S and T
- Here, declarative link discovery

Definition (Declarative Link Discovery, formal, similarities)

- Given sets S and T of resources and relation $\mathcal{R}$
- Find $M = \{(s,t) \in S \times T : \mathcal{R}(s,t)\}$
- Common approach: Find $M' = \{(s,t) \in S \times T : \sigma(s,t) \geq \theta\}$

Definition (Declarative Link Discovery, formal, distances)

- Given sets S and T of resources and relation $\mathcal{R}$
- Find $M = \{(s,t) \in S \times T : \mathcal{R}(s,t)\}$
- Common approach: Find $M' = \{(s,t) \in S \times T : \delta(s,t) \leq \tau\}$

- Most common: $\mathcal{R} =$ `owl:sameAs`
- Also known as deduplication [Nen+15]

Definition

- Goal: Address all possible relations $\mathcal{R}$
- Declarative Link Discovery: Similarity/distance defined using property values (incl. property chains)

- Goal: Address all possible relations $\mathcal{R}$
- Declarative Link Discovery: Similarity/distance defined using property values (incl. property chains)
- Example: $\mathcal{R} =$ `:sameModel`

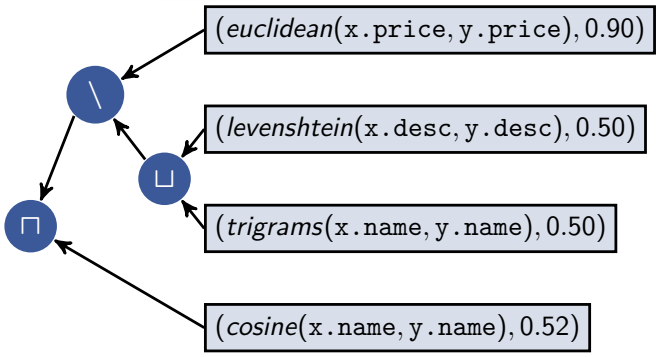
```
:s770fm rdfs:label "S770FM"@en
:s770fm rdf:type :SABER
:s770fm :model :770
:s770fm :top :FlamedMaple
:s770fm :producer :Ibanez
```

```
:s770fm rdfs:label "S770BEM"@en
:s770fm rdf:type :SABER
:s770fm :model :770
:s770fm :top :BirdEyeMaple
:s770fm :producer :Ibanez
```

1. **Time complexity**
   - Large number of triples (e.g., LinkedTCGA with 20.4 billion triples [Sal+14])
   - Quadratic a-priori runtime
   - 69 days for mapping cities from DBpedia to Geonames
   - Solutions usually in-memory (insufficient heap space)

1. **Time complexity**
   - Large number of triples (e.g., LinkedTCGA with 20.4 billion triples [Sal+14])
   - Quadratic a-priori runtime
   - 69 days for mapping cities from DBpedia to Geonames
   - Solutions usually in-memory (insufficient heap space)

2. **Accuracy**
   - Combination of several attributes required for high precision
   - Tedious discovery of most adequate mapping
   - Dataset-dependent similarity functions



1. **Time complexity**
   - LIMES algorithm [NA11]
   - MultiBlock [IJB11]
   - HR^3 [Ngo12]
   - AEGLE [GSN16]
   - Summary and Challenges
2. **Accuracy**
   - RAVEN [Ngo+11]
   - EAGLE [NL12]
   - COALA [NLC13]
   - Summary and Challenges
3. **Benchmarking**
   - Benchmarking [NGF16]
   - Synthetic Benchmarks [Sav+15]
   - Real Benchmarks [Mor+11]
   - Summary and Challenges
4. **Hands-On Session**

That's all Folks!

Axel Ngonga
AKSW Research Group
Institute for Applied Informatics
ngonga@informatik.uni-leipzig.de

Irini Fundulaki
ICS FORTH
fundul@ics.forth.gr

Mohamed Ahmed Sherif
AKSW Research Group
Institute for Applied Informatics
sherif@informatik.uni-leipzig.de

This work was supported by grants from the EU H2020 Framework Programme provided for the project HOBBIT (GA no. 688227).

References I

Kleanthi Georgala, Mohamed Ahmed Sherif, and Axel-Cyrille Ngonga Ngomo. "An Efficient Approach for the Generation of Allen Relations". In: ECAI 2016 - 22nd European Conference on Artificial Intelligence, 29 August-2 September 2016, The Hague, The Netherlands - Including Prestigious Applications of Artificial Intelligence (PAIS 2016). Ed. by Gal A. Kaminka et al. Vol. 285. Frontiers in Artificial Intelligence and Applications. IOS Press, 2016, pp. 948–956. ISBN: 978-1-61499-671-2. DOI: 10.3233/978-1-61499-672-9-948. URL: http://dx.doi.org/10.3233/978-1-61499-672-9-948.

Robert Isele, Anja Jentzsch, and Christian Bizer. "Efficient Multidimensional Blocking for Link Discovery without losing Recall". In: Proceedings of the 14th International Workshop on the Web and Databases 2011, WebDB 2011, Athens, Greece, June 12, 2011. Ed. by Amélie Marian and Vasilis Vassalos. 2011. URL: http://webdb2011.rutgers.edu/papers/Paper%2039/silk.pdf.

Yasar Khan et al. "SAFE: Policy Aware SPARQL Query Federation Over RDF Data Cubes". In: Proceedings of the 7th International Workshop on Semantic Web Applications and Tools for Life Sciences, Berlin, Germany, December 9-11, 2014. Ed. by Adrian Paschke et al. Vol. 1320. CEUR Workshop Proceedings. CEUR-WS.org, 2014. URL: http://ceur-ws.org/Vol-1320/Preface_SWAT4LS2014.pdf.

Jens Lehmann et al. "deqa: Deep Web Extraction for Question Answering". In: The Semantic Web - ISWC 2012 - 11th International Semantic Web Conference, Boston, MA, USA, November 11-15, 2012, Proceedings, Part II. Ed. by Philippe Cudré-Mauroux et al. Vol. 7650. Lecture Notes in Computer Science. Springer, 2012, pp. 131–147. ISBN: 978-3-642-35172-3. DOI: 10.1007/978-3-642-35173-0_9. URL: http://dx.doi.org/10.1007/978-3-642-35173-0_9.

References III

Mohamed Morsey et al. "DBpedia SPARQL Benchmark - Performance Assessment with Real Queries on Real Data". In: The Semantic Web - ISWC 2011 - 10th International Semantic Web Conference, Bonn, Germany, October 23-27, 2011, Proceedings, Part I. Ed. by Lora Aroyo et al. Vol. 7031. Lecture Notes in Computer Science. Springer, 2011, pp. 454–469. ISBN: 978-3-642-25072-9. DOI: 10.1007/978-3-642-25073-6_29. URL: http://dx.doi.org/10.1007/978-3-642-25073-6_29.

Axel-Cyrille Ngonga Ngomo and Sören Auer. "LIMES - A Time-Efficient Approach for Large-Scale Link Discovery on the Web of Data". In: IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011. Ed. by Toby Walsh. IJCAI/AAAI, 2011, pp. 2312–2317. ISBN: 978-1-57735-516-8. DOI: 10.5591/978-1-57735-516-8/IJCAI11-385. URL: http://dx.doi.org/10.5591/978-1-57735-516-8/IJCAI11-385.

Markus Nentwig et al. "A survey of current Link Discovery frameworks". In: Semantic Web Preprint (2015), pp. 1–18.

Axel-Cyrille Ngonga Ngomo, Alejandra Garcıa-Rojas, and Irini Fundulaki. "HOBBIT: Holistic Benchmarking of Big Linked Data". In: ERCIM News 2016.105 (2016). URL: http://ercim-news.ercim.eu/en105/r-i/hobbit-holistic-benchmarking-of-big-linked-data.

Axel-Cyrille Ngonga Ngomo et al. "RAVEN - active learning of link specifications". In: Proceedings of the 6th International Workshop on Ontology Matching, Bonn, Germany, October 24, 2011. Ed. by Pavel Shvaiko et al. Vol. 814. CEUR Workshop Proceedings. CEUR-WS.org, 2011. URL: http://ceur-ws.org/Vol-814/om2011_Tpaper3.pdf.

Axel-Cyrille Ngonga Ngomo. "Link Discovery with Guaranteed Reduction Ratio in Affine Spaces with Minkowski Measures". In: The Semantic Web - ISWC 2012 - 11th International Semantic Web Conference, Boston, MA, USA, November 11-15, 2012, Proceedings, Part I. Ed. by Philippe Cudré-Mauroux et al. Vol. 7649. Lecture Notes in Computer Science. Springer, 2012, pp. 378–393. ISBN: 978-3-642-35175-4. DOI: 10.1007/978-3-642-35176-1_24. URL: http://dx.doi.org/10.1007/978-3-642-35176-1_24.

Axel-Cyrille Ngonga Ngomo and Klaus Lyko. "EAGLE: Efficient Active Learning of Link Specifications Using Genetic Programming". In: The Semantic Web: Research and Applications - 9th Extended Semantic Web Conference, ESWC 2012, Heraklion, Crete, Greece, May 27-31, 2012. Proceedings. Ed. by Elena Simperl et al. Vol. 7295. Lecture Notes in Computer Science. Springer, 2012, pp. 149–163. ISBN: 978-3-642-30283-1. DOI: 10.1007/978-3-642-30284-8_17. URL: http://dx.doi.org/10.1007/978-3-642-30284-8_17.

Axel-Cyrille Ngonga Ngomo, Klaus Lyko, and Victor Christen. "COALA - Correlation-Aware Active Learning of Link Specifications". In: The Semantic Web: Semantics and Big Data, 10th International Conference, ESWC 2013, Montpellier, France, May 26-30, 2013. Proceedings. Ed. by Philipp Cimiano et al. Vol. 7882. Lecture Notes in Computer Science. Springer, 2013, pp. 442–456. ISBN: 978-3-642-38287-1. DOI: 10.1007/978-3-642-38288-8_30. URL: http://dx.doi.org/10.1007/978-3-642-38288-8_30.

References VI

Muhammad Saleem et al. "TopFed: TCGA Tailored Federated Query Processing and Linking to LOD". In: J. Biomedical Semantics 5 (2014), p. 47. DOI: 10.1186/2041-1480-5-47. URL: http://dx.doi.org/10.1186/2041-1480-5-47.

Muhammad Saleem et al. "A fine-grained evaluation of SPARQL endpoint federation systems". In: Semantic Web 7.5 (2015), pp. 493–518. DOI: 10.3233/SW-150186. URL: http://dx.doi.org/10.3233/SW-150186.

Tzanina Saveta et al. "LANCE: Piercing to the Heart of Instance Matching Tools". In: The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part I. Ed. by Marcelo Arenas et al. Vol. 9366. Lecture Notes in Computer Science. Springer, 2015, pp. 375–391. ISBN: 978-3-319-25006-9. DOI: 10.1007/978-3-319-25007-6_22. URL: http://dx.doi.org/10.1007/978-3-319-25007-6_22.

Saeedeh Shekarpour, Axel-Cyrille Ngonga Ngomo, and Sören Auer. "Question answering on interlinked data". In: 22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013. Ed. by Daniel Schwabe et al. International World Wide Web Conferences Steering Committee / ACM, 2013, pp. 1145–1156. ISBN: 978-1-4503-2035-1. URL: http://dl.acm.org/citation.cfm?id=2488488.

Link Discovery Tutorial

Part I: Efficiency

Mohamed Ahmed Sherif[1], Irini Fundulaki[2], Axel-Cyrille Ngonga Ngomo[1]

(1) Institute for Applied Informatics, Germany
(2) FORTH, Greece

May 28th, 2017
Portoroz, Slovenia.

Table of Contents

Table of Contents

Definition (Declarative Link Discovery)

- Given sets S and T of resources and relation $\mathcal{R}$
- Find $M = \{(s,t) \in S \times T : \mathcal{R}(s,t)\}$
- Here, find $M' = \{(s,t) \in S \times T : \sigma(s,t) \geq \theta\}$

Definition (Declarative Link Discovery)

- Given sets S and T of resources and relation $\mathcal{R}$
- Find $M = \{(s,t) \in S \times T : \mathcal{R}(s,t)\}$
- Here, find $M' = \{(s,t) \in S \times T : \sigma(s,t) \geq \theta\}$

Problem

- Naïve complexity $\in O(S \times T)$, i.e., $O(n^2)$
- Example: $\approx$ 70 days for cities in DBpedia and LinkedGeoData

Definition (Declarative Link Discovery)

- Given sets S and T of resources and relation $\mathcal{R}$
- Find $M = \{(s,t) \in S \times T : \mathcal{R}(s,t)\}$
- Here, find $M' = \{(s,t) \in S \times T : \sigma(s,t) \geq \theta\}$

Problem

- Naïve complexity $\in O(S \times T)$, i.e., $O(n^2)$
- Example: $\approx$ 70 days for cities in DBpedia and LinkedGeoData

Solutions

1. Reduced number of comparisons $C(\mathcal{A}) \geq |M'|$ [NA11; IJB11; Ngo12; Ngo13]
2. Use planning [Ngo14]
3. Reduce the complexity of linking to $< n^2$ [GSN16]
4. Use features of hardware environment [Ngo+13; NH16]

Table of Contents

Definition (Declarative Link Discovery)

- Given sets S and T of resources and relation $\mathcal{R}$
- Find $M = \{(s,t) \in S \times T : \mathcal{R}(s,t)\}$
- Here, find $M' = \{(s,t) \in S \times T : \delta(s,t) \leq \tau\}$

- Some δ are distances (in the mathematical sense) [NA11]
- Examples
  - Levenshtein distance
  - Minkowski distance

Definition (Declarative Link Discovery)

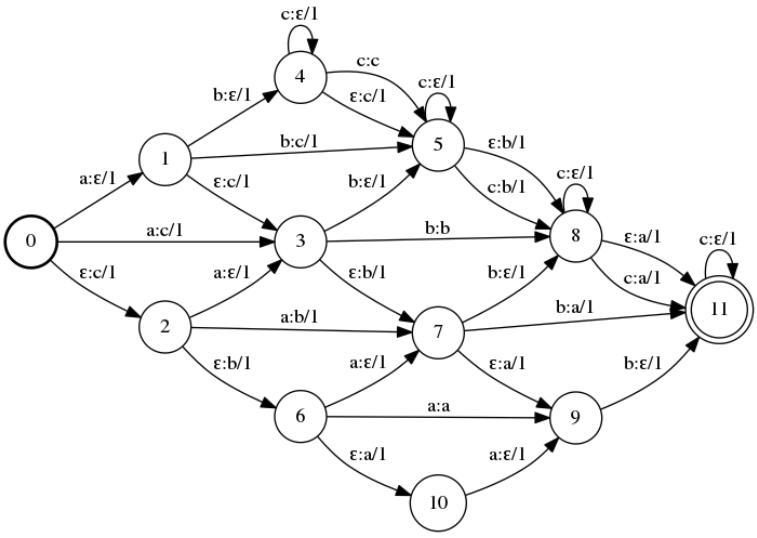
- Given sets S and T of resources and relation $\mathcal{R}$
- Find $M = \{(s,t) \in S \times T : \mathcal{R}(s,t)\}$
- Here, find $M' = \{(s,t) \in S \times T : \delta(s,t) \leq \tau\}$

- Some δ are distances (in the mathematical sense) [NA11]
- Examples
  - Levenshtein distance
  - Minkowski distance

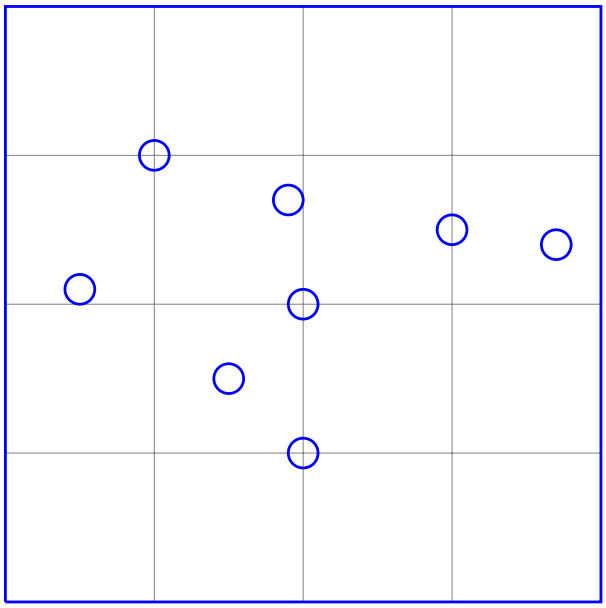
Intuition

- Distances abide by triangle inequality
- $\delta(x,z) - \delta(z,y) \leq \delta(x,y) \leq \delta(x,z) + \delta(z,y)$
- Use exemplars for pessimistic approximation
- Reduce number of computations

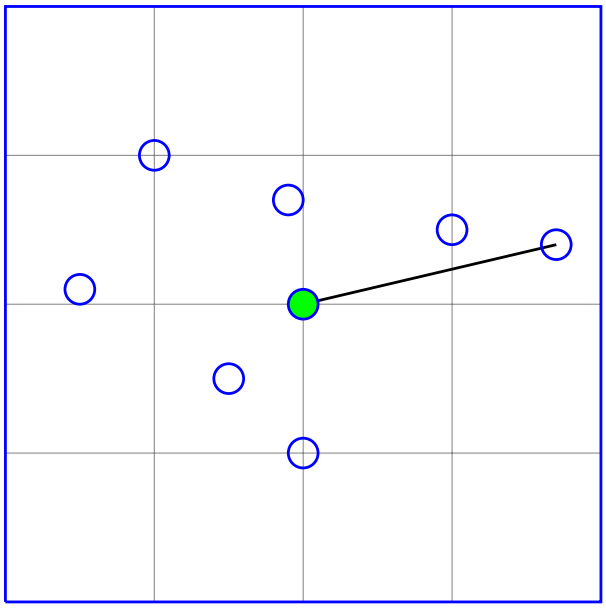
1. Start with random $e_1 \in T$
2. $e_{n+1} = \arg\max_{x \in T} \sum_{i=1}^{n} \delta(x, e_i)$
3. Assign each t to $e(t)$ with $e(t) = \arg\min_{e_i} \delta(t, e_i)$
4. Approximate $\delta(s, t)$ by using $\delta(s, t) \geq \delta(s, e(t)) - \delta(e(t), t)$



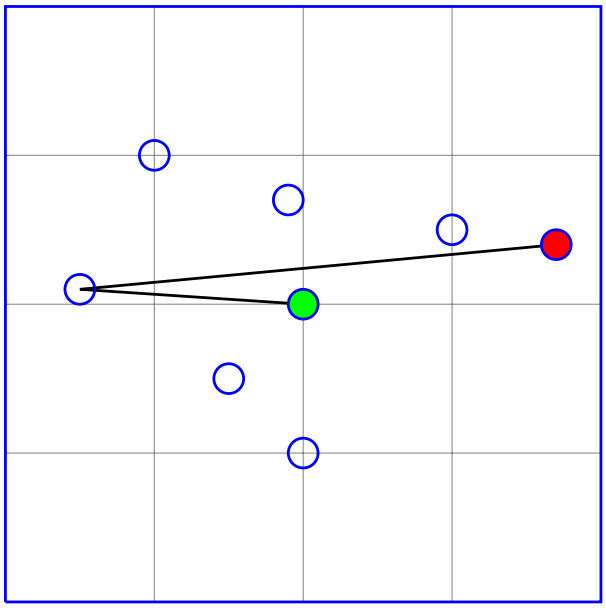
1. Start with random $e_1 \in T$



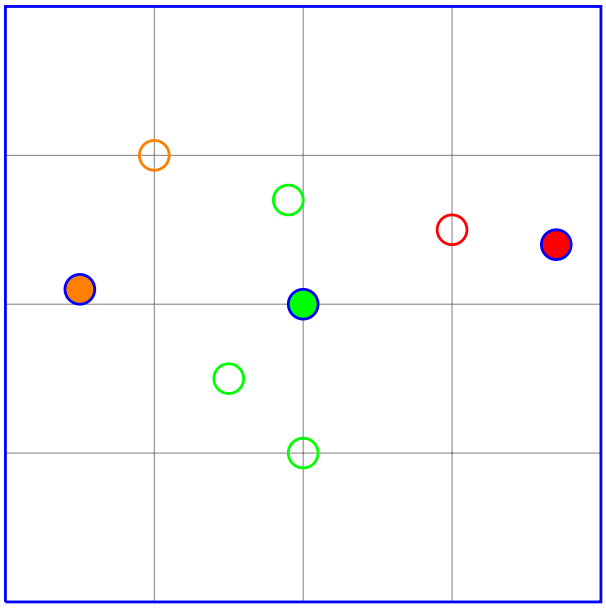
1. Start with random $e_1 \in T$



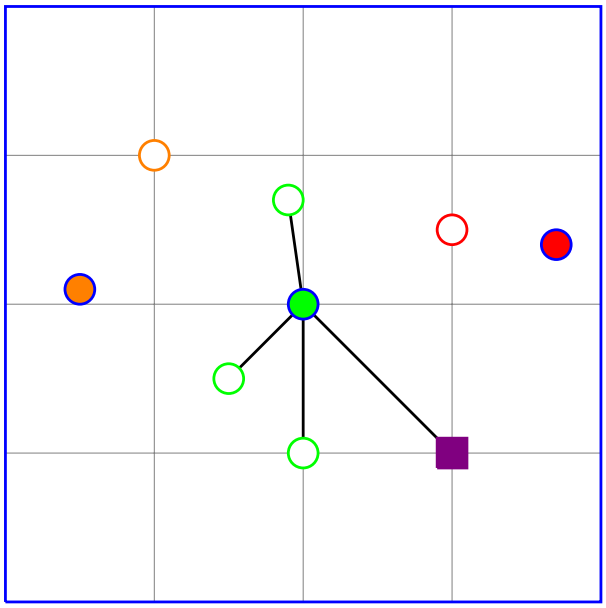
1. Start with random $e_1 \in T$
2. $e_{n+1} = \arg\max_{x \in T} \sum_{i=1}^{n} \delta(x, e_i)$



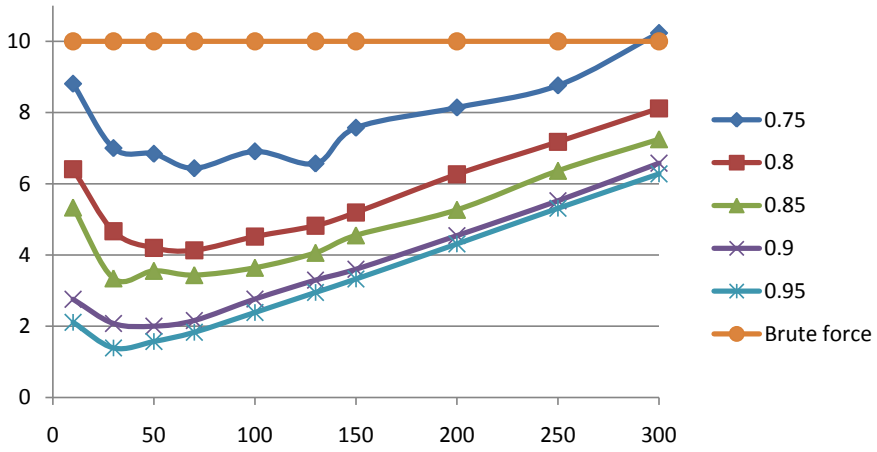
1. Start with random $e_1 \in T$
2. $e_{n+1} = \arg\max_{x \in T} \sum_{i=1}^{n} \delta(x, e_i)$
3. Assign each t to $e(t)$ with $e(t) = \arg\min_{e_i} \delta(t, e_i)$



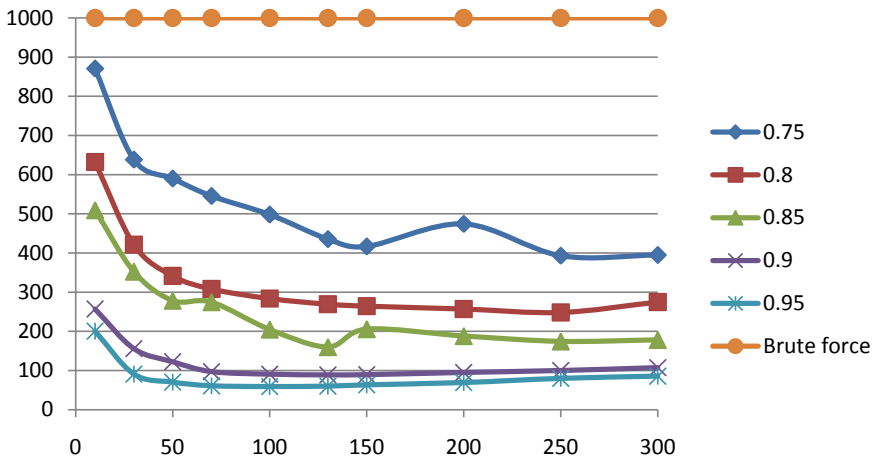
1. Start with random $e_1 \in T$
2. $e_{n+1} = \arg\max_{x \in T} \sum_{i=1}^{n} \delta(x, e_i)$
3. Assign each t to $e(t)$ with $e(t) = \arg\min_{e_i} \delta(t, e_i)$
4. Approximate $\delta(s, t)$ by using $\delta(s, t) \geq \delta(s, e(t)) - \delta(e(t), t)$



- Measure = Levenshtein
- Threshold = 0.9
- KB base size = 10^3
- x-axis is number of exemplars
- y-axis is number of comparisons



- Measure = Levenshtein
- Threshold = 0.9
- KB base size = 10^4
- x-axis is number of exemplars
- y-axis is number of comparisons
- Optimal number of examplars $\approx \sqrt{|T|}$



- Time-efficent approach
- Reduces number of comparisons through approximation
- No guarantee pertaining to reduction ratio

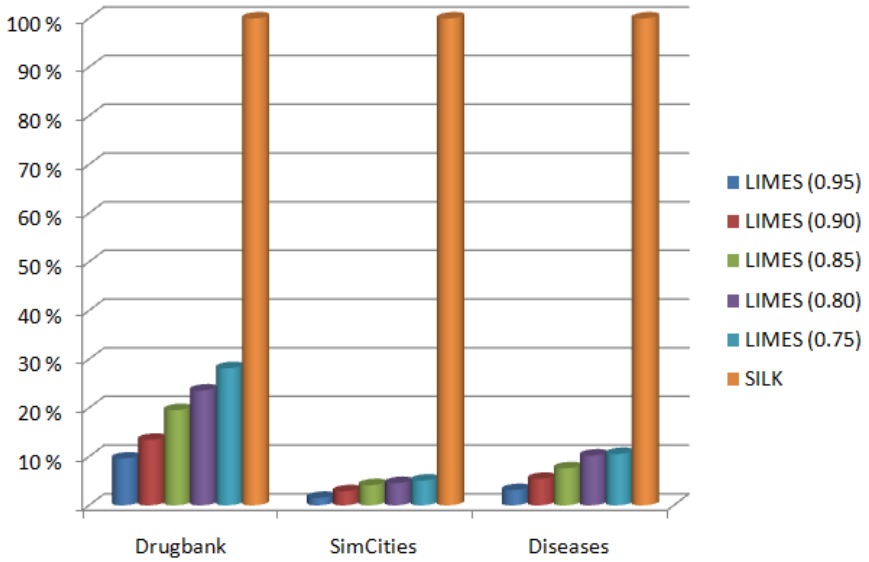


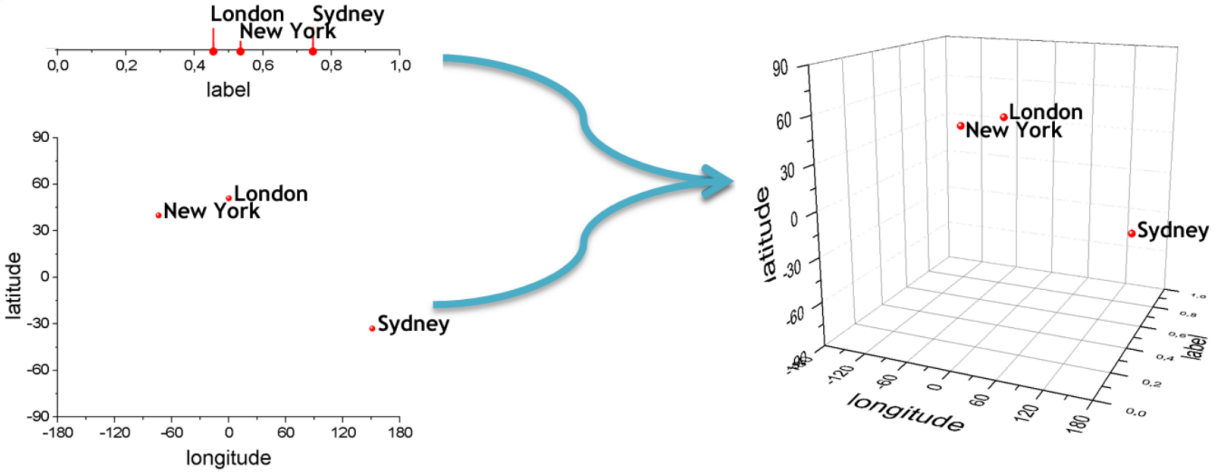
Table of Contents

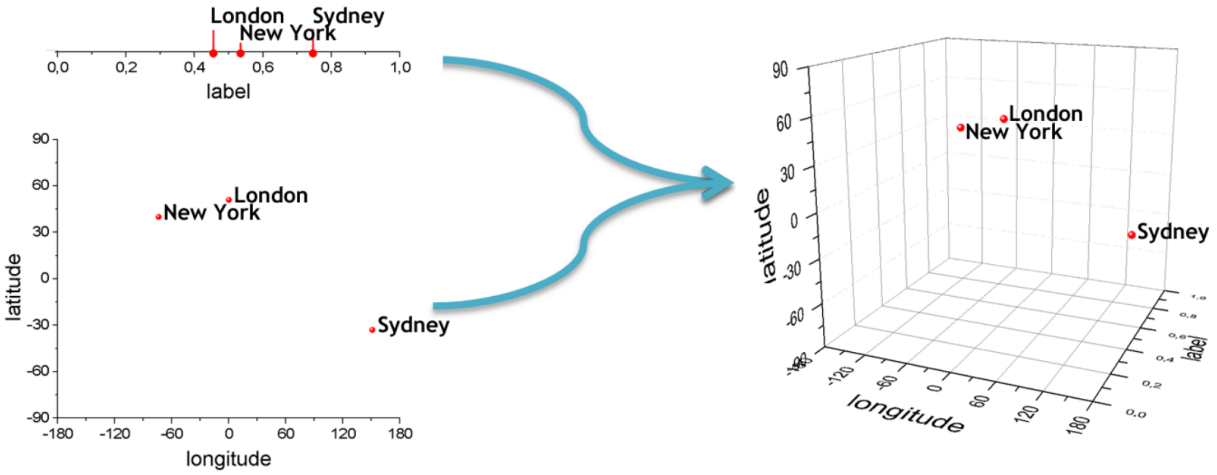
Idea

- Create multidimensional index of data
- Partition the space so that $\sigma(x, y) < \theta \Rightarrow \mathit{index}(x) \neq \mathit{index}(y)$
- Decrease number of computations by only comparing pairs (x, y) with $\mathit{index}(x) = \mathit{index}(y)$

Idea

- Create multidimensional index of data
- Partition the space so that $\sigma(x, y) < \theta \Rightarrow index(x) \neq index(y)$
- Decrease number of computations by only comparing pairs (x, y) with $index(x) = index(y)$





- Similarity functions are **aggregations** of atomic measures *sim*, e.g., $\sigma = MIN(levenshtein, jaccard)$
- Each atomic measure must define a blocking function $block : (S \cup T) \times [0, 1] \rightarrow \mathcal{P}(\mathbb{N}^n)$
- Each aggregation must define a **similarity**, a **block** and a **threshold aggregation**

- $sim = \frac{levenshtein(s,t)}{\max(|s|,|t|)}$ (normalized levenshtein)
- $block : \Sigma^q \rightarrow \mathbb{N}$
  - Maps each q-gram to exactly one block (e.g., Goedelisation)
  - Only $c = \max(|s|, |t|)(1 - \theta) \cdot q + 1$ q-grams to be indexed per string
- $index(s, \theta) = \{block(qgrams(s_{[0...c]})\}$
- String mapped to several blocks
- Comparison only within blocks

- $sim = \frac{levenshtein(s,t)}{\max(|s|,|t|)}$ (normalized levenshtein)
- $block : \Sigma^q \rightarrow \mathbb{N}$
  - Maps each q-gram to exactly one block (e.g., Goedelisation)
  - Only $c = \max(|s|, |t|)(1-\theta) \cdot q + 1$ q-grams to be indexed per string
- $index(s, \theta) = \{block(qgrams(s_{[0...c]})\}$
- String mapped to several blocks
- Comparison only within blocks

- Comparison of drugs in DBpedia and Drugbank

Setting	Comparisons	Runtime	Links
Full comparison	22,242,292	430s	1,403
Blocking (100 blocks)	906,314	44s	1,349
Blocking (1,000 blocks)	322,573	14s	1,287
MultiBlock	122,630	6s	1,403

Table of Contents

Definition (Declarative Link Discovery)

- Given sets S and T of resources and relation $\mathcal{R}$
- Find $M = \{(s,t) \in S \times T : \mathcal{R}(s,t)\}$
- Here, find $M' = \{(s,t) \in S \times T : \delta(s,t) \leq \theta\}$

Definition (Declarative Link Discovery)

- Given sets S and T of resources and relation $\mathcal{R}$
- Find $M = \{(s,t) \in S \times T : \mathcal{R}(s,t)\}$
- Here, find $M' = \{(s,t) \in S \times T : \delta(s,t) \leq \theta\}$

- Intuition: Reduce the number of comparisons $C(\mathcal{A}) \geq |M'|$
- Maximize reduction ratio: $RR(\mathcal{A}) = 1 - \frac{C(\mathcal{A})}{|S||T|}$

Definition (Declarative Link Discovery)

- Given sets S and T of resources and relation $\mathcal{R}$
- Find $M = \{(s,t) \in S \times T : \mathcal{R}(s,t)\}$
- Here, find $M' = \{(s,t) \in S \times T : \delta(s,t) \leq \theta\}$

- Intuition: Reduce the number of comparisons $C(\mathcal{A}) \geq |M'|$
- Maximize reduction ratio: $RR(\mathcal{A}) = 1 - \frac{C(\mathcal{A})}{|S||T|}$

Question

- Can we devise lossless approaches with guaranteed RR?
- Advantages
  - Space management
  - Runtime prediction
  - Resource scheduling

- Best achievable reduction ratio: $RR_{\text{max}} = 1 - \frac{|M'|}{|S||T|}$

- Best achievable reduction ratio: $RR_{\text{max}} = 1 - \frac{|M'|}{|S||T|}$
- Approach $\mathcal{H}(\alpha)$ fulfills RR guarantee criterion, iff:

$$\forall r < RR_{\text{max}}, \exists \alpha : RR(\mathcal{H}(\alpha)) \geq r$$

- Best achievable reduction ratio: $RR_{\text{max}} = 1 - \frac{|M'|}{|S||T|}$
- Approach $\mathcal{H}(\alpha)$ fulfills RR guarantee criterion, iff:

$$\forall r < RR_{\text{max}}, \exists \alpha : RR(\mathcal{H}(\alpha)) \geq r$$

- Here, we use relative reduction ratio (RRR):

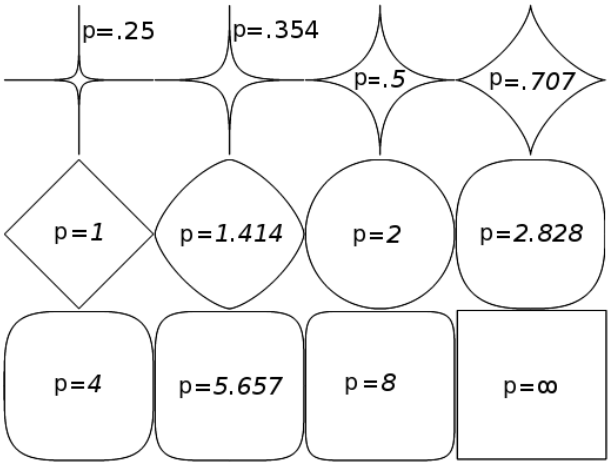
$$RRR(\mathcal{A}) = \frac{RR_{\text{max}}}{RR(\mathcal{A})}$$

Formal Goal

Devise $\mathcal{H}(\alpha) : \forall r > 1, \exists \alpha : RRR(\mathcal{H}(\alpha)) \leq r$

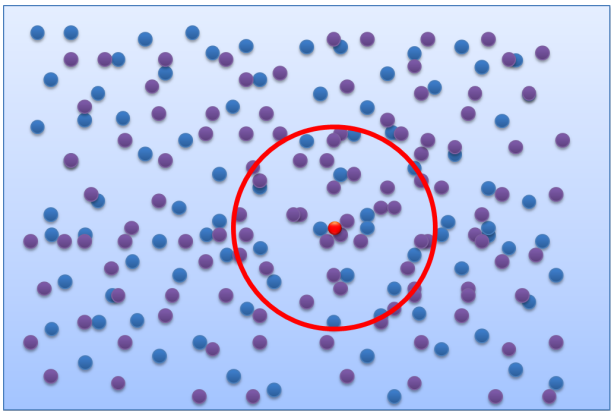
Minkowski Distance

$$\delta(s,t) = \sqrt[p]{\sum_{i=1}^{n} |s_i - t_i|^p}, p \geq 2$$



HYPPO

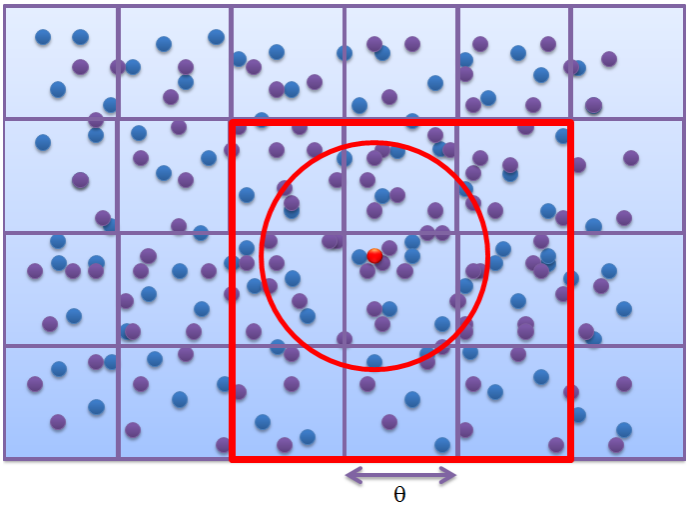
- $\delta(s,t) \leq \theta$ describes a hypersphere
- Approximate hypersphere by using a hypercube
  - Easy to compute
  - No loss of recall (blocking)



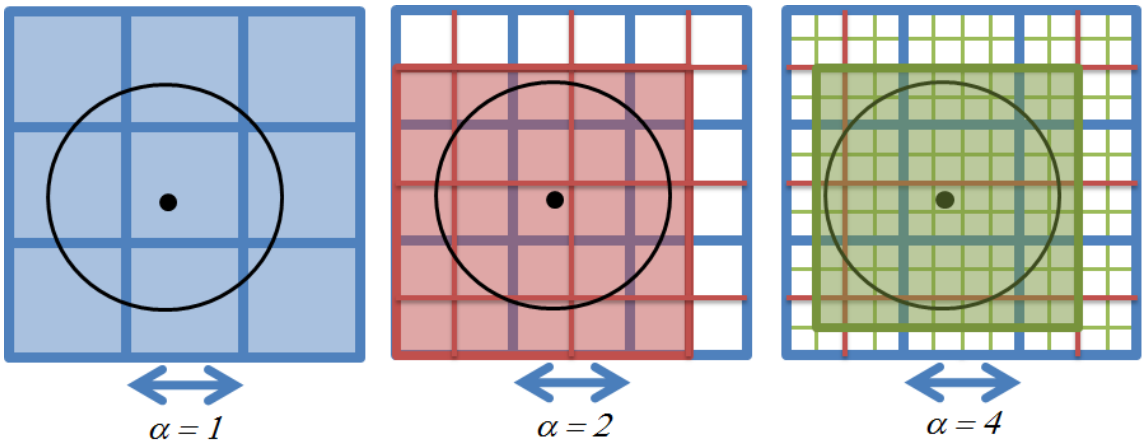
- Set width of single hypercube to $\Delta = \theta/\alpha$

- Set width of single hypercube to $\Delta = \theta/\alpha$
- Tile $\Omega = S \cup T$ into the adjacent cubes C
  - Coordinates: $(c_1, \ldots, c_n) \in \mathbb{N}^n$
  - Contains points $\omega \in \Omega : \forall i \in \{1 \ldots n\}, c_i\Delta \leq \omega_i < (c_i + 1)\Delta$

- Set width of single hypercube to $\Delta = \theta/\alpha$
- Tile $\Omega = S \cup T$ into the adjacent cubes C
  - Coordinates: $(c_1, \ldots, c_n) \in \mathbb{N}^n$
  - Contains points $\omega \in \Omega : \forall i \in \{1 \ldots n\}, c_i\Delta \leq \omega_i < (c_i + 1)\Delta$



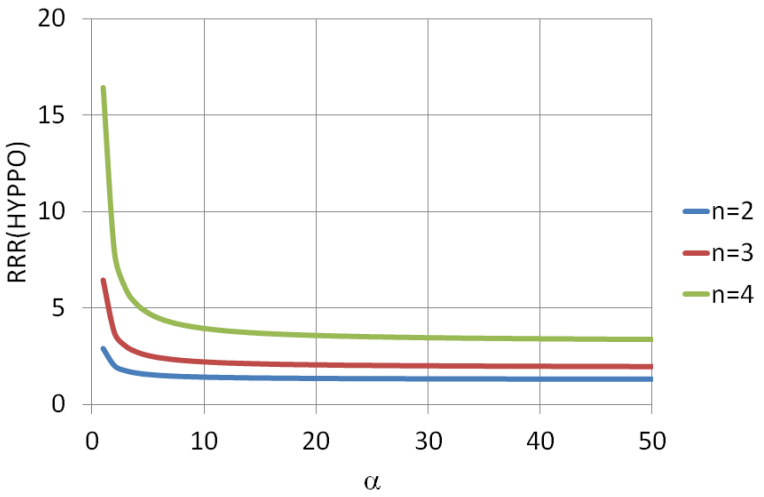
- Combine $(2\alpha + 1)^n$ hypercubes around $C(\omega)$ to approximate hypersphere



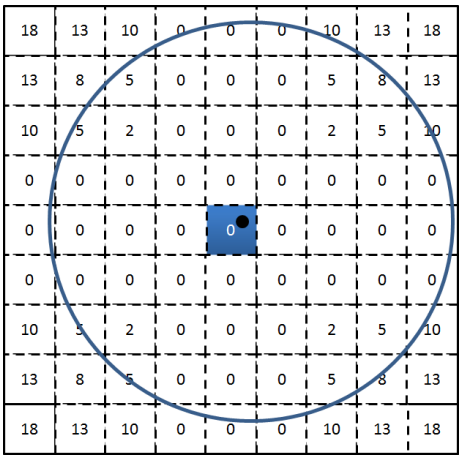
- $RRR(HYPPO(\alpha)) = \frac{(2\alpha+1)^n}{\alpha^n S(n)}$
- $\lim_{\alpha \to \infty} RRR(HYPPO(\alpha)) = \frac{2^n}{S(n)}$

- RRR(HYPPO) for $p = 2$, $n = 2, 3, 4$ and $2 \leq \alpha \leq 50$

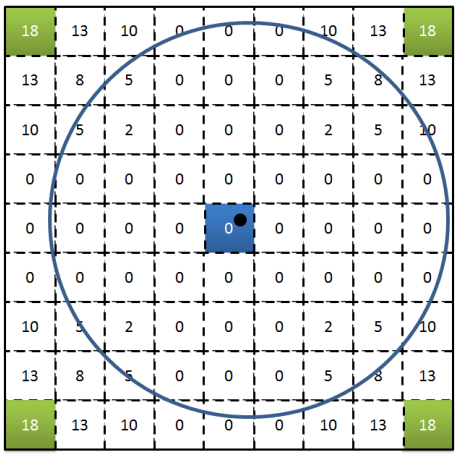
- RRR(HYPPO) for $p = 2$, $n = 2, 3, 4$ and $2 \leq \alpha \leq 50$
- $\lim_{\alpha \to \infty} RRR(HYPPO(\alpha)) = \frac{4}{\pi} \approx 1.27$ $(n = 2)$
- $\lim_{\alpha \to \infty} RRR(HYPPO(\alpha)) = \frac{6}{\pi} \approx 1.91$ $(n = 3)$
- $\lim_{\alpha \to \infty} RRR(HYPPO(\alpha)) = \frac{32}{\pi^2} \approx 3.24$ $(n = 4)$



$$index(C, \omega) = \begin{cases} 0 \text{ if } \exists i : |c_i - c(\omega)_i| \leq 1, 1 \leq i \leq n, \\ \sum_{i=1}^{n} (|c_i - c(\omega)_i| - 1)^p \text{ else,} \end{cases}$$



- Compare $C(\omega)$ with C iff $index(C, \omega) \leq \alpha^p$
- $\alpha = 4, p = 2$



Claims

- No loss of recall
- $\lim_{\alpha \to \infty} RRR(\mathcal{HR}^3(\alpha)) = 1$

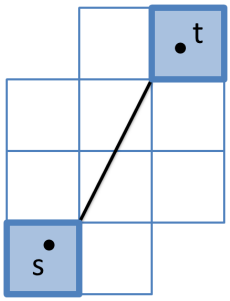
Lemma

$index(C, s) = x \Rightarrow \forall t \in C \ \delta^p(s, t) > x\Delta^p$

Lemma

$index(C, s) = x \Rightarrow \forall t \in C \ \delta^p(s, t) > x\Delta^p$

- $p = 2, n = 2, index(C, s) = 5$



Lemma

$index(C, s) = x \Rightarrow \forall t \in C\ \delta^p(s, t) > x\Delta^p$

Proof.

- $index(C, s) = x \Rightarrow \sum_{i=1}^{n}(|c_i - c_i(s)| - 1)^p = x$
- Out of definition of cube index follows $|s_i - t_i| > (|c_i - c_i(s)| - 1)\Delta$
- Thus, $\sum_{i=1}^{n}|s_i - t_i|^p > \sum_{i=1}^{n}(|c_i - c_i(s)| - 1)^p\Delta^p$
- Therewith, $\delta^p(s, t) > x\Delta^p$

□

Lemma

$\forall s \in S$: *index*$(C, s) > \alpha^p$ *implies that all* $t \in C$ *are non-matches*

Proof.

Follows directly from Lemma 1:
index$(C, s) > \alpha^p \Rightarrow \forall t \in C, \delta^p(s, t) > \Delta^p \alpha^p = \theta^p$ □

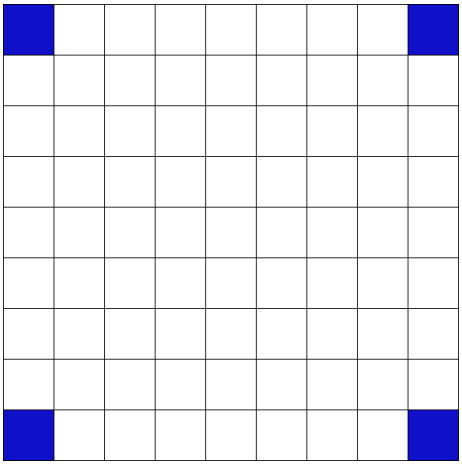
Claims

- No loss of recall ✓
- $\lim_{\alpha \to \infty} RRR(\mathcal{HR}^3(\alpha)) = 1$

Lemma

$\forall \alpha > 1\ RRR(\mathcal{HR}^3(2\alpha)) < RRR(\mathcal{HR}^3(\alpha))$

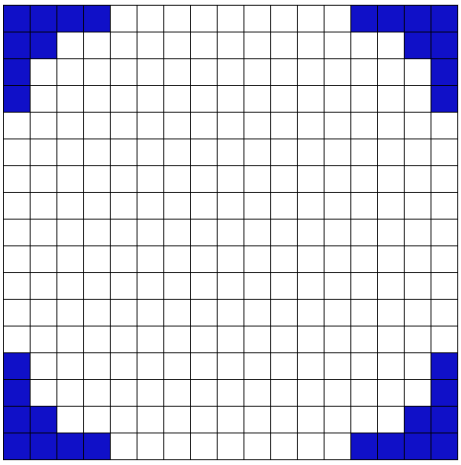
- $p = 2, \alpha = 4$



Lemma

$\forall \alpha > 1\ RRR(HR^3(2\alpha)) < RRR(HR^3(\alpha))$

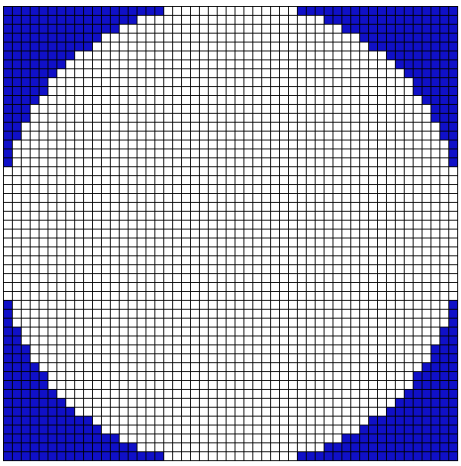
- $p = 2, \alpha = 8$



Lemma

$\forall \alpha > 1\ RRR(HR^3(2\alpha)) < RRR(HR^3(\alpha))$

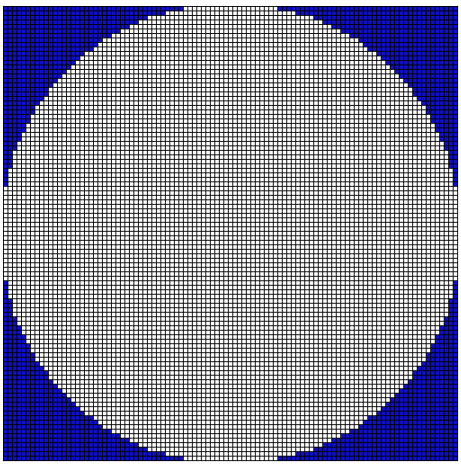
- $p = 2, \alpha = 25$



Lemma

$\forall \alpha > 1\ RRR(HR^3(2\alpha)) < RRR(HR^3(\alpha))$

- $p = 2, \alpha = 50$



Theorem

$$\lim_{\alpha \to \infty} RRR(\mathcal{HR}^3(\alpha)) = 1$$

Proof.

- $\alpha \to \infty \Rightarrow \Delta \to 0$
- Thus, $C(s) = \{s\}, C = \{t\}$
- Index function : $\sum_{i=1}^{n} \Delta^p(|c_i(s) - c_i| - 1)^p \leq \Delta^p \alpha^p$
- $\Delta \to 0 \Rightarrow \sum_{i=1}^{n} |s_i - t_i|^p \leq \theta^p$

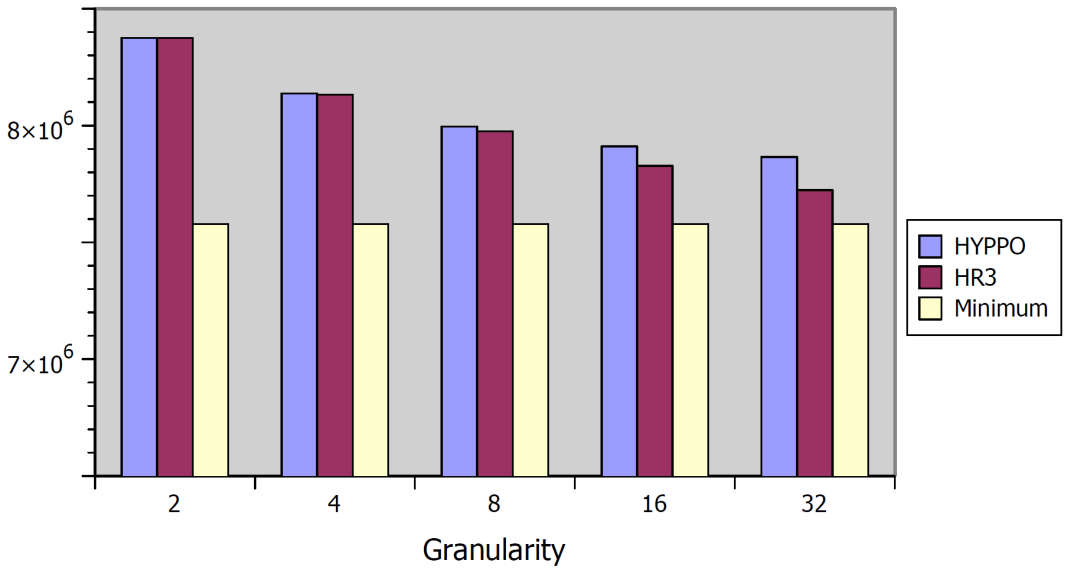
□

Claims

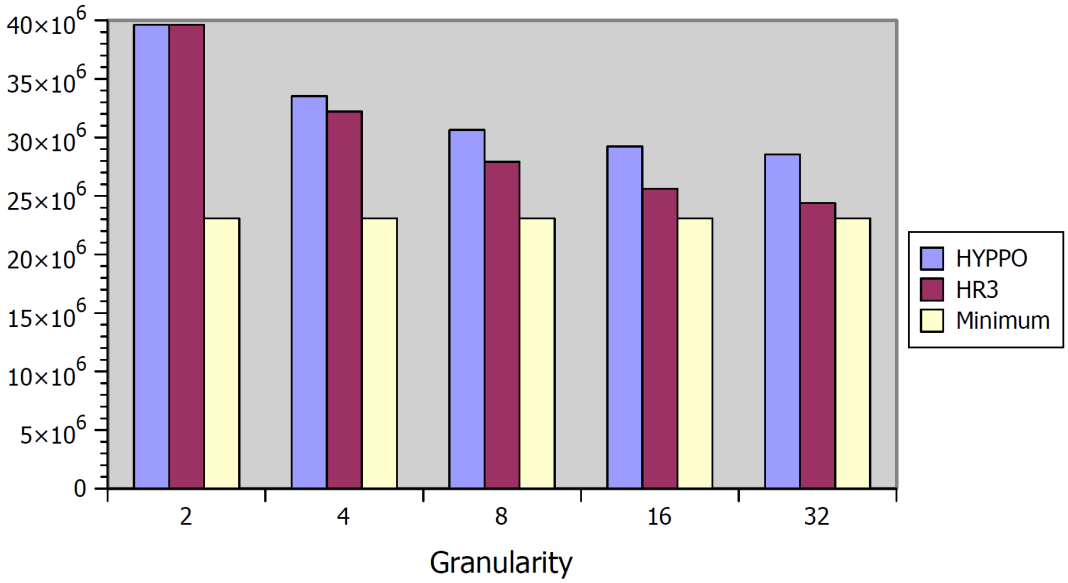
- No loss of recall ✓
- $\lim_{\alpha \to \infty} RRR(\mathcal{HR}^3(\alpha)) = 1$ ✓

- Compare $\mathcal{HR}^3$ with LIMES 0.5's HYPPO and SILK 2.5.1
- Experimental Setup:
  - Deduplicating DBpedia places by minimum elevation, elevation and maximum elevation ($\theta = 49m, 99m$).
  - Geonames and LinkedGeoData by longitude and latitude ($\theta = 1°, 9°$)
- Windows 7 Enterprise machine 64-bit computer with a 2.8GHz i7 processor with 8GB RAM.

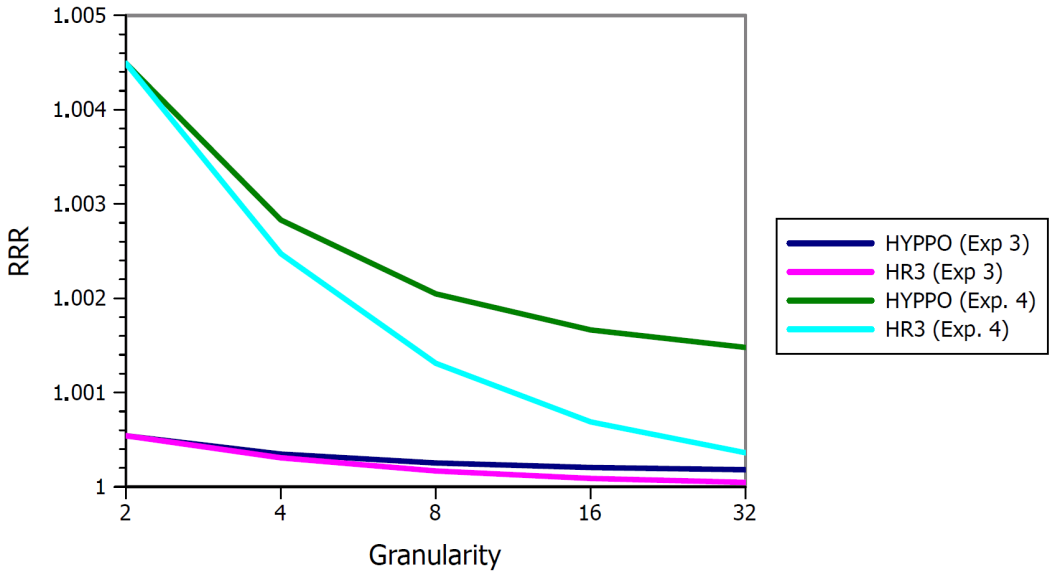
- Experiment 2: Deduplicating DBpedia places, $\theta = 99m$
- 0.64×10^6 less comparisons



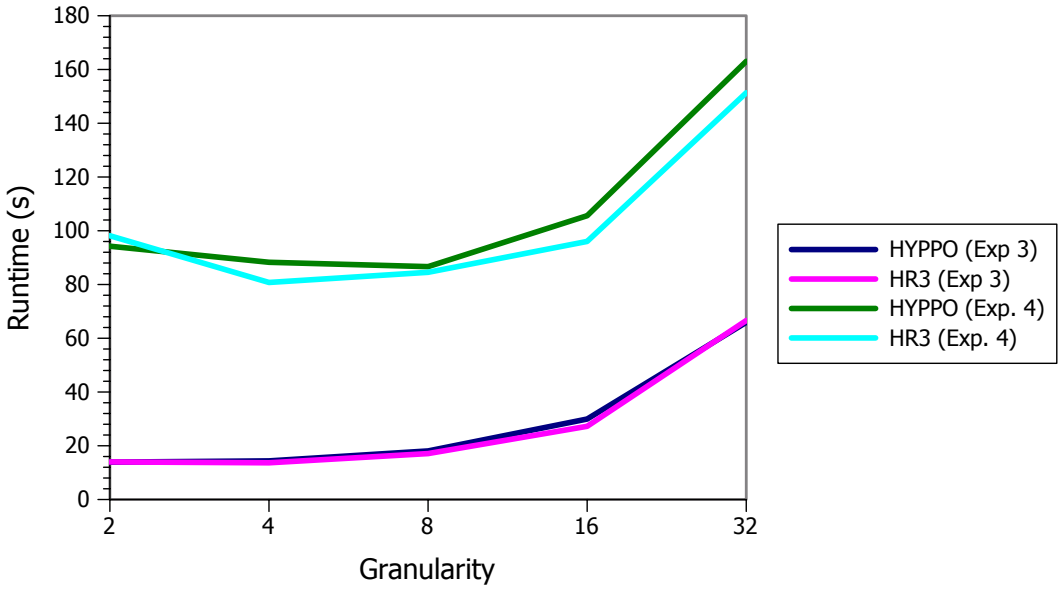
- Experiment 4: Linking Geonames and LinkedGedData, $\theta = 9°$
- 4.3×10^6 less comparisons



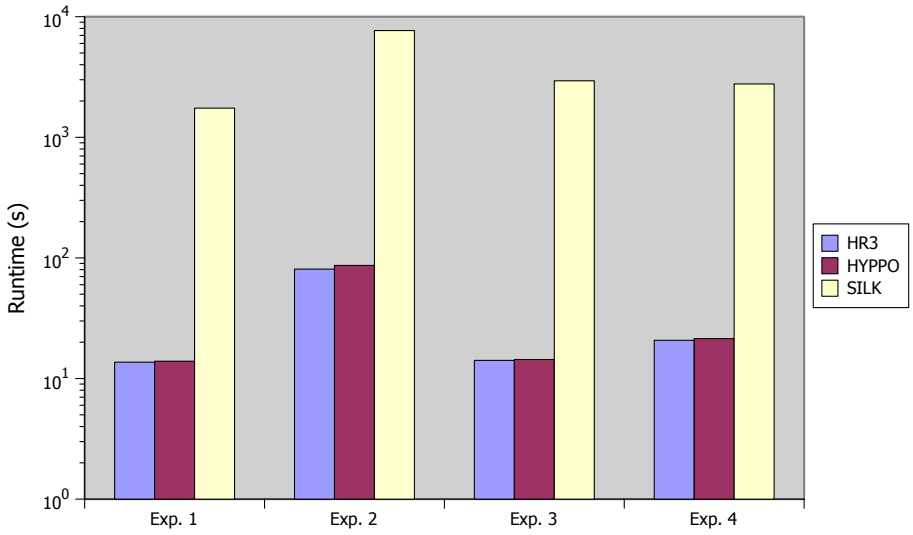
- Experiment 3,4: Geonames and LGD, $\theta = 1,9°$



- Experiment 3,4: Geonames and LGD, $\theta = 1,9°$

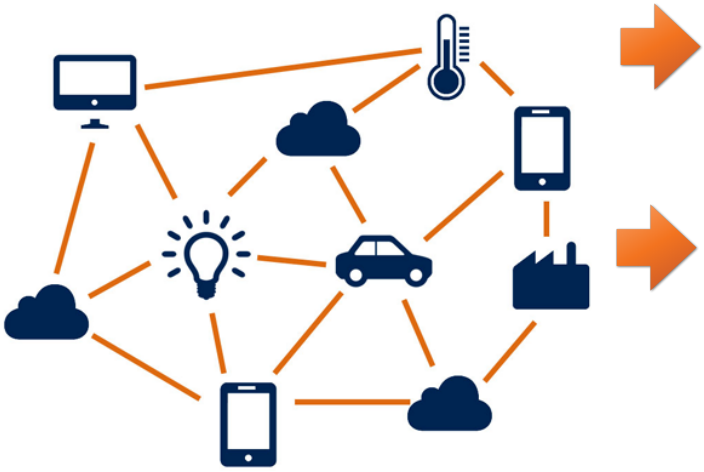


- Experiment 1, 2: DBpedia, $\theta = 49,99m$
- Experiment 3, 4: Geonames and LGD, $\theta = 1,9°$



- **Main result**: new category of algorithms for link discovery
- Outperforms the state of the art (runtime, comparisons)
- Future Work
  - Combine $\mathcal{HR}^3$ with multi-indexing approach
  - Devise resource management approach
  - Develop other algorithms (esp. for strings) with the same/similar theoretical guarantees

Table of Contents



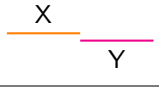
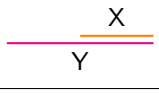
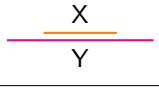
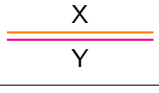
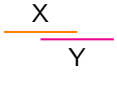
```
:E1 rdfs:label "Engine failure"@en
:E1 rdf:type :Error
:E1 :beginDate :"2016-04-22T11:39:35"
:E1 :endDate :"2015-04-22T11:39:37"
```

```
:E2 rdfs:label "Car accident"@en
:E2 rdf:type :Accident
:E2 :beginDate :"2015-06-28T11:45:22"
:E2 :endDate :"2015-06-28T11:45:24"
```

- Need to create links between events, e.g., `:startsEvent`
- Need to deal with volume and velocity

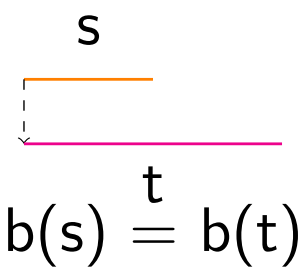
Definition (Event)

- Events can be modeled as time intervals: $v = (b(v), e(v))$
  - $b(v)$ is the beginning time (:beginDate)
  - $e(v)$ is the end time (:endDate)
  - $b(v) < e(v)$

Relation	Notation	Inverse	Illustration
X before Y	$bf(X, Y)$	$bfi(X, Y)$	
X meets Y	$m(X, Y)$	$mi(X, Y)$	
X finishes Y	$f(X, Y)$	$fi(X, Y)$	
X starts Y	$st(X, Y)$	$sti(X, Y)$	
X during Y	$d(X, Y)$	$di(X, Y)$	
X equal Y	$eq(X, Y)$	$eq(X, Y)$	
X overlaps with Y	$ov(X, Y)$	$ovi(X, Y)$	

AEGLE: Allen's intErval alGebra for Link discovEry

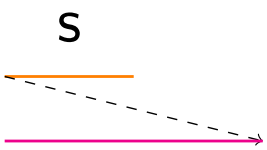
- Efficient computation of temporal relations between events
- Allen's Interval Algebra: distinct, exhaustive, and qualitative relations between time intervals
- Intuition
  - Expressing 13 Allen relations using 8 atomic relations
  - Time is ordered: Find matching entities using two sorted lists


s
t
b(s) = b(t)

$$b(s) = b(t)$$

$$b(s) < e(t)$$

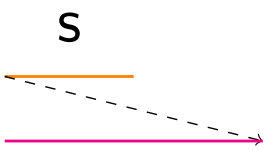
$$b(s) \overset{t}{=} b(t)$$



$$b(s) \overset{t}{<} e(t)$$

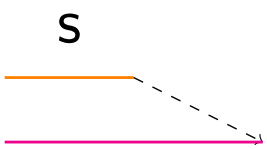
$$e(s) \overset{t}{>} b(t)$$

$$b(s) \overset{t}{=} b(t)$$



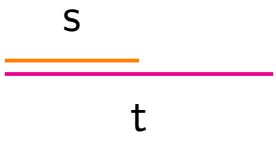
$$b(s) \overset{t}{<} e(t)$$

$$e(s) \overset{t}{>} b(t)$$

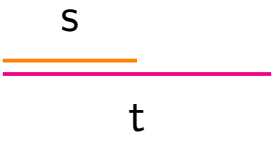


$$e(s) \overset{t}{<} e(t)$$

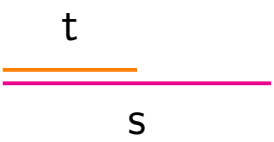
- Compute 8 atomic Boolean relations between begin and end points
  - *BeginBegin* (*BB*) for $b(s), b(t)$:
    - $BB^1(s,t) \Leftrightarrow (b(s) < b(t))$
    - $BB^0(s,t) \Leftrightarrow (b(s) = b(t))$
    - $BB^{-1}(s,t) \Leftrightarrow (b(s) > b(t)) \Leftrightarrow \neg(BB^1(s,t) \vee BB^0(s,t))$
  - *BeginEnd*(*BE*) for $b(s), e(t)$
  - *EndBegin*(*EB*) for $e(s), b(t)$
  - *EndEnd*(*EE*) for $e(s), e(t)$



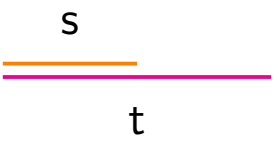
$$st(s,t) \Leftrightarrow BB^0(s,t) \wedge BE^1(s,t) \wedge EB^{-1}(s,t) \wedge EE^1(s,t) \Leftrightarrow \{BB^0(s,t) \wedge EE^1(s,t)\}$$



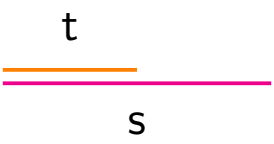
$$st(s,t) \Leftrightarrow BB^0(s,t) \wedge BE^1(s,t) \wedge EB^{-1}(s,t) \wedge EE^1(s,t) \Leftrightarrow \{BB^0(s,t) \wedge EE^1(s,t)\}$$



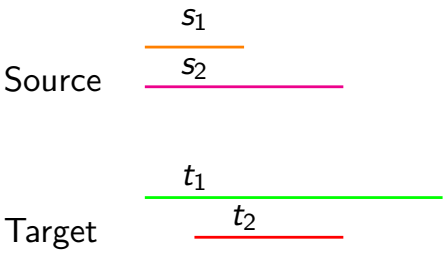
$$sti(s,t) \Leftrightarrow BB^0(s,t) \wedge BE^1(s,t) \wedge EB^{-1}(s,t) \wedge EE^{-1}(s,t) \Leftrightarrow$$
$$\{BB^0(s,t) \wedge EE^{-1}(s,t)\} =$$
$$\{\ BB^0(s,t) \wedge \neg(EE^0(s,t) \vee\ EE^1(s,t))\}$$



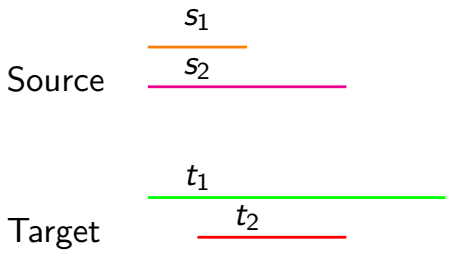
$$st(s,t) \Leftrightarrow BB^0(s,t) \wedge BE^1(s,t) \wedge EB^{-1}(s,t) \wedge EE^1(s,t) \Leftrightarrow \{BB^0(s,t) \wedge EE^1(s,t)\}$$



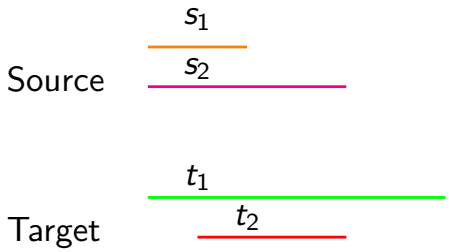
$$sti(s,t) \Leftrightarrow BB^0(s,t) \wedge BE^1(s,t) \wedge EB^{-1}(s,t) \wedge EE^{-1}(s,t) \Leftrightarrow$$
$$\{BB^0(s,t) \wedge EE^{-1}(s,t)\} =$$
$$\{\ BB^0(s,t) \wedge \neg(EE^0(s,t) \vee\ EE^1(s,t))\}$$


s_1
s_2
Source
t_1
t_2
Target

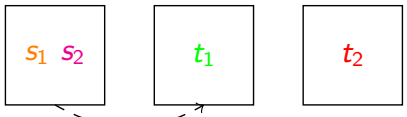
- For *st*:



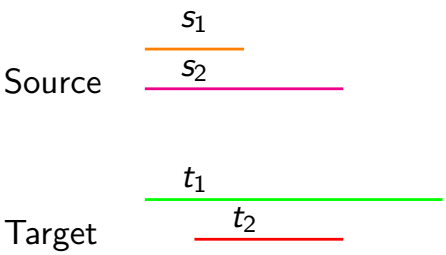
- For *st*:
  - Compute BB^0:

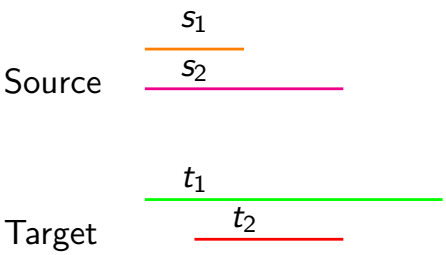


- For *st*:

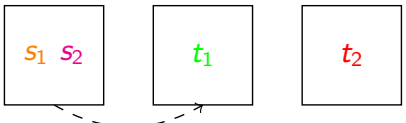


  - Compute BB^0:
    - $\{(s_1, t_1), (s_2, t_1)\}$

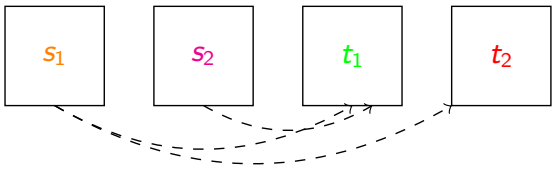




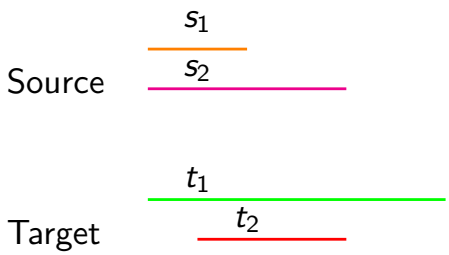
- For *st*:



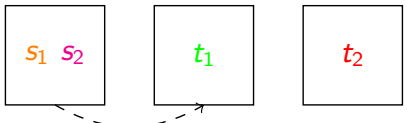
  - Compute BB^0:
    - $\{(s_1, t_1), (s_2, t_1)\}$



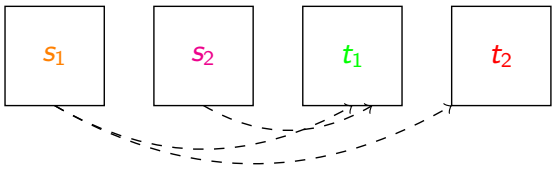
  - Compute EE^1:
    - $\{(s_1, t_1), (s_2, t_1), (s_1, t_2)\}$



- For *st*:



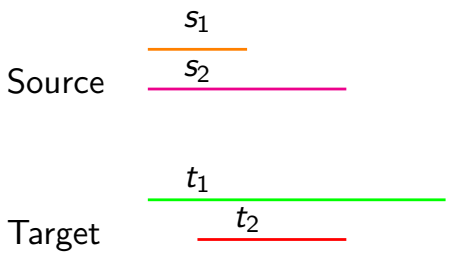
  - Compute BB^0:
    - $\{(s_1, t_1), (s_2, t_1)\}$



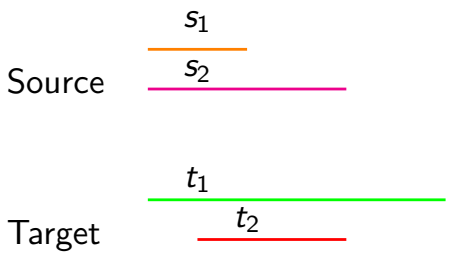
  - Compute EE^1:
    - $\{(s_1, t_1), (s_2, t_1), (s_1, t_2)\}$

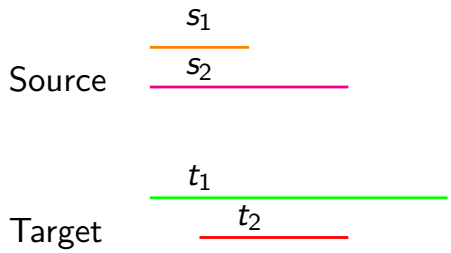
- Intersection between BB^0 and EE^1: $\{(s_1, t_1), (s_2, t_1)\}$

- For *sti*:

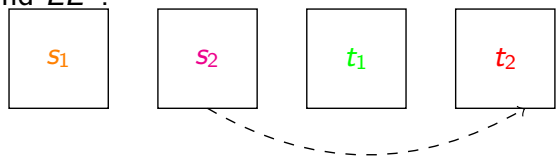


- For *sti*:
  - Retrieve BB^0 and EE^1:

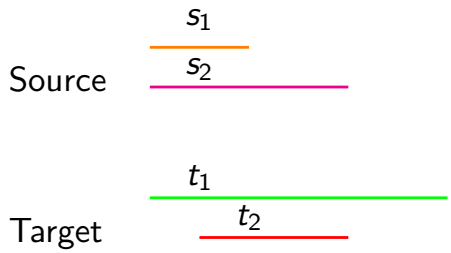




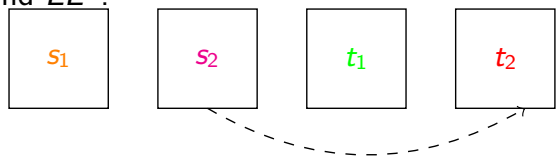
- For *sti*:
  - Retrieve BB^0 and EE^1:



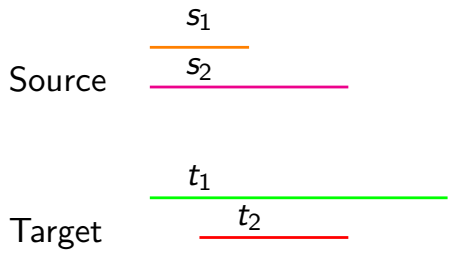
  - Compute EE^0:
    - $\{(s_2, t_2)\}$



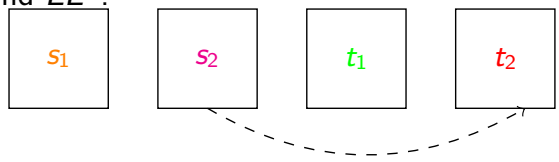
- For *sti*:
  - Retrieve BB^0 and EE^1:



  - Compute EE^0:
    - $\{(s_2, t_2)\}$
  - Union between EE^0 and EE^1: $\{(s_1, t_1), (s_2, t_1), (s_2, t_1), (s_2, t_2)\}$



- For *sti*:
  - Retrieve BB^0 and EE^1:



  - Compute EE^0:
    - $\{(s_2, t_2)\}$
  - Union between EE^0 and EE^1: $\{(s_1, t_1), (s_2, t_1), (s_2, t_1), (s_2, t_2)\}$
  - Difference between BB^0 and EE^0, EE^1: $\{(s_1, t_2), (s_2, t_2)\}$

- Datasets: $S = T$

Log Type	Dataset name	Size	Unique $b(s)$	Unique $e(s)$
Machinery	3KMachines	3,154	960	960
	30KMachines	28,869	960	960
	300KMachines	288,690	960	960
Query	3KQueries	3,888	3,636	3,638
	30KQueries	30,635	3,070	3,070
	300KQueries	303,991	184	184

- Datasets: $S = T$

Log Type	Dataset name	Size	Unique $b(s)$	Unique $e(s)$
Machinery	3KMachines	3,154	960	960
	30KMachines	28,869	960	960
	300KMachines	288,690	960	960
Query	3KQueries	3,888	3,636	3,638
	30KQueries	30,635	3,070	3,070
	300KQueries	303,991	184	184

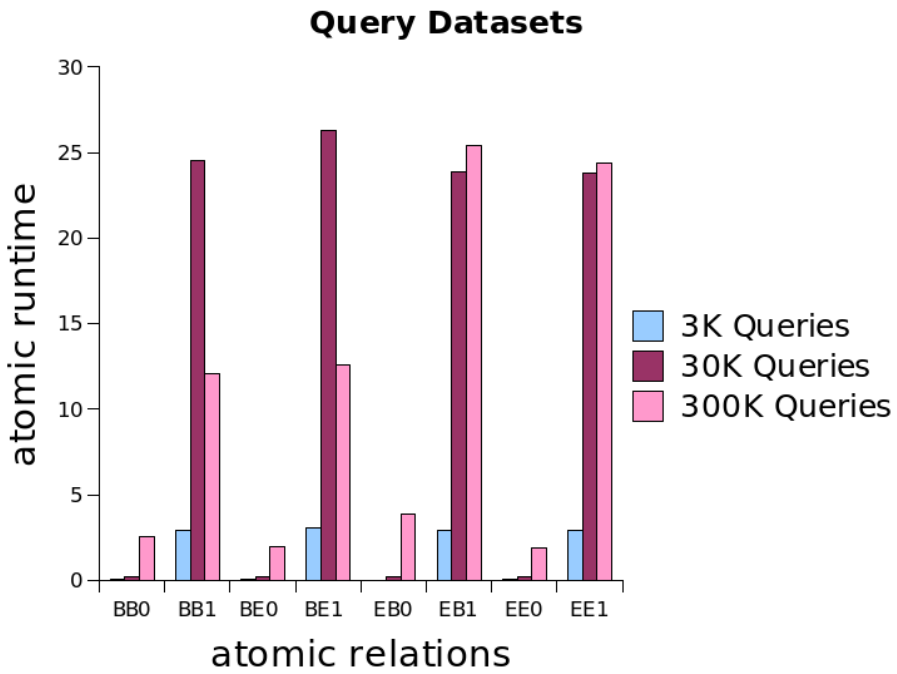
- State-of-the-art:
  - SILK extended to deal with spatio-temporal data
  - Baseline for *eq* using brute-force

- Datasets: $S = T$

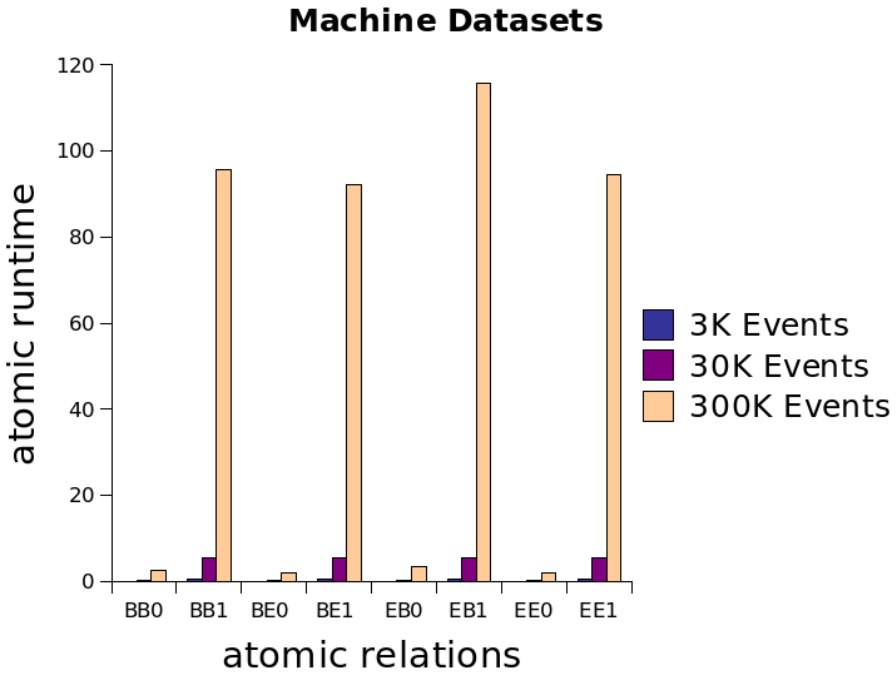
Log Type	Dataset name	Size	Unique $b(s)$	Unique $e(s)$
Machinery	3KMachines	3,154	960	960
	30KMachines	28,869	960	960
	300KMachines	288,690	960	960
Query	3KQueries	3,888	3,636	3,638
	30KQueries	30,635	3,070	3,070
	300KQueries	303,991	184	184

- State-of-the-art:
  - SILK extended to deal with spatio-temporal data
  - Baseline for *eq* using brute-force
- Evaluation measures:
  - atomic runtime of each of the atomic relations
  - relation runtime required to compute each Allen's relation
  - total runtime required to compute all 13 relations

- Q_1: Does the reduction of Allen relations to 8 atomic relations influence the overall runtime of the approach?



- Q_1: Does the reduction of Allen relations to 8 atomic relations influence the overall runtime of the approach?



- Q_2: How does AEGLE perform when compared with the state of the art in terms of time efficiency?

Log Type	Dataset Name	Total Runtime		
		AEGLE	AEGLE *	SILK
Machine	3KMachines	11.26	5.51	294.00
	30KMachines	1,016.21	437.79	29,846.00
	300KMachines	189,442.16	78,416.61	NA
Query	3KQueries	26.94	17.91	541.00
	30KQueries	**988.78**	463.27	**33,502.00**
	300KQueries	**211,996.88**	86,884.98	**NA**

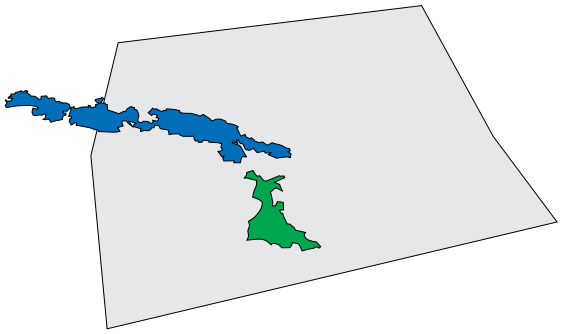
		Machine			Query		
Relation	Approach	3KMachines	30KMachines	300KMachines	3KQueries	30KQueries	300KQueries
m	AEGLE	0.02	**0.19**	3.42	0.02	0.21	3.89
	SILK	23.00	**2,219.00**	NA	41.00	2,466.00	NA
eq	AEGLE	0.05	0.79	49.84	0.05	**0.45**	348.51
	SILK	23.00	2,250.00	NA	41.00	**2,473.00**	NA
	baseline	2.05	171.10	23,436.30	3.15	**196.09**	31,452.54
ovi	AEGLE	3.16	222.27	38,226.32	11.97	**257.59**	**42,121.68**
	SILK	22.00	2,189.00	NA	42.00	**2,503.00**	**NA**

Conclusion

- AEGLE = efficient temporal linking
  - Reduction of 13 Allen Interval relations to 8 atomic relations
  - Efficiency: simple sorting with complexity $O(n \log n)$
  - Scalable LD

Table of Contents

- Large Geo-spatial datasets
  - LinkedGeoData contains $> 20+$ billion triples
  - NUTS contains up to $1,500$ points per resources
- Only 7.1% of the links between resources connect geo-spatial entities (Ngonga Ngomo, 2013)
- Geo-spatial resources available on the LOD
  - Described using polygons
  - Large in number
  - Demands the computation of topological relations
- Naïve computation of topological relations is impracticable for geo-spatial resources

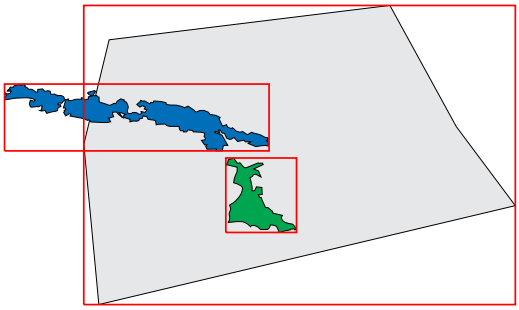


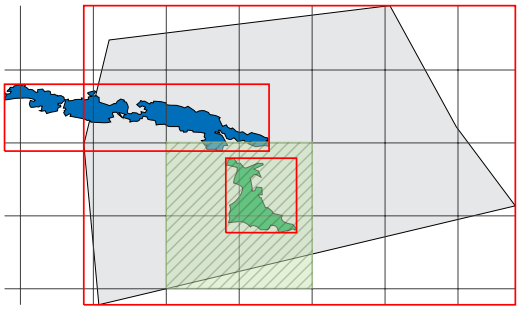
- Standard to describe the topological relations in 2D space.
- DE-9IM is to based on the intersection matrix:

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

- There must be at least one shared point for a relation to be hold
- Except for the `disjoint` relation $\Rightarrow$ inverse of the `intersects` relation
- Accelerating the computation of whether two geometries share at least one point, accelerates the computation of any topological relation

- RADON implements improved indexing approach based on
  1. Minimum bounding boxes (MBB)
  2. Space tiling

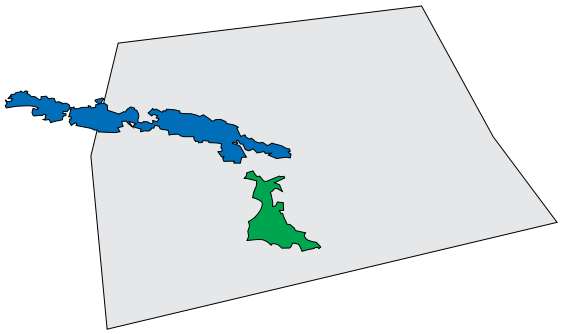




- Large geometries that span over a large number of hypercubes ⇒ large spatial index when used as S
- *Estimated Total Hypervolume* (ETH) of a set of geometries X

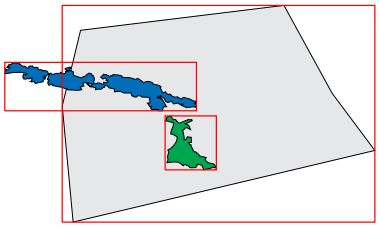
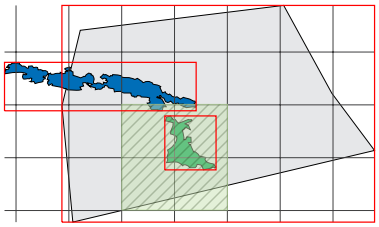
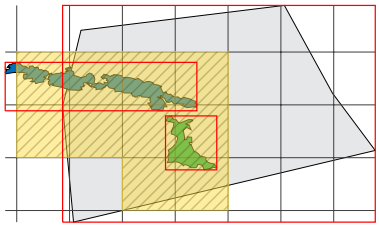
$$ETH(X) = |X| \prod_{i=1}^{d} \frac{1}{|X|} \sum_{x \in X} \left(\max_{p \in x}\{\kappa_i(p)\} - \min_{p \in x}\{\kappa_i(p)\} \right)$$

- If $ETH(S) > ETH(T)$, swaps S and T and computes the reverse relation r' instead of r
- e.g. if r is `covered` and $ETH(S) > ETH(T)$, then swaps S and T and computes `coveredBy`

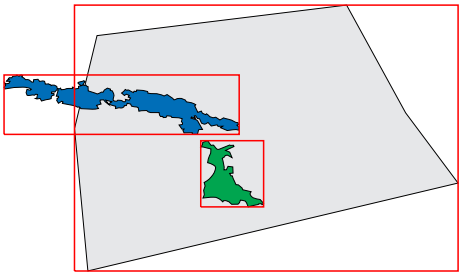


Since $ETH(NUTS) > ETH(CLC)$, then S = CLC and T = NUTS

- Insert all geometries $s \in S$ into index $I(s)$
  1. Computes $MBB(s)$
  2. Maps each s to all hypercubes over $MBB(s)$ spans
- Same procedure for all $t \in T$ but only index geometries t that are potentially in hypercubes already contained in $I(S)$

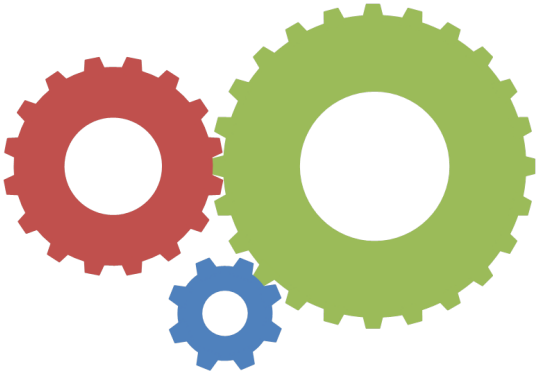




- Discards unnecessary computations using the TESTMBB procedure
- TESTMBB optimizes the subset of DE-9IM relations for relations where
  - one geometry has interior or boundary points in the exterior of the other geometry
  - e.g. `equals`, `covers` and `within`
- For other relations, TESTMBB returns *true*
- If TESTMBB returns `false`
  - No need to compute the expensive computation of the topological relation

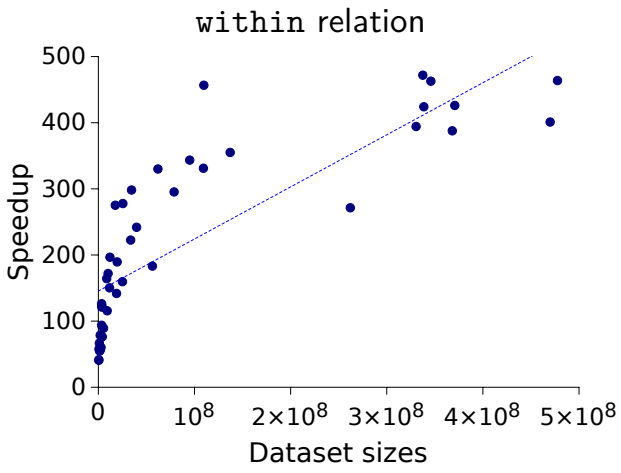


- TESTMBB(`within`, `blue`) = `false`
- TESTMBB(`within`, `green`) = `true`

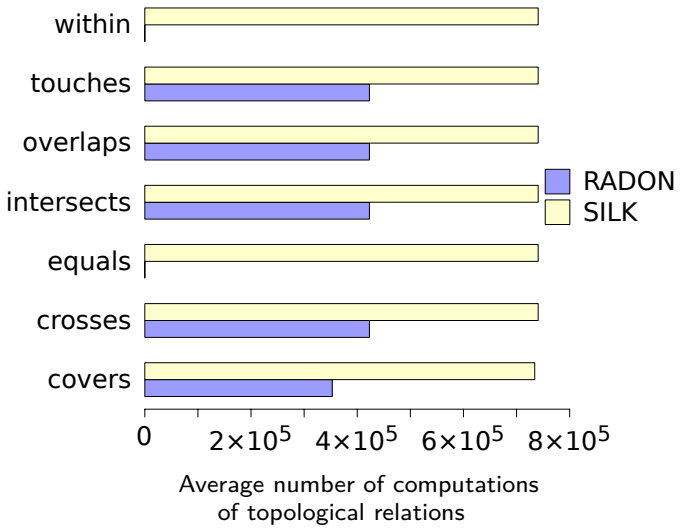
- Topological relations
  - Subset of the 7 topological relations
  - i.e. `within`, `touches`, `overlaps`, `intersects`, `equals`, `crosses` and `covers`
- Hardware
  - 64-core 2.3 GHz, OpenJDK 64-Bit Server
  - 20 GB RAM with timeout limit of 2 hours
- State of the art
  1. SILK
  2. STRABON
- Datasets
  1. NUTS
  2. CORINE Land Cover (CLC)



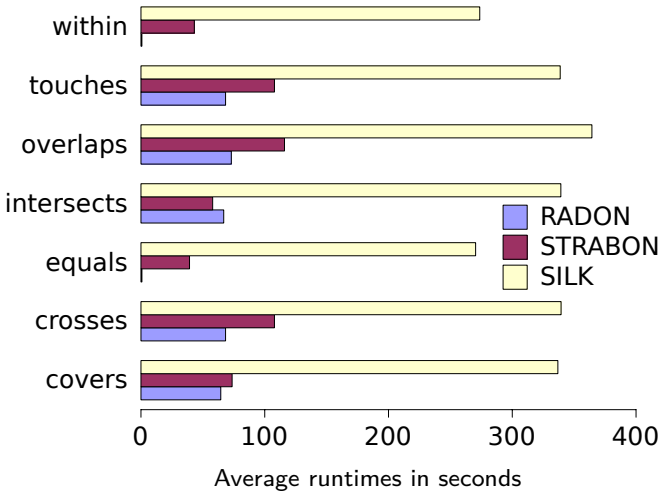
- RADON vs. SILK
- 44 subsets of the CLC vs. the full NUTS
- 7 basic topological relations
- 308 experiments
- Single core
- RADON achieves a linear speedup relative to the dataset sizes
- Up to 450 times faster for the `within` relation



- Same setting as in previous experiments
- RADON runs significantly less computations of the relations
- RADON carries out only 3 and 4 computations for the `equals` and `within` relations respectively
- On average, 449 times less computations per relation



- Same setting as in previous experiments
- On average, RADON is faster then
  - SILK by 65.62 times
  - STRABON by 11.99 times
- STRABON outperforms RADON on the `intersects` relation
  - STRABON uses an **R-tree-over-GiST** spatial index over the stored geometries in the underlying **PostGIS** database
  - **R-tree-over-GiST** is highly optimized for the retrieval of spatially connected objects



- Merge all 44 sub-datasets of CLC (CLC_M)
- CLC_M contains $2,209,538$ resource
- CLC_M as both source and target datasets
- Simple round robin load balancing policy
- On average, within the 2 hours time limit
  - RADON finishes in 20.83 minutes
  - SILK finalizes 1.16% of the tasks
  - SILK would need 4.36 days with 8 threads (linear extrapolation)
  - RADON is 834.69 times faster than Silk

Relation	#Thr.	RADON	SILK	Speedup
equals	1	24.11	36,500 (0.33%)	1,513.58
	2	13.15	21,667 (0.55%)	1,647.58
	4	6.81	11,750 (1.02%)	1,725.77
	8	3.79	6,286 (1.91%)	1,658.78
intersects	1	93.17	37,500 (0.32%)	402.50
	2	49.03	20,667 (0.58%)	421.53
	4	25.11	12,000 (1.00%)	477.81
	8	13.04	6,300 (1.90%)	483.24
within	1	36.47	35,000 (0.34%)	959.74
	2	18.26	20,667 (0.58%)	1,131.86
	4	9.44	11,765 (1.02%)	1,246.34
	8	5.92	6,202 (1.93%)	1,048.34
covers	1	35.62	36,000 (0.33%)	1,010.75
	2	18.51	21,029 (0.57%)	1,136.10
	4	10.23	12,000 (1.00%)	1,172.50
	8	5.33	6,300 (1.90%)	1,182.13
touches	1	94.50	35,500 (0.34%)	375.68
	2	47.71	22,196 (0.54%)	465.18
	4	25.09	12,121 (0.99%)	483.08
	8	13.30	6,381 (1.88%)	479.75

- STRABON does not finish any of the experiments within the 2-hours time limit
- No progress feedback from STRABON
- STRABON performance estimation
  - STRABON vs. sub-CLC dataset 10 times $< CLC_m$
  - Optimistic (STRABON scales linearly), average speedup of 24
  - Realistic (STRABON scales in $O(n^2)$), average speedup of 241

Relation	#Thr.	RADON	SILK	Speedup
equals	1	24.11	36,500 (0.33%)	1,513.58
	2	13.15	21,667 (0.55%)	1,647.58
	4	6.81	11,750 (1.02%)	1,725.77
	8	3.79	6,286 (1.91%)	1,658.78
intersects	1	93.17	37,500 (0.32%)	402.50
	2	49.03	20,667 (0.58%)	421.53
	4	25.11	12,000 (1.00%)	477.81
	8	13.04	6,300 (1.90%)	483.24
within	1	36.47	35,000 (0.34%)	959.74
	2	18.26	20,667 (0.58%)	1,131.86
	4	9.44	11,765 (1.02%)	1,246.34
	8	5.92	6,202 (1.93%)	1,048.34
covers	1	35.62	36,000 (0.33%)	1,010.75
	2	18.51	21,029 (0.57%)	1,136.10
	4	10.23	12,000 (1.00%)	1,172.50
	8	5.33	6,300 (1.90%)	1,182.13
touches	1	94.50	35,500 (0.34%)	375.68
	2	47.71	22,196 (0.54%)	465.18
	4	25.09	12,121 (0.99%)	483.08
	8	13.30	6,381 (1.88%)	479.75

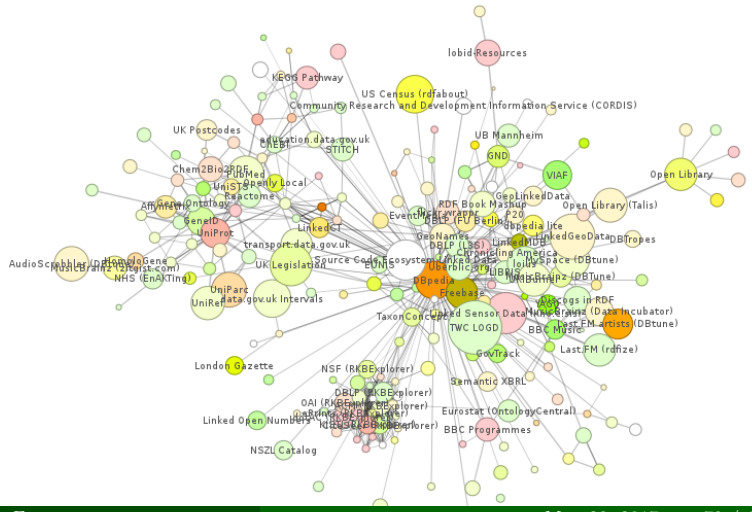
- Conclusion
  - Presented RADON, an approach for rapid discovery of topological relations among geo-spatial resources
  - Sparse index combines space tiling and MBB approximation
  - RADON is complete and correct
  - Outperforms the SOTA by up to 3 orders of magnitude
- Future work
  - More sophisticated load balancing approaches
  - Other topology approximation methods
  - Topological relations in higher dimensions

Table of Contents

What if ...

- Data does not fit memory C, i.e.,
- $|S| + |T| > |C|$

1. Common problem
2. Growing size and number of datasets
   - $\approx 150 \times 10^9$ triples
   - $\approx 10 \times 10^3$ datasets
   - Largest dataset with $> 20 \times 10^9$ *triples*
3. Mostly in-memory solutions

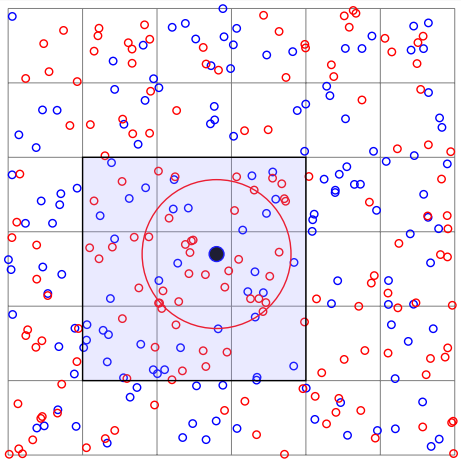


Insight

- Most approaches rely on divide-and-merge paradigm
- Example: HR^3
  - $\sigma(s,t) \geq \theta \Leftrightarrow \delta(s,t) \leq \Delta$

Insight

- Most approaches rely on divide-and-merge paradigm
- Example: HR^3
  - $\sigma(s,t) \geq \theta \Leftrightarrow \delta(s,t) \leq \Delta$

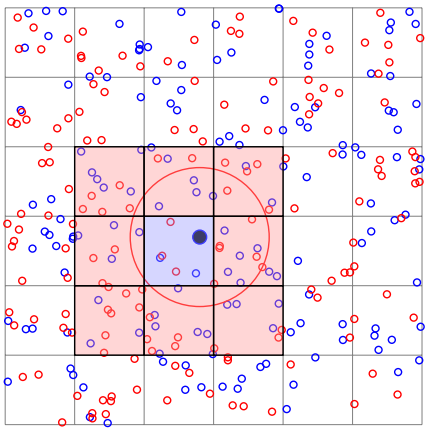


1. Define $\mathcal{S} = \{S_1, \ldots, S_n\}$ with $S_i \subseteq S \wedge \bigcup_i S_i = S$

1. Define $\mathcal{S} = \{S_1, \ldots, S_n\}$ with $S_i \subseteq S \wedge \bigcup_i S_i = S$
2. Define $\mathcal{T} = \{T_1, \ldots, S_m\}$ with $T_j \subseteq T \wedge \bigcup_j T_j = T$

1. Define $\mathcal{S} = \{S_1, \ldots, S_n\}$ with $S_i \subseteq S \wedge \bigcup_i S_i = S$
2. Define $\mathcal{T} = \{T_1, \ldots, S_m\}$ with $T_j \subseteq T \wedge \bigcup_j T_j = T$
3. Find mapping function $\mu : \mathcal{S} \rightarrow 2^{\mathcal{T}}$ with
   - elements of S_i only compared with elements of sets in $\mu(S_i)$
   - union of results over all $S_i \in \mathcal{S}$ is exactly M'.

1. Define $\mathcal{S} = \{S_1, \ldots, S_n\}$ with $S_i \subseteq S \wedge \bigcup_i S_i = S$
2. Define $\mathcal{T} = \{T_1, \ldots, S_m\}$ with $T_j \subseteq T \wedge \bigcup_j T_j = T$
3. Find mapping function $\mu : \mathcal{S} \rightarrow 2^{\mathcal{T}}$ with
   - elements of S_i only compared with elements of sets in $\mu(S_i)$
   - union of results over all $S_i \in \mathcal{S}$ is exactly M'.

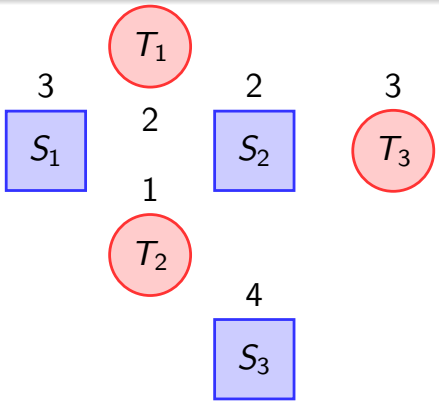


Definition

- A task E_{ij} stands for comparing S_i with $T_j \in \mu(S_i)$

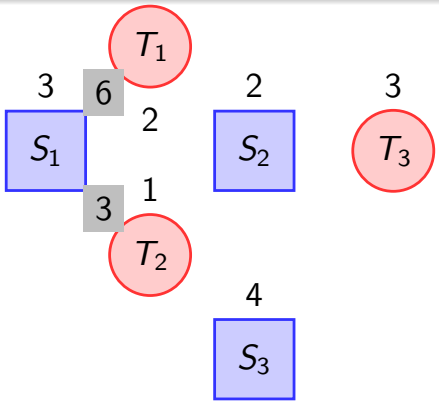
Definition

- A task E_{ij} stands for comparing S_i with $T_j \in \mu(S_i)$
- Task Graph $G = (V, E, w_v, w_e)$, with
  - $V = \mathcal{S} \cup \mathcal{T}$
  - $w_v(v) = |V|$
  - $w_e(e_{ij}) = |S_i||T_j|$



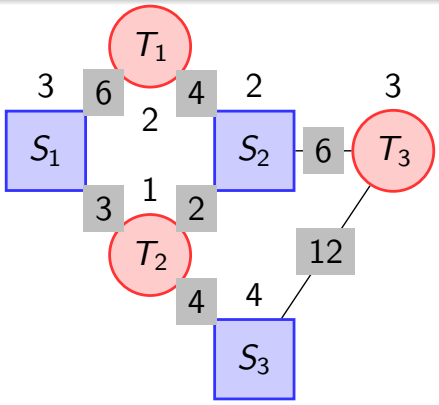
Definition

- A task E_{ij} stands for comparing S_i with $T_j \in \mu(S_i)$
- Task Graph $G = (V, E, w_v, w_e)$, with
  - $V = \mathcal{S} \cup \mathcal{T}$
  - $w_v(v) = |V|$
  - $w_e(e_{ij}) = |S_i||T_j|$



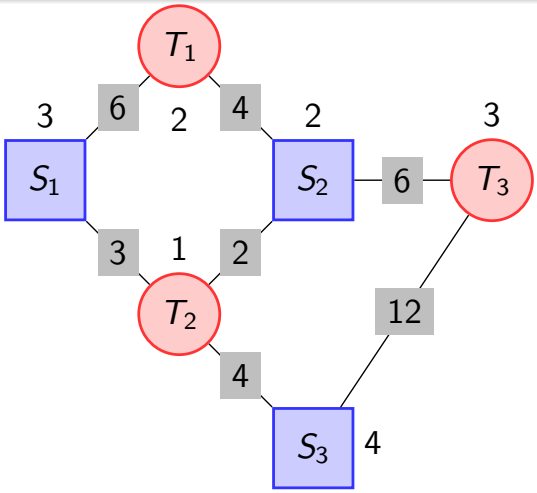
Definition

- A task E_{ij} stands for comparing S_i with $T_j \in \mu(S_i)$
- Task Graph $G = (V, E, w_v, w_e)$, with
  - $V = \mathcal{S} \cup \mathcal{T}$
  - $w_v(v) = |V|$
  - $w_e(e_{ij}) = |S_i||T_j|$



Locality maximization

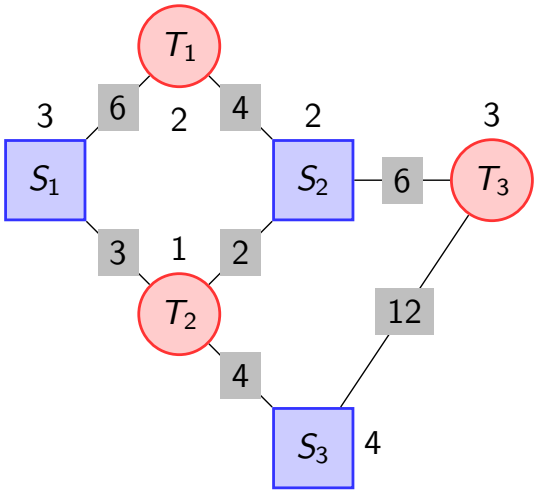
- **Two-step approach**:
  1. **Clustering**: Find groups of nodes that fit in memory and
  2. **Scheduling**: Compute sequence of groups that minimizes hard drive access



Naïve Approach

- Cluster by S_i

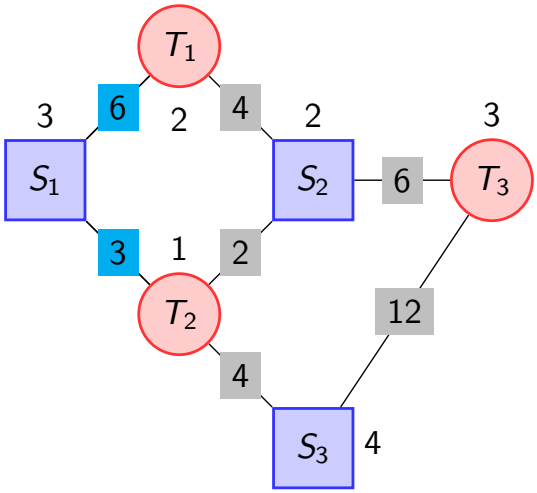
- Example: $|C| = 7$



Naïve Approach

- Cluster by S_i

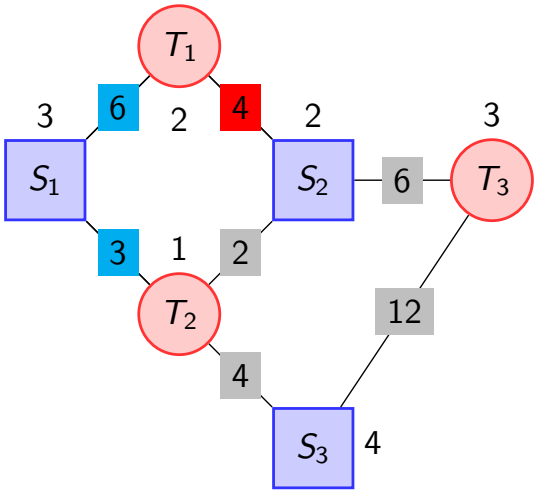
- Example: $|C| = 7$



Naïve Approach

- Cluster by S_i

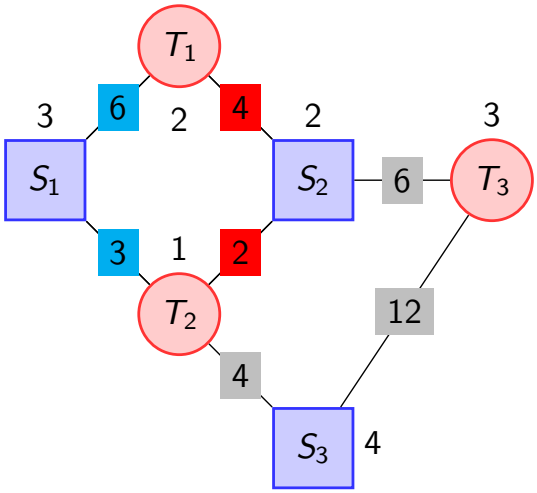
- Example: $|C| = 7$



Naïve Approach

- Cluster by S_i

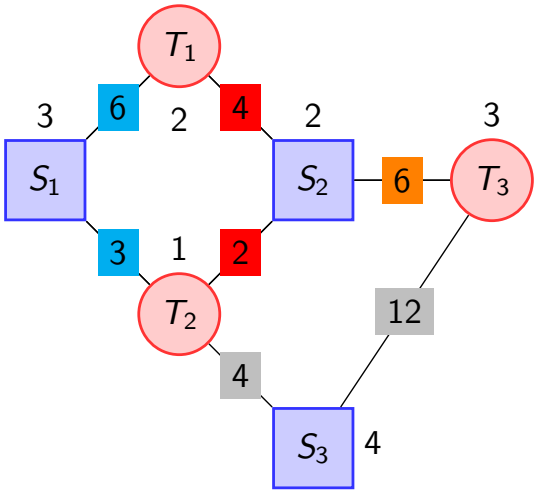
- Example: $|C| = 7$



Naïve Approach

- Cluster by S_i

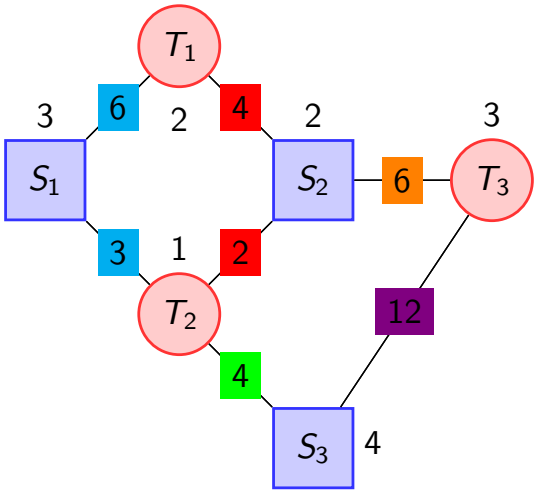
- Example: $|C| = 7$



Naïve Approach

- Cluster by S_i

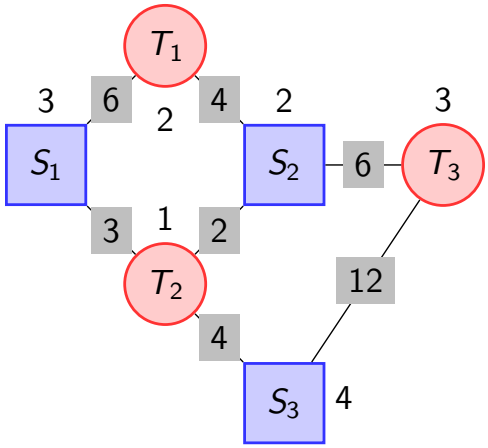
- Example: $|C| = 7$



Greedy Approach

- Start by largest task
- Add connected largest tasks until none fits in C

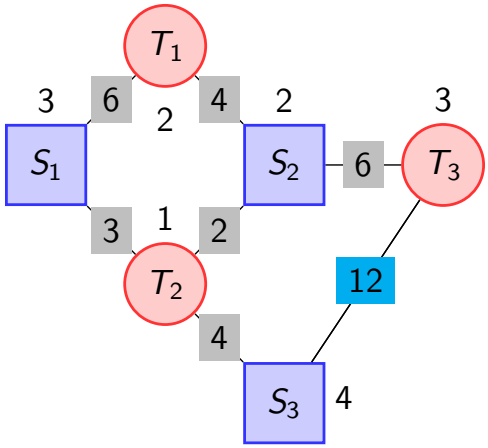
- Example: $|C| = 7$



Greedy Approach

- Start by largest task
- Add connected largest tasks until none fits in C

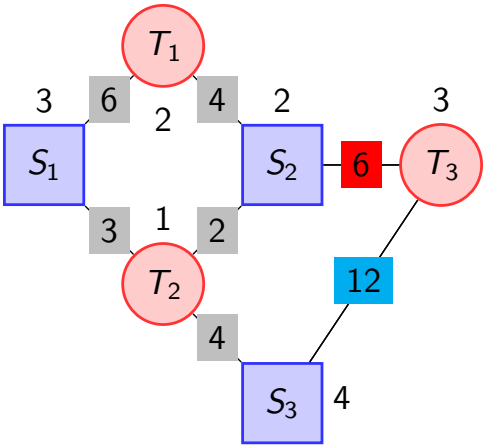
- Example: $|C| = 7$



Greedy Approach

- Start by largest task
- Add connected largest tasks until none fits in C

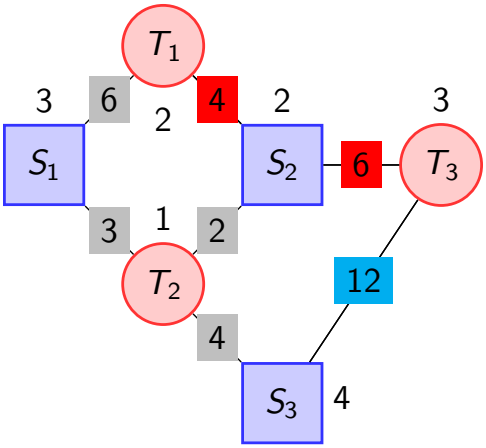
- Example: $|C| = 7$



Greedy Approach

- Start by largest task
- Add connected largest tasks until none fits in C

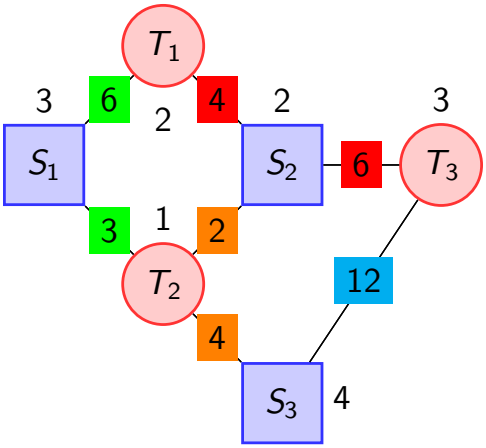
- Example: $|C| = 7$



Greedy Approach

- Start by largest task
- Add connected largest tasks until none fits in C

- Example: $|C| = 7$



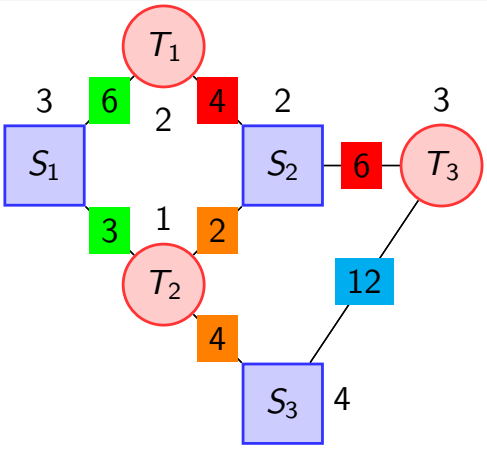
Insights

- **Output of clustering**: Sequence $G_1, \ldots, G_N$ of clusters
- **Intuition**: Consecutive clusters should share data
- **Goal**: Maximize overlap of generated sequence

Insights

- **Output of clustering**: Sequence $G_1, \ldots, G_N$ of clusters
- **Intuition**: Consecutive clusters should share data
- **Goal**: Maximize overlap of generated sequence

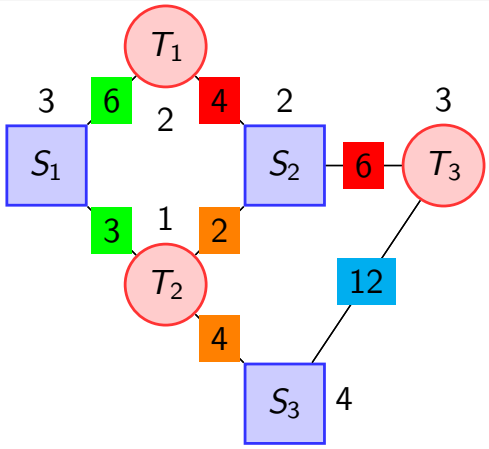
- Overlap $o(G_i, G_j) = \sum_{v \in V(G_i) \cap V(G_j)} |v|$
- o(G_4, G_1) = 4



Insights

- **Output of clustering**: Sequence $G_1, \ldots, G_N$ of clusters
- **Intuition**: Consecutive clusters should share data
- **Goal**: Maximize overlap of generated sequence

- Overlap $o(G_i, G_j) = \sum_{v \in V(G_i) \cap V(G_j)} |v|$
- o(G_4, G_1) = 4
- Overlap $o(G_1, \ldots, G_N) = \sum_{i=1}^{N-1} o(G_i, G_{i+1})$
- o(G_4, G_1, G_2, G_1) = 9



Best-Effort

- Select random pair of clusters
- If permutation improves overlap, then permute
- Relies on local knowledge for scalability

Best-Effort

- Select random pair of clusters
- If permutation improves overlap, then permute
- Relies on local knowledge for scalability

- Trick:

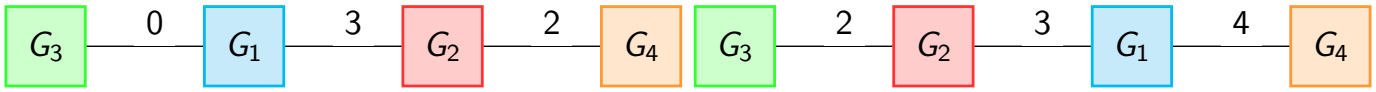
$$\Delta(G_i, G_j) = (o(G_{i-1}, G_j) + o(G_j, G_{i+1}) + o(G_{j-1}, G_i) + o(G_i, G_{j+1})) - (o(G_{i-1}, G_i) + o(G_i, G_{i+1}) + o(G_{j-1}, G_j) + o(G_j, G_{j+1})) \quad (1)$$

Greedy

- Start with random cluster
- Choose next cluster with largest overlap
- Global knowledge needed

Greedy

- Start with random cluster
- Choose next cluster with largest overlap
- Global knowledge needed



- **Datasets**
  1. DBP: 1 million labels from DBpedia version 04-2015
  2. LGD: 0.8 million places from LinkedGeoData
- **Hardware**
  1. Intel Xeon E5-2650 v3 processors (2.30GHz)
  2. Ubuntu 14.04.3 LTS
  3. 10GB RAM
- **Measures**
  1. Total runtime
  2. Hit ratio

- Only show results of LGD
- Results on DBP lead to similar insights

	Runtimes		Hit Ratio	
$\|C\|$	*Naive*	*Greedy*	*Naive*	*Greedy*
100	568.0	646.3	0.57	0.77
200	518.3	594.0	0.66	0.80
400	532.0	593.3	0.67	0.80
1,000	5,974.0	118,454.7	0.51	0.64
2,000	6,168.0	115,450.0	0.51	0.63
4,000	7,118.3	121,901.7	0.50	0.63

- Only show results of LGD
- Results on DBP lead to similar insights

	Runtimes		Hit Ratio	
$\|C\|$	*Naive*	*Greedy*	*Naive*	*Greedy*
100	568.0	646.3	0.57	0.77
200	518.3	594.0	0.66	0.80
400	532.0	593.3	0.67	0.80
1,000	5,974.0	118,454.7	0.51	0.64
2,000	6,168.0	115,450.0	0.51	0.63
4,000	7,118.3	121,901.7	0.50	0.63

Conclusion

1. Naïve approach is more efficient
2. Greedy approach is more effective
3. Select naïve approach for clustering

- Only show results of LGD
- Results on DBP lead to similar insights

	Runtimes (ms)		Hit ratio	
$\|C\|$	*Best-Effort*	*Greedy*	*Best-Effort*	*Greedy*
100	571.3	1,599.3	0.56	0.68
200	565.7	1,448.3	0.66	0.85
400	581.0	1,379.3	0.67	0.88
1,000	5,666.0	814,271.7	0.51	0.86
2,000	6,268.0	810,855.0	0.51	0.86
4,000	6,675.7	814,041.7	0.50	0.86

- Only show results of LGD
- Results on DBP lead to similar insights

	Runtimes (ms)		Hit ratio	
$\|C\|$	*Best-Effort*	*Greedy*	*Best-Effort*	*Greedy*
100	571.3	1,599.3	0.56	0.68
200	565.7	1,448.3	0.66	0.85
400	581.0	1,379.3	0.67	0.88
1,000	5,666.0	814,271.7	0.51	0.86
2,000	6,268.0	810,855.0	0.51	0.86
4,000	6,675.7	814,041.7	0.50	0.86

Conclusion

1. Best-effort approach more time-efficient
2. Greedy more effective
3. Best-effort approach is to be used for scheduling

	Runtimes (ms)					
$\|C\|$	GNOME	*FIFO*	*F2*	*LFU*	*LRU*	*SLRU*
1,000	**5,974.0**	37,161.0	42,090.3	45,906.7	54,194.3	56,904.3
2,000	**6,168.0**	31,977.0	39,071.3	39,872.0	45,473.0	46,795.0
4,000	**7,118.3**	21,337.0	40,860.0	28,028.3	26,816.7	27,200.0
	Hit ratio					
1,000	**0.51**	0.17	0.16	0.19	0.17	0.17
2,000	**0.51**	0.29	0.30	0.32	0.30	0.30
4,000	0.51	0.54	0.55	**0.59**	0.55	0.56

	Runtimes (ms)					
$\|C\|$	GNOME	*FIFO*	*F2*	*LFU*	*LRU*	*SLRU*
1,000	**5,974.0**	37,161.0	42,090.3	45,906.7	54,194.3	56,904.3
2,000	**6,168.0**	31,977.0	39,071.3	39,872.0	45,473.0	46,795.0
4,000	**7,118.3**	21,337.0	40,860.0	28,028.3	26,816.7	27,200.0
	Hit ratio					
1,000	**0.51**	0.17	0.16	0.19	0.17	0.17
2,000	**0.51**	0.29	0.30	0.32	0.30	0.30
4,000	0.51	0.54	0.55	**0.59**	0.55	0.56

Conclusion

1. GNOME is more time-efficient
2. Leads to higher hit rates in most cases

	100k	*200k*	*400k*	*800k*
LGD	362,141.3	1,452,922.0	5,934,038.7	20,001,965.7
DBP	434,630.7	1,790,350.7	6,677,923.0	12,653,403.3

Conclusion

- Sub-quadratic growth of runtime
- Runtime grows linearly with number of mappings
- For LGD, 360 – 370 mappings/s

Table of Contents

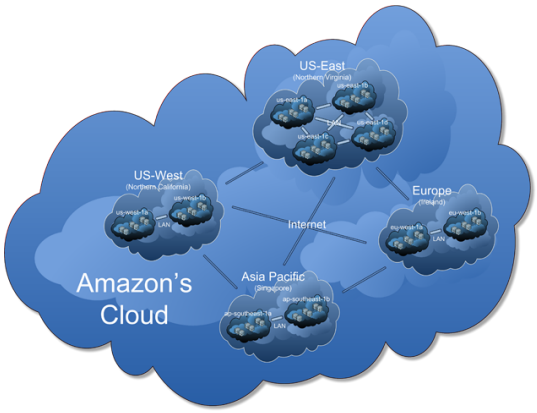
1. **Devise better algorithms**

- Blocking
- Algorithms for given metrics
  - PPJoin+, EDJoin
  - $\mathcal{HR}^3$

1. **Devise better algorithms**

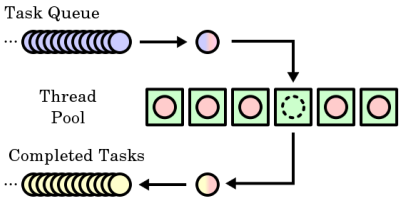
- Blocking
- Algorithms for given metrics
  - PPJoin+, EDJoin
  - $\mathcal{HR}^3$

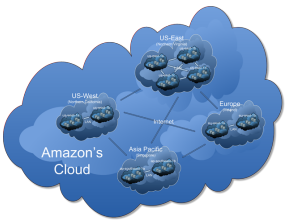
2. **Use parallel hardware**



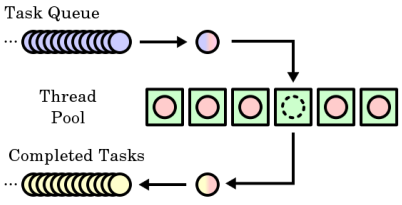
- Threads pools
- MapReduce
- Massively Parallel Hardware

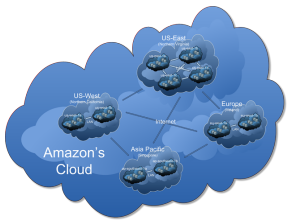
- Different architectures
  - Memory (shared, hybrid, distributed)
  - Execution paths (different, same)
  - Location (remote, local)





- Different architectures
  - Memory (shared, hybrid, distributed)
  - Execution paths (different, same)
  - Location (remote, local)





Question

- When should which use which hardware?

Premises

- Given an algorithm that runs on all three architectures ...

Premises

- Given an algorithm that runs on all three architectures ...
- **Note to self**: Implement one
  - Picked $\mathcal{HR}^3$
  - Reduction-ratio-optimal

Premises

- Given an algorithm that runs on all three architectures ...
- **Note to self**: Implement one
  - Picked $\mathcal{HR}^3$
  - Reduction-ratio-optimal

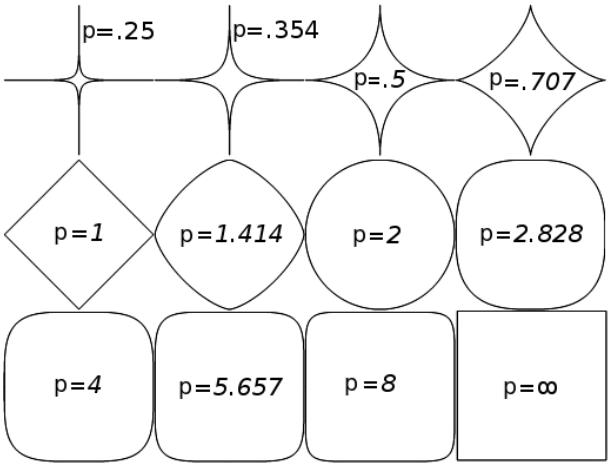
Reach for the Cloud?

Goals

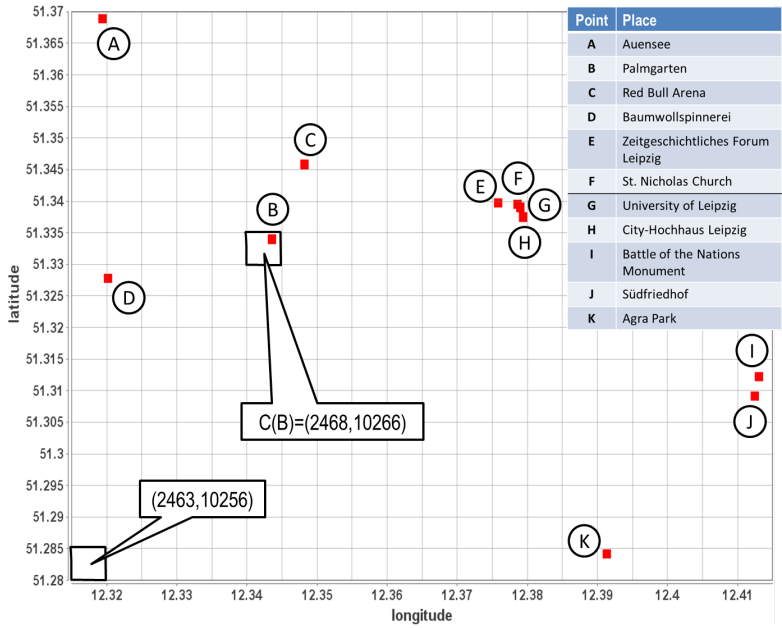
- Compare runtimes on all three parallel architectures
- Find break-even points

1. Assume spaces with Minkowski metric and $p \geq 2$

$$\delta(s,t) = \sqrt[p]{\sum_{i=1}^{n} |s_i - t_i|^p}$$

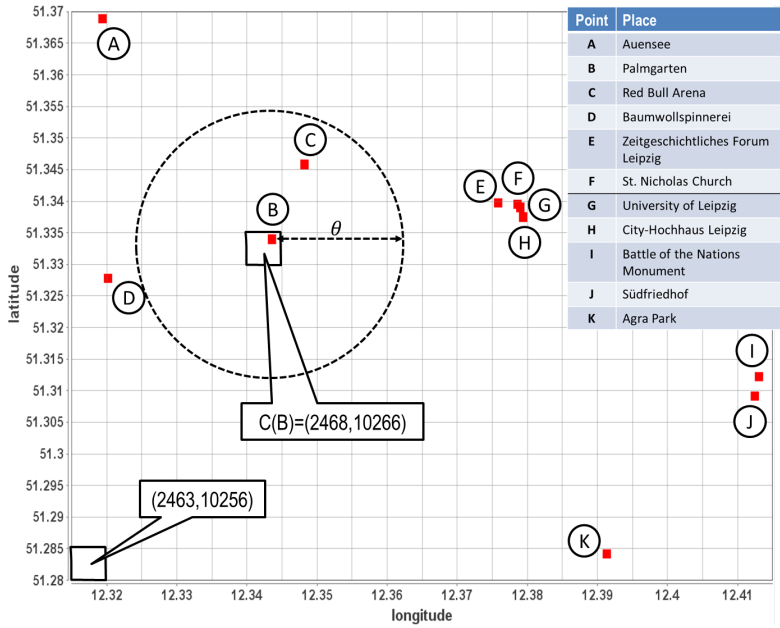


2. Create grid of width $\Delta = \theta/\alpha$



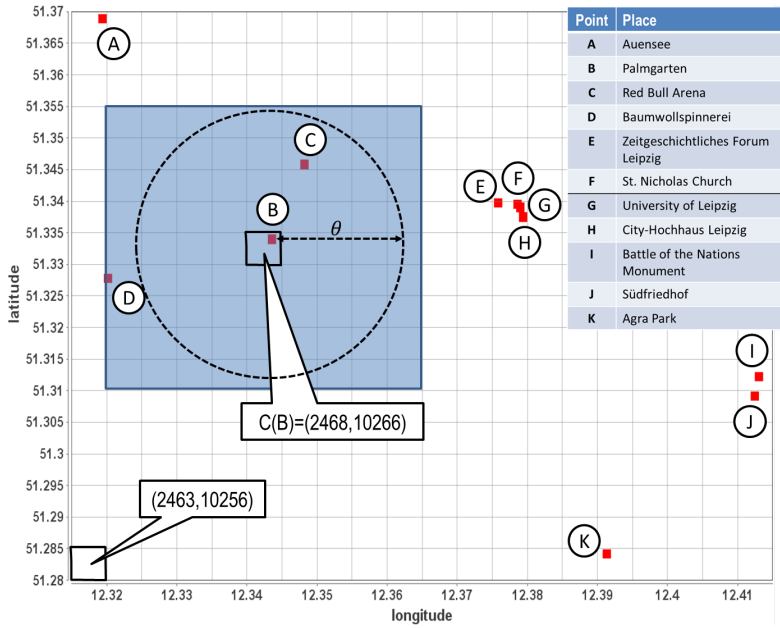
Point	Place	Longitude	Latitude	C(Point)	
A	Auensee	12.3194	51.3688	(2463,10273)	Π_0
B	Palmgarten	12.3436	51.3339	(2468,10226)	
C	Red Bull Arena	12.3482	51.3457	(2469,10269)	
D	Baumwollspinnerei	12.3201	51.3277	(2464,10265)	
E	Zeitgeschichtliches Forum Leipzig	12.3758	51.3397	(2475,10267)	
F	St. Nicholas Church	12.3786	51.3395	(2475,10267)	
G	University of Leipzig	12.3790	51.3390	(2475,10267)	Π_1
H	City-Hochhaus Leipzig	12.3794	51.3375	(2475,10267)	
I	Battle of the Nations Monument	12.4130	51.3122	(2482,10262)	
J	Südfriedhof	12.4125	51.3091	(2482,10261)	
K	Agra Park	12.3913	51.2841	(2478,10256)	

3. Link discovery condition describes a hypersphere



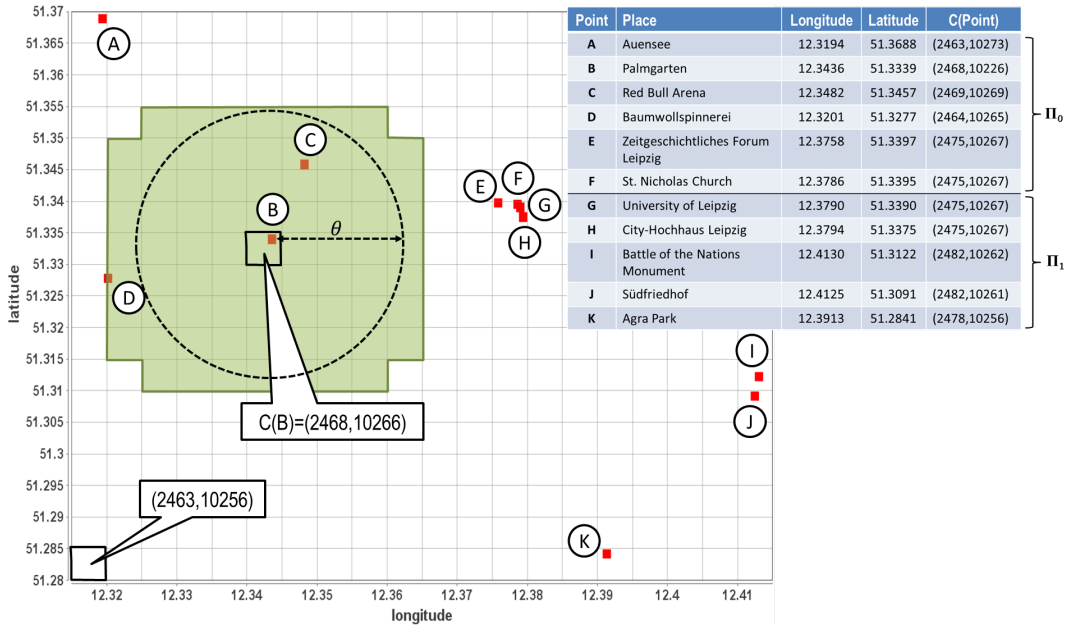
Point	Place	Longitude	Latitude	C(Point)	
A	Auensee	12.3194	51.3688	(2463,10273)	Π_0
B	Palmgarten	12.3436	51.3339	(2468,10226)	
C	Red Bull Arena	12.3482	51.3457	(2469,10269)	
D	Baumwollspinnerei	12.3201	51.3277	(2464,10265)	
E	Zeitgeschichtliches Forum Leipzig	12.3758	51.3397	(2475,10267)	
F	St. Nicholas Church	12.3786	51.3395	(2475,10267)	
G	University of Leipzig	12.3790	51.3390	(2475,10267)	Π_1
H	City-Hochhaus Leipzig	12.3794	51.3375	(2475,10267)	
I	Battle of the Nations Monument	12.4130	51.3122	(2482,10262)	
J	Südfriedhof	12.4125	51.3091	(2482,10261)	
K	Agra Park	12.3913	51.2841	(2478,10256)	

4. Approximate hypersphere with hypercube



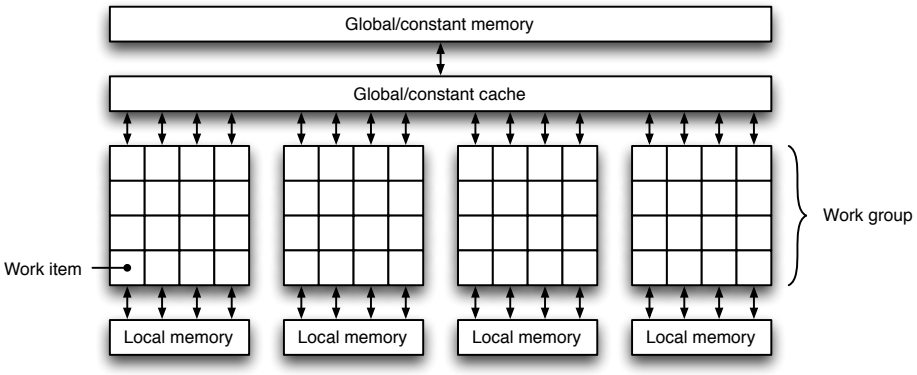
Point	Place	Longitude	Latitude	C(Point)	
A	Auensee	12.3194	51.3688	(2463,10273)	Π_0
B	Palmgarten	12.3436	51.3339	(2468,10226)	
C	Red Bull Arena	12.3482	51.3457	(2469,10269)	
D	Baumwollspinnerei	12.3201	51.3277	(2464,10265)	
E	Zeitgeschichtliches Forum Leipzig	12.3758	51.3397	(2475,10267)	
F	St. Nicholas Church	12.3786	51.3395	(2475,10267)	
G	University of Leipzig	12.3790	51.3390	(2475,10267)	Π_1
H	City-Hochhaus Leipzig	12.3794	51.3375	(2475,10267)	
I	Battle of the Nations Monument	12.4130	51.3122	(2482,10262)	
J	Südfriedhof	12.4125	51.3091	(2482,10261)	
K	Agra Park	12.3913	51.2841	(2478,10256)	

- Use index to discard portions of hypercube



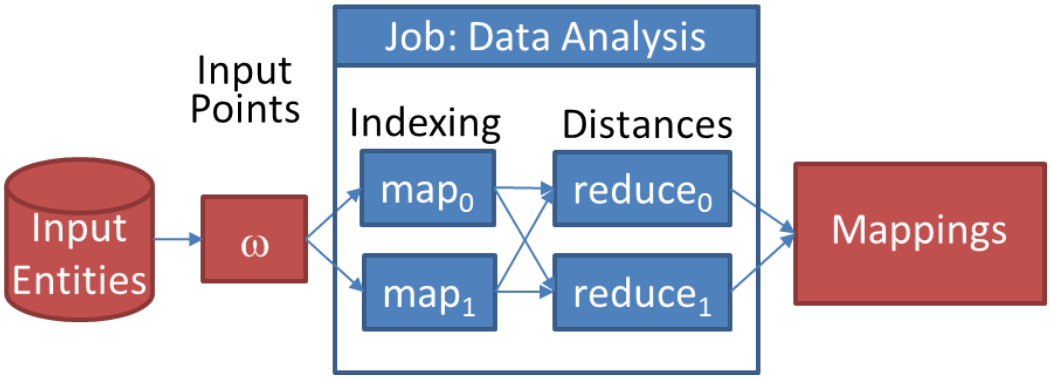
Point	Place	Longitude	Latitude	C(Point)	
A	Auensee	12.3194	51.3688	(2463,10273)	Π_0
B	Palmgarten	12.3436	51.3339	(2468,10226)	
C	Red Bull Arena	12.3482	51.3457	(2469,10269)	
D	Baumwollspinnerei	12.3201	51.3277	(2464,10265)	
E	Zeitgeschichtliches Forum Leipzig	12.3758	51.3397	(2475,10267)	
F	St. Nicholas Church	12.3786	51.3395	(2475,10267)	
G	University of Leipzig	12.3790	51.3390	(2475,10267)	Π_1
H	City-Hochhaus Leipzig	12.3794	51.3375	(2475,10267)	
I	Battle of the Nations Monument	12.4130	51.3122	(2482,10262)	
J	Südfriedhof	12.4125	51.3091	(2482,10261)	
K	Agra Park	12.3913	51.2841	(2478,10256)	

- Large number of simple compute cores
- Same instruction, multiple data
- Bottleneck: PCI Express Bus
  1. Run discretization on CPU
  2. Run indexing on GPU
  3. Run comparisons on CPU

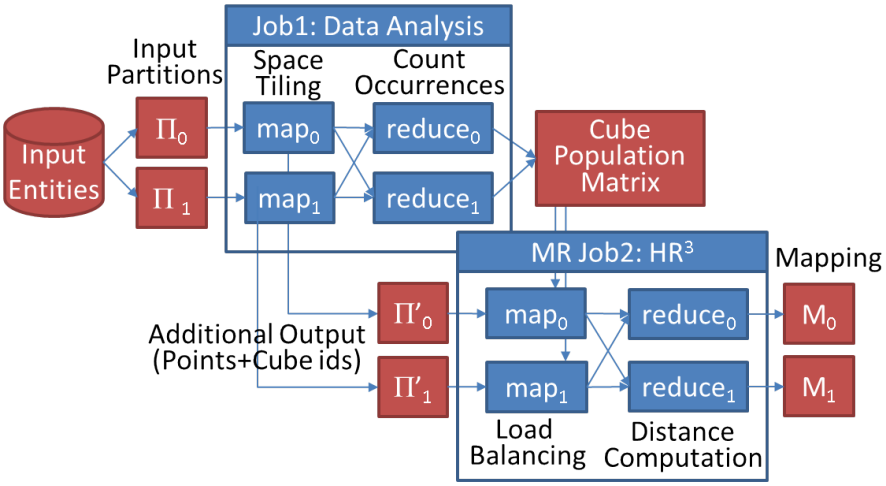


- **Naive Approach**

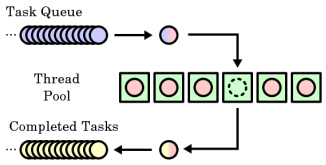
- **Naive Approach**
  1. Rely on Map Reduce paradigm
  2. Run discretization and assignment to cubes in map step
  3. Run distance computation in reduce step

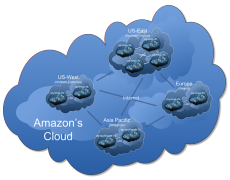


- **Load Balancing**
  1. Run two jobs
  2. Job1: Compute cube population matrix
  3. Job2: Distribute balanced linking tasks across mappers and reducers



1. CPU (Java)
   - 32-core server running Linux 10.0.4
   - AMD Opteron 6128 clocket at 2.0GHz
2. GPU (C++)
   - AMD Radeon 7870 with 20 compute units, 64 parallel threads
   - Host program ran on Intel Core i7 3770 CPU with 8GB RAM and Linux 12.10
   - Ran the Java code on the same machine for scaling
3. Cloud (Java)
   - 10 c1.medium nodes (2 cores, 1.7GB) for small experiments
   - 30 c1.large nodes (8 cores, 7GB) for large experiments

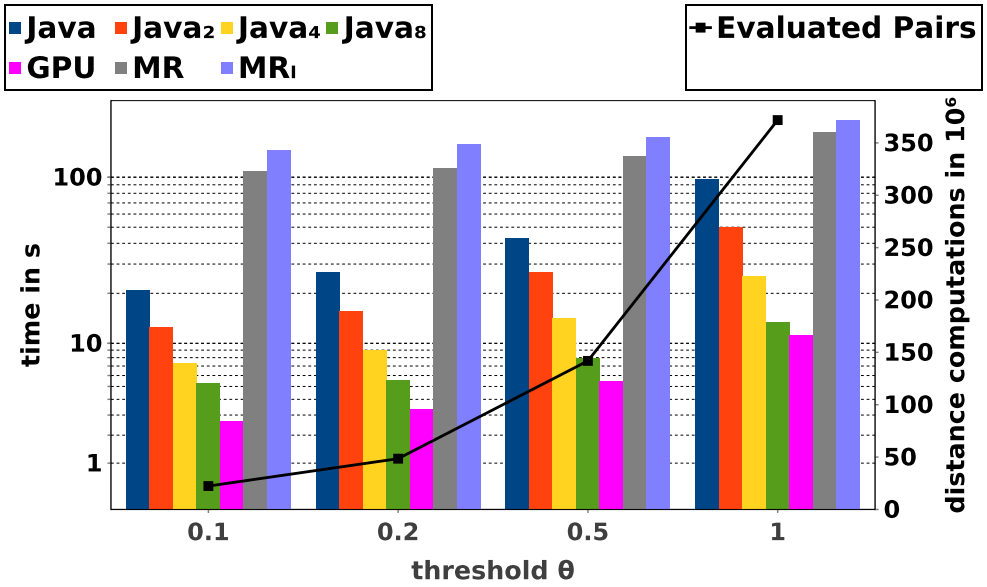




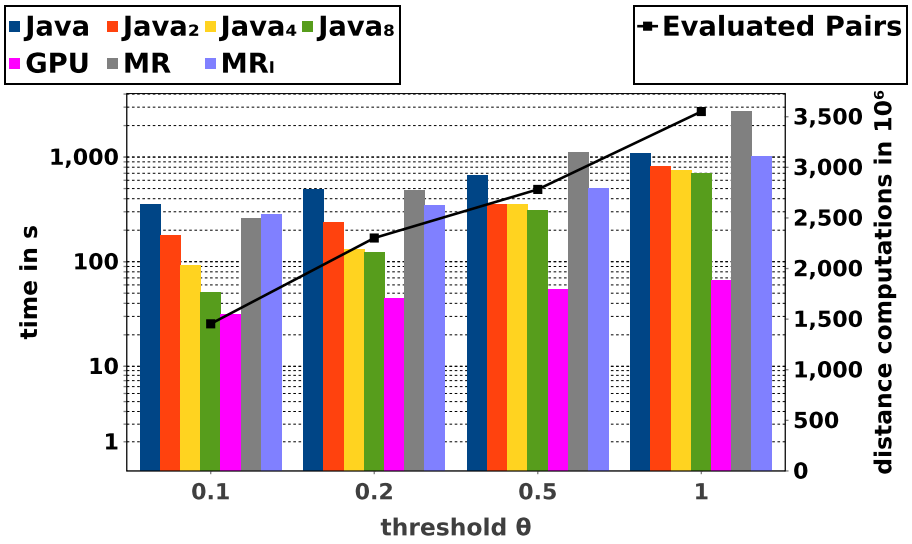
- Run deduplication task
- Evaluate behavior on different number of dimensions
- **Important**: Scale results
  - Different hardware (2-7 times faster C++ workstation)
  - Programming language
- Evaluate scalability (DS_4)

Dataset	Source	Size	Features
DS_1	DBPedia	25,781	min/medium/max elevation
DS_2	DBPedia	475,000	latitude, longitude
DS_3	Linked Geo Data	500,000	latitude, longitude
DS_4	Linked Geo Data	6,000,000	latitude, longitude

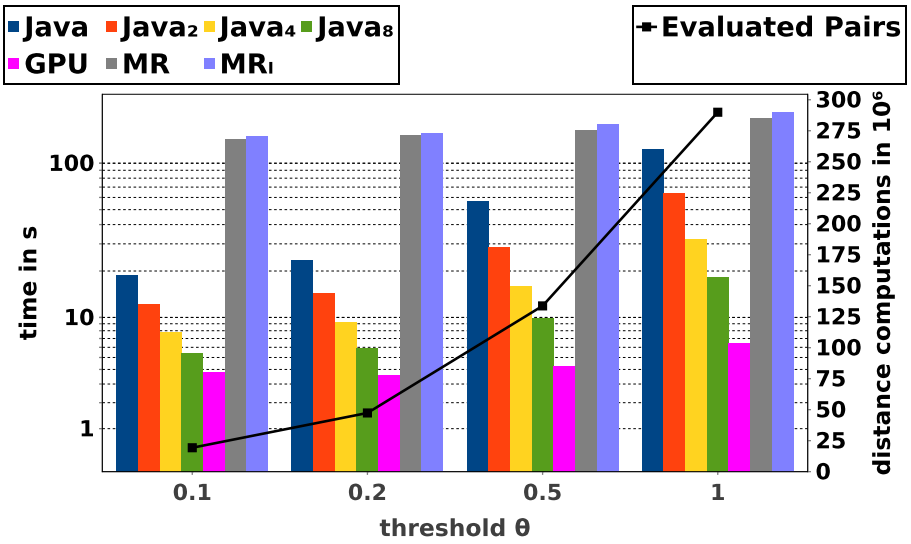
- DBpedia, 3 dimensions, 26K
- GPU scales better



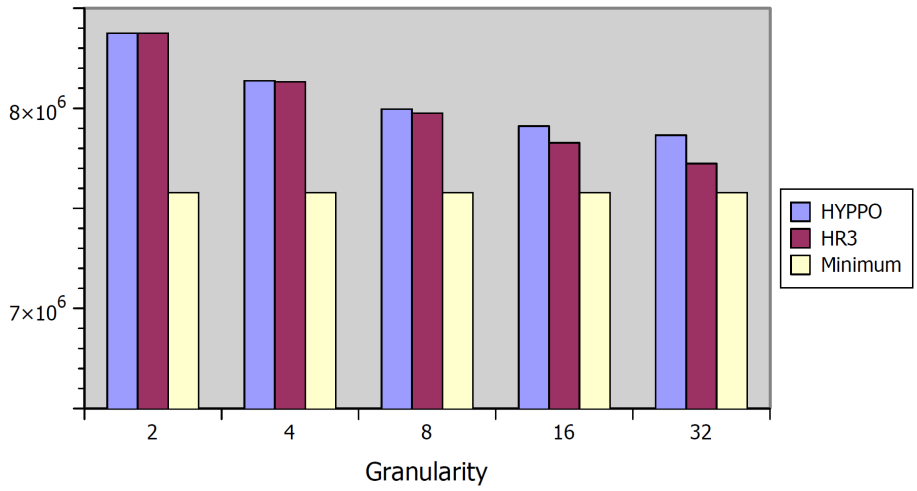
- DBpedia, 2 dimensions, 475K
- GPUs scale better across different θ
- **Break-even point** $\approx 10^8$ results



- LinkedGeoData, 2 dimensions, 500K
- Similar picture



- LinkedGeoData, 2 dimensions, 6M
- Cloud (with load balancing) better for $\approx 10^{10}+$ results



- **Question**: When should we use which hardware for link discovery?
- **Results**
  - Implemented $\mathcal{HR}^3$ on different hardware
  - Provided the first implementation of link discovery on GPUs
  - Devised a load balancing approach for linking on the cloud
  - Discovered 10^8 and 10^{10} results as break-even points

- **Question**: When should we use which hardware for link discovery?
- **Insights**
  - Load balancing important for using the cloud
  - GPUs: Need faster buses that PCIe (e.g., Firewire speed)
  - **Accurate use of local resources sufficient for most of the current applications**

Table of Contents

- Four approaches to scalability
  1. Improve reduction ratio (LIMES, MultiBlock, HYPPO, $\mathcal{HR}^3$, ...)
  2. Reduce runtime complexity (AEGLE, RADON)
  3. Better use of hardware (GNOME)
  4. Improve execution of specifications

- Four approaches to scalability
  1. Improve reduction ratio (LIMES, MultiBlock, HYPPO, $\mathcal{HR}^3$, ...)
  2. Reduce runtime complexity (AEGLE, RADON)
  3. Better use of hardware (GNOME)
  4. Improve execution of specifications
- Challenges include
  1. Increase number of reduction-ratio-optimal approaches ($\mathcal{HR}^3$, ORCHID)
  2. Adaptive resource scheduling
  3. Self-regulating approaches
  4. Distribution in modern in-memory architecture (SPARK)
  5. ...

HOBBIT
Holistic Benchmarking
of Big Linked Data

This work was supported by grants from the EU H2020 Framework Programme provided for the project HOBBIT (GA no. 688227).

References I

Kleanthi Georgala, Mohamed Ahmed Sherif, and Axel-Cyrille Ngonga Ngomo. "An Efficient Approach for the Generation of Allen Relations". In: ECAI 2016 - 22nd European Conference on Artificial Intelligence, 29 August-2 September 2016, The Hague, The Netherlands - Including Prestigious Applications of Artificial Intelligence (PAIS 2016). Ed. by Gal A. Kaminka et al. Vol. 285. Frontiers in Artificial Intelligence and Applications. IOS Press, 2016, pp. 948–956. ISBN: 978-1-61499-671-2. DOI: 10.3233/978-1-61499-672-9-948. URL: http://dx.doi.org/10.3233/978-1-61499-672-9-948.

Robert Isele, Anja Jentzsch, and Christian Bizer. "Efficient Multidimensional Blocking for Link Discovery without losing Recall". In: Proceedings of the 14th International Workshop on the Web and Databases 2011, WebDB 2011, Athens, Greece, June 12, 2011. Ed. by Amélie Marian and Vasilis Vassalos. 2011. URL: http://webdb2011.rutgers.edu/papers/Paper%2039/silk.pdf.

Axel-Cyrille Ngonga Ngomo and Sören Auer. "LIMES - A Time-Efficient Approach for Large-Scale Link Discovery on the Web of Data". In: IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011. Ed. by Toby Walsh. IJCAI/AAAI, 2011, pp. 2312–2317. ISBN: 978-1-57735-516-8. DOI: 10.5591/978-1-57735-516-8/IJCAI11-385. URL: http://dx.doi.org/10.5591/978-1-57735-516-8/IJCAI11-385.

Axel-Cyrille Ngonga Ngomo et al. "When to Reach for the Cloud: Using Parallel Hardware for Link Discovery". In: The Semantic Web: Semantics and Big Data, 10th International Conference, ESWC 2013, Montpellier, France, May 26-30, 2013. Proceedings. Ed. by Philipp Cimiano et al. Vol. 7882. Lecture Notes in Computer Science. Springer, 2013, pp. 275–289. ISBN: 978-3-642-38287-1. DOI: 10.1007/978-3-642-38288-8_19. URL: http://dx.doi.org/10.1007/978-3-642-38288-8_19.

Axel-Cyrille Ngonga Ngomo. "Link Discovery with Guaranteed Reduction Ratio in Affine Spaces with Minkowski Measures". In: The Semantic Web - ISWC 2012 - 11th International Semantic Web Conference, Boston, MA, USA, November 11-15, 2012, Proceedings, Part I. Ed. by Philippe Cudré-Mauroux et al. Vol. 7649. Lecture Notes in Computer Science. Springer, 2012, pp. 378–393. ISBN: 978-3-642-35175-4. DOI: 10.1007/978-3-642-35176-1_24. URL: http://dx.doi.org/10.1007/978-3-642-35176-1_24.

Axel-Cyrille Ngonga Ngomo. "ORCHID - Reduction-Ratio-Optimal Computation of Geo-spatial Distances for Link Discovery". In: The Semantic Web - ISWC 2013 - 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part I. Ed. by Harith Alani et al. Vol. 8218. Lecture Notes in Computer Science. Springer, 2013, pp. 395–410. ISBN: 978-3-642-41334-6. DOI: 10.1007/978-3-642-41335-3_25. URL: http://dx.doi.org/10.1007/978-3-642-41335-3_25.

References IV

Axel-Cyrille Ngonga Ngomo. "HELIOS - Execution Optimization for Link Discovery". In: The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I. Ed. by Peter Mika et al. Vol. 8796. Lecture Notes in Computer Science. Springer, 2014, pp. 17–32. ISBN: 978-3-319-11963-2. DOI: 10.1007/978-3-319-11964-9_2. URL: http://dx.doi.org/10.1007/978-3-319-11964-9_2.

Axel-Cyrille Ngonga Ngomo and Mofeed M. Hassan. "The Lazy Traveling Salesman - Memory Management for Large-Scale Link Discovery". In: The Semantic Web. Latest Advances and New Domains - 13th International Conference, ESWC 2016, Heraklion, Crete, Greece, May 29 - June 2, 2016, Proceedings. Ed. by Harald Sack et al. Vol. 9678. Lecture Notes in Computer Science. Springer, 2016, pp. 423–438. ISBN: 978-3-319-34128-6. DOI: 10.1007/978-3-319-34129-3_26. URL: http://dx.doi.org/10.1007/978-3-319-34129-3_26.

Link Discovery Tutorial

Part II: Accuracy

Mohamed Ahmed Sherif[1], Irini Fundulaki[2], Axel-Cyrille Ngonga Ngomo[1]

(1) Institute for Applied Informatics, Germany
(2) FORTH, Greece

May 28th, 2017
Portoroz, Slovenia.

Table of Contents

Table of Contents

Definition (Declarative Link Discovery)

- Given sets S and T of resources and relation $\mathcal{R}$
- Find $M = \{(s,t) \in S \times T : \mathcal{R}(s,t)\}$
- Here, find $M' = \{(s,t) \in S \times T : \sigma(s,t) \geq \theta\}$

Definition (Declarative Link Discovery)

- Given sets S and T of resources and relation $\mathcal{R}$
- Find $M = \{(s,t) \in S \times T : \mathcal{R}(s,t)\}$
- Here, find $M' = \{(s,t) \in S \times T : \sigma(s,t) \geq \theta\}$

Definition (Classification perspective)

- Given sets S and T of resources and relation $\mathcal{R}$
- Find $M = \{(s,t) \in S \times T : \mathcal{C}(s,t) = +1\}$
- Here, $\mathcal{C}(s,t) = +1 \leftrightarrow \sigma(s,t) \geq \theta$

Definition (Declarative Link Discovery)

- Given sets S and T of resources and relation $\mathcal{R}$
- Find $M = \{(s,t) \in S \times T : \mathcal{R}(s,t)\}$
- Here, find $M' = \{(s,t) \in S \times T : \sigma(s,t) \geq \theta\}$

Definition (Classification perspective)

- Given sets S and T of resources and relation $\mathcal{R}$
- Find $M = \{(s,t) \in S \times T : \mathcal{C}(s,t) = +1\}$
- Here, $\mathcal{C}(s,t) = +1 \leftrightarrow \sigma(s,t) \geq \theta$

- Classical machine learning problem [Ngo+11; NL12]
- Dedicated techniques perform better
- Unsupervised, active and unsupervised techniques possible

Challenges

1. Creation of labeled training data tedious
2. Need automated means for automatic class and property matching
3. Need for efficient execution of link specifications
4. Dedicated machine learning approaches necessary

Challenges

1. Creation of labeled training data tedious
2. Need automated means for automatic class and property matching
3. Need for efficient execution of link specifications
4. Dedicated machine learning approaches necessary

Solutions

1. Use active learning approach for link discovery
2. Rely on hospital/resident algorithm
3. See previous section
4. Topic of this section

Table of Contents

Definition (Classification perspective)

- Given sets S and T of resources and relation $\mathcal{R}$
- Find $M = \{(s,t) \in S \times T : \mathcal{C}(s,t) = +1\}$
- Here, $\mathcal{C}(s,t) = +1 \leftrightarrow \sigma(s,t) \geq \theta$

Definition (Classification perspective)

- Given sets S and T of resources and relation $\mathcal{R}$
- Find $M = \{(s,t) \in S \times T : \mathcal{C}(s,t) = +1\}$
- Here, $\mathcal{C}(s,t) = +1 \leftrightarrow \sigma(s,t) \geq \theta$

- Learning classifier C involves learning
  1. Two sets of restrictions that specify the sets S resp. T,
  2. the components $\sigma_1 \ldots \sigma_n$ of a complex similarity measure σ
  3. a set of thresholds θ_1, ..., θ_n for $\sigma_1, \ldots, \sigma_n$

Definition (Classification perspective)

- Given sets S and T of resources and relation $\mathcal{R}$
- Find $M = \{(s,t) \in S \times T : \mathcal{C}(s,t) = +1\}$
- Here, $\mathcal{C}(s,t) = +1 \leftrightarrow \sigma(s,t) \geq \theta$

- Learning classifier C involves learning
  1. Two sets of restrictions that specify the sets S resp. T,
  2. the components $\sigma_1 \ldots \sigma_n$ of a complex similarity measure σ
  3. a set of thresholds θ_1, ..., θ_n for $\sigma_1, \ldots, \sigma_n$

Assumptions

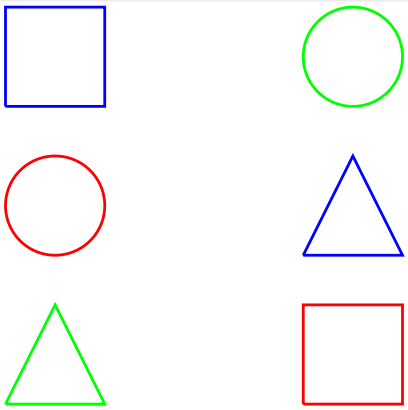
- Restrictions are class restrictions
- Classifier shape is given (e.g., linear combination)

Class and Property Restrictions

- Define class similarity function
- Solve corresponding hospital-resident problem
- Based on extension of stable marriage problem

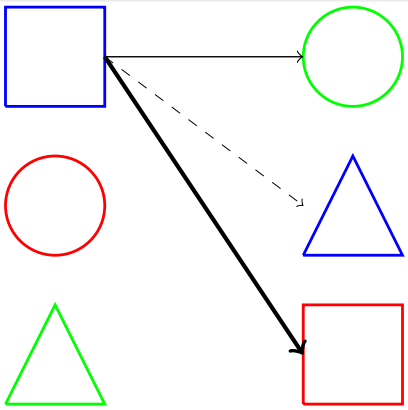
Class and Property Restrictions

- Define class similarity function
- Solve corresponding hospital-resident problem
- Based on extension of stable marriage problem



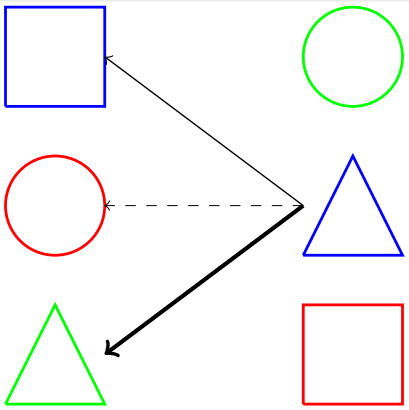
Class and Property Restrictions

- Define class similarity function
- Solve corresponding hospital-resident problem
- Based on extension of stable marriage problem



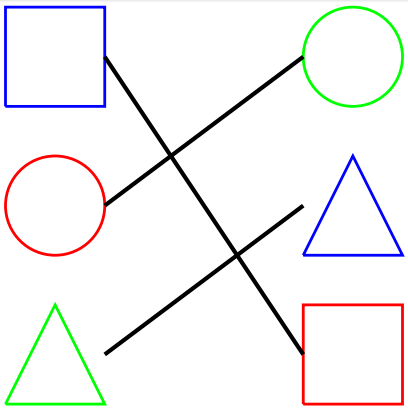
Class and Property Restrictions

- Define class similarity function
- Solve corresponding hospital-resident problem
- Based on extension of stable marriage problem



Class and Property Restrictions

- Define class similarity function
- Solve corresponding hospital-resident problem
- Based on extension of stable marriage problem



Class Restrictions

- Similarity function
  - String similarity
  - Number of shared property values amongst instances
  - ...
- Solve corresponding hospital-resident problem

Class Restrictions

- Similarity function
  - String similarity
  - Number of shared property values amongst instances
  - ...
- Solve corresponding hospital-resident problem

Source	Target	S	T
Drugbank	Disesome	Targets	Genes
Sider	Diseasome	Side-Effect	Diseases
DBpedia	Dailymed	Organization	Organization
Sider	Dailymed	Drugs	Offer
Drugbank	DBpedia	Targets	Protein

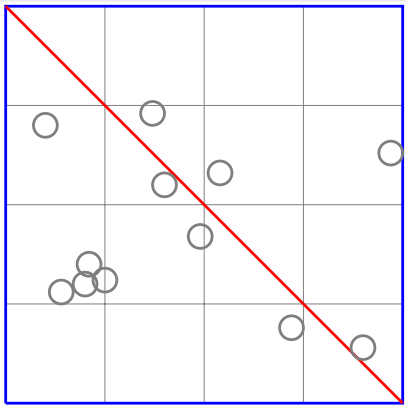
- Property mapping similar
- Leads to $\sigma_1 \ldots \sigma_n$

Learning Threshold

- Active perceptron learning
- Begin with educated guess, e.g., $\theta_i = 0.9$
- Update thresholds based on most informative examples

Learning Threshold

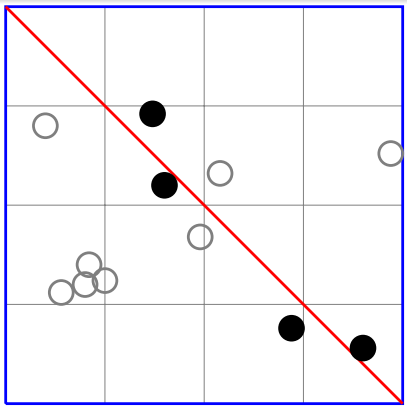
- Active perceptron learning
- Begin with educated guess, e.g., $\theta_i = 0.9$
- Update thresholds based on most informative examples



- Guess initial classifier

Learning Threshold

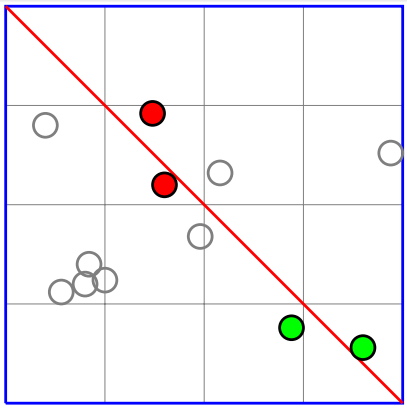
- Active perceptron learning
- Begin with educated guess, e.g., $\theta_i = 0.9$
- Update thresholds based on most informative examples



- Pick most informative examples, i.e., unclassified and closest to boundary

Learning Threshold

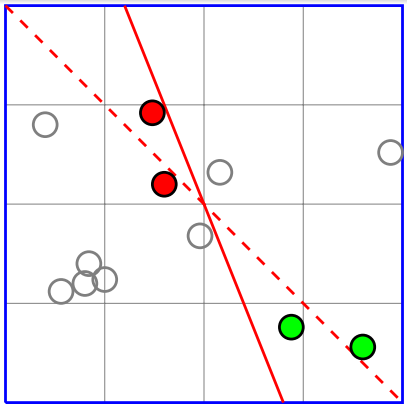
- Active perceptron learning
- Begin with educated guess, e.g., $\theta_i = 0.9$
- Update thresholds based on most informative examples



- Ask for classification from oracle

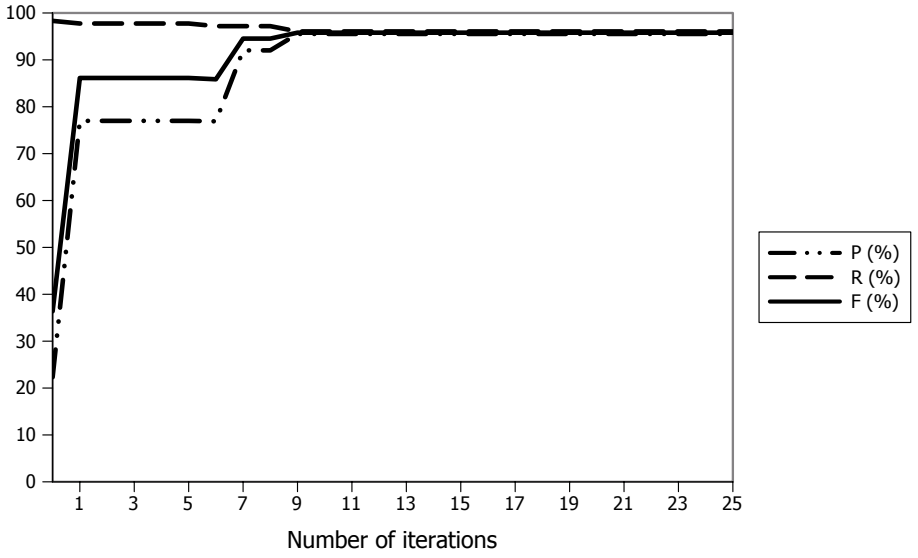
Learning Threshold

- Active perceptron learning
- Begin with educated guess, e.g., $\theta_i = 0.9$
- Update thresholds based on most informative examples



- Update classifier

- Evaluation on Diseases (Diseasome to DBpedia)
- Learning rate = 0.02
- 10 questions/iteration
- F-measure of up to 92%



- Learning rate = 0.02
- 10 questions/iteration

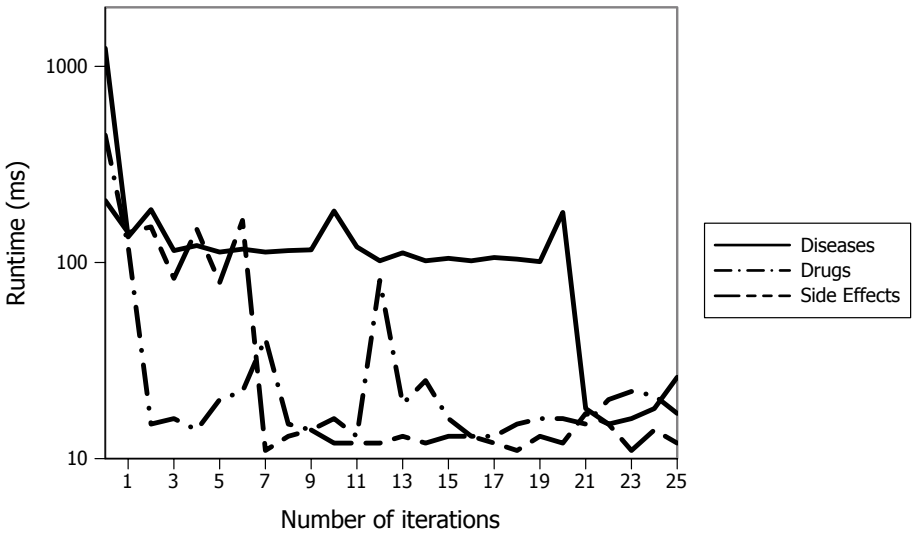


Table of Contents

Eagle

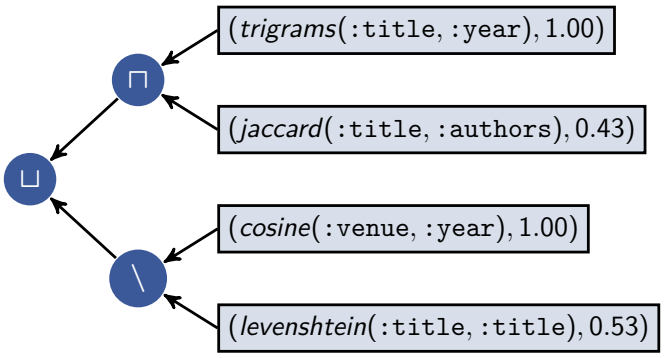
Efficient Active Learning of Link Specifications using Genetic Programming

EAGLE

- Provides means for automatic class and property matching
- Minimizes human labeling effort through active learning
- Allow for learning generic specs (limitation of RAVEN)
- Similar approaches [**NIK+12**; **ISE+12**]

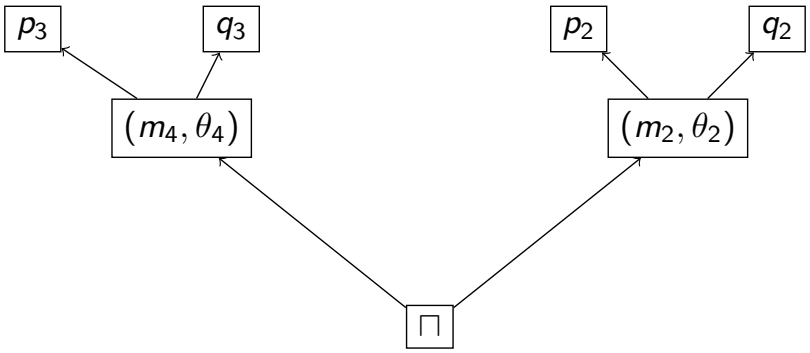
- Same formal setting as RAVEN
  - Two sets of restrictions resp. that specify the sets S resp. T,
  - a specification of mapping properties $(p_1, q_1), \ldots, (p_n, q_n)$ for the elements of S and T and
  - a specification of a complex similarity measure σ as the combination of several atomic similarity measures $\sigma_1, \ldots, \sigma_n$ and of a set of thresholds $\theta_1, \ldots, \theta_n$ such that θ_i is the threshold for σ_i.

- Can learn generic classifier type

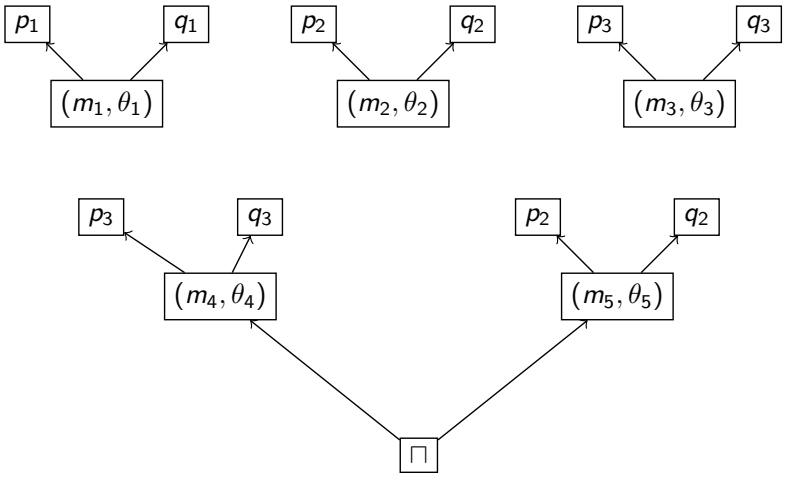


EAGLE

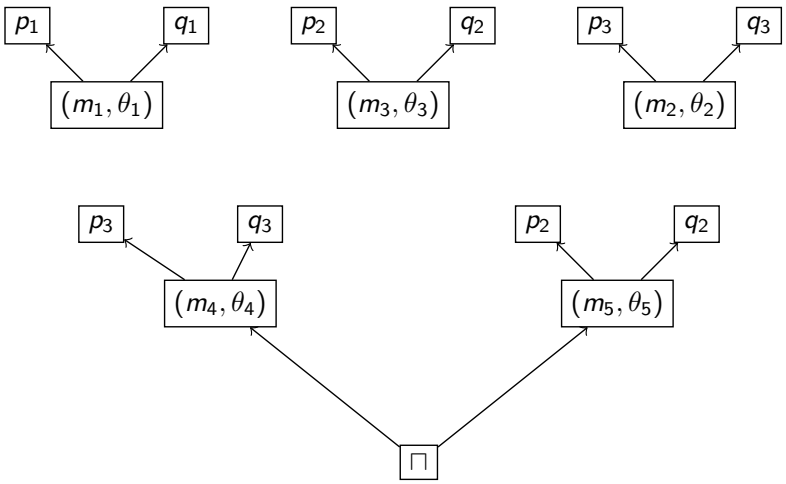
- Idea: Specifications are trees
- Goal: Learn elements of trees through genetic operations until best LS is found



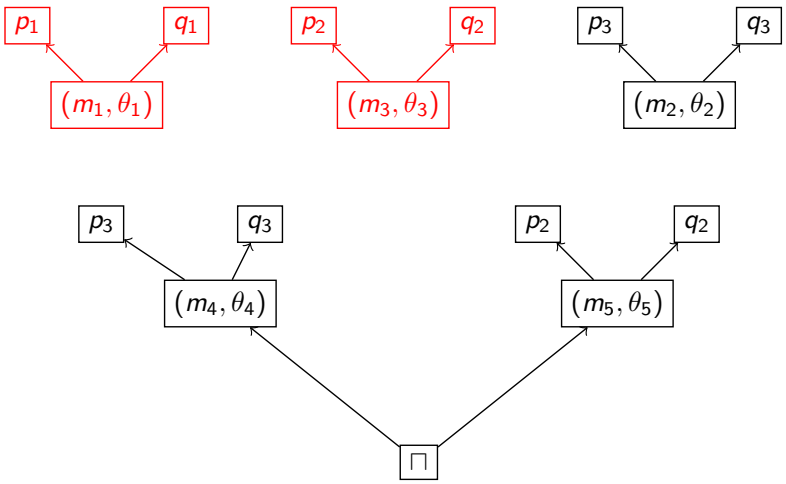
- Random process (property pairs, thresholds)
- Compute fitness
- Fitness = F-Measure w.r.t known data



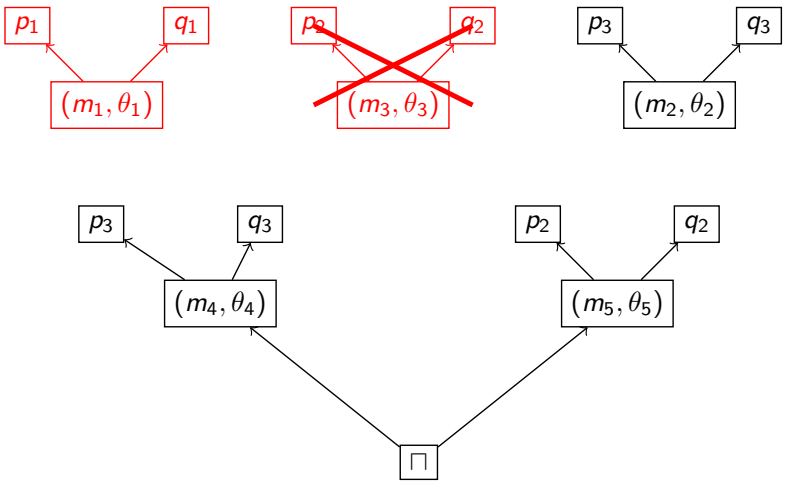
- Tournament between two individuals
- Two operators: Mutation and crossover



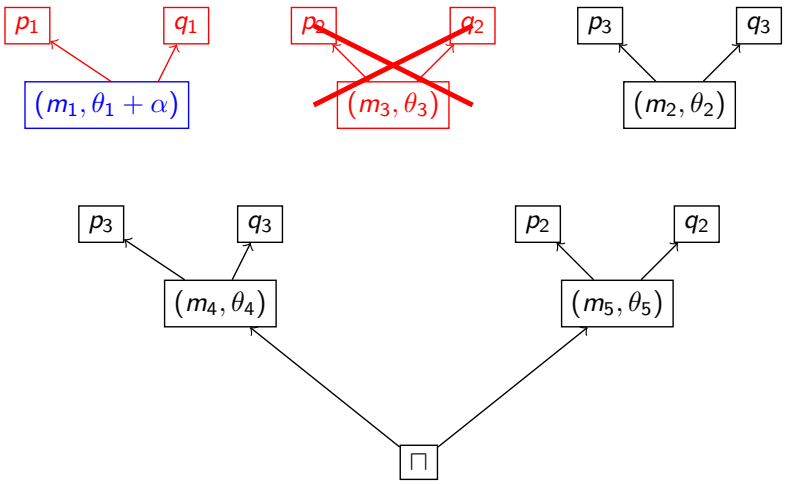
- Tournament between two individuals
- Two operators: Mutation and crossover



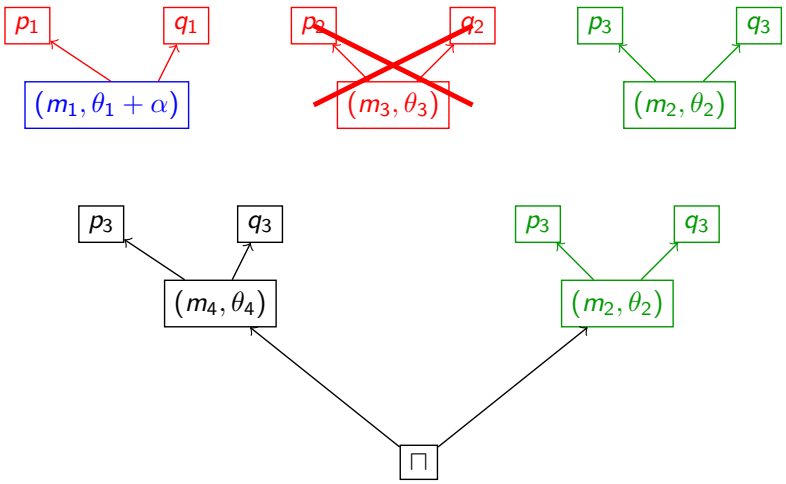
- Tournament between two individuals
- Two operators: Mutation and crossover



- Tournament between two individuals
- Two operators: Mutation and crossover



- Tournament between two individuals
- Two operators: Mutation and crossover

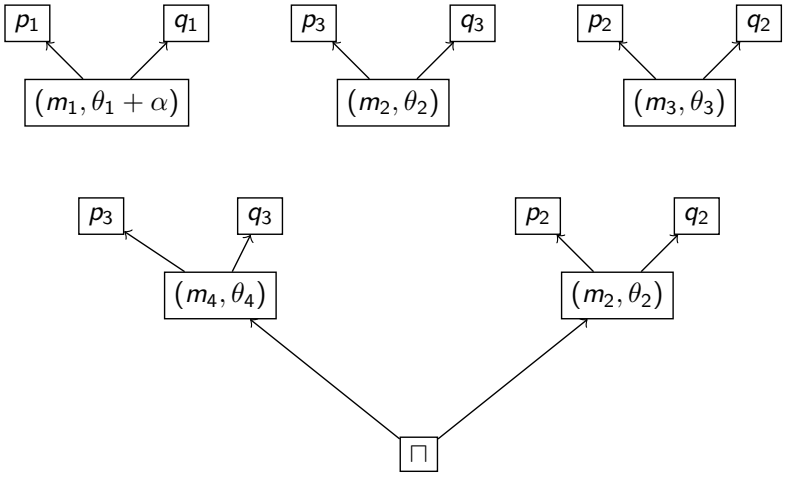


- Previous approaches define amount of information of link as closeness to the decision boundary
- Here, use disagreement amongst elements of population of size n

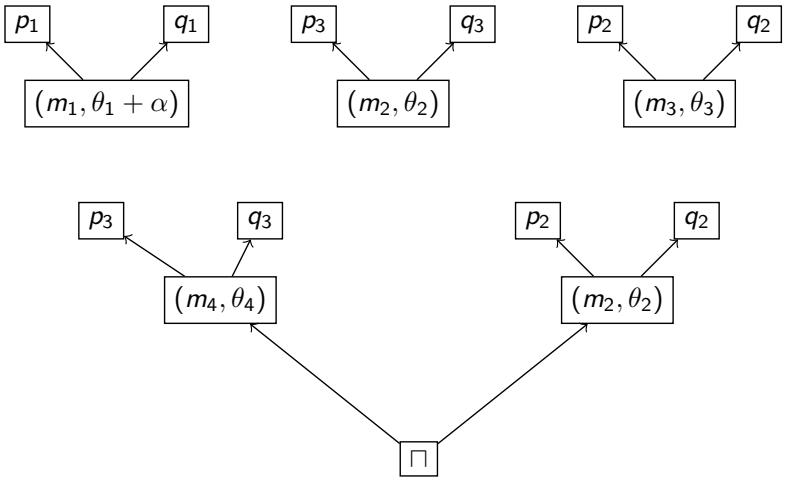
$$\delta((s,t)) = (n - |\mathcal{M}_i^t : (s,t) \in \mathcal{M}_i|)(n - |\mathcal{M}_i^t : (s,t) \notin \mathcal{M}_i|)$$

- Function is maximal when $\frac{n}{2}$ count (s,t) as positive and $\frac{n}{2}$ as negative
- Can be modeled with other functions such as entropy

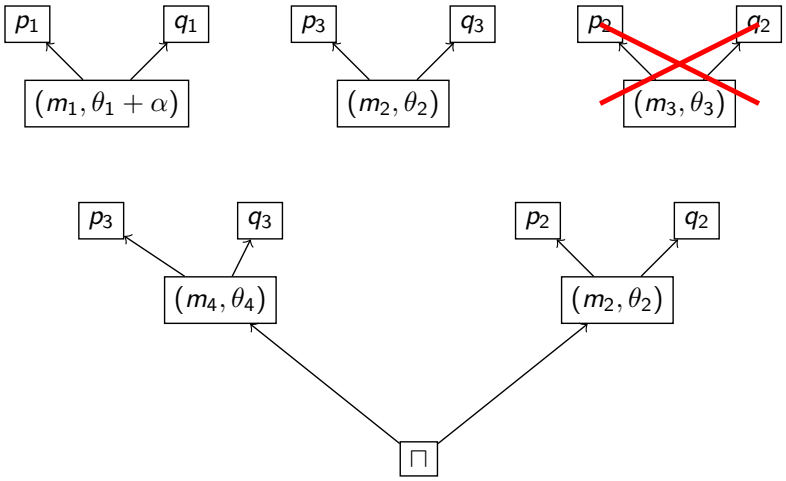
- Compute $\delta((s,t))$ for all (s,t) returned by a LS
- Pick k most informative
- Require labeling from user
- Update list of positive and negative examples



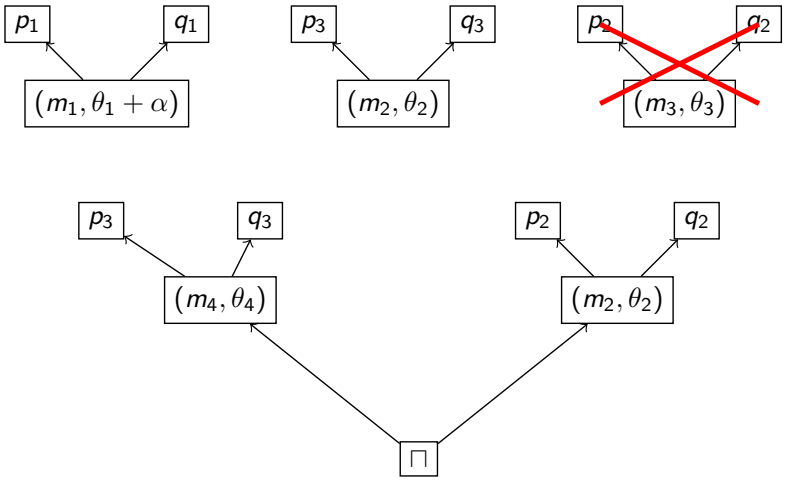
- Fitness = F-Measure w.r.t known data



- Fitness = F-Measure w.r.t known data



- Fitness = F-Measure w.r.t known data



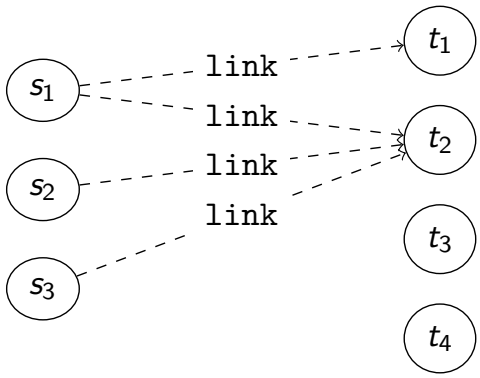
- If termination conditions not met, goto Step 2
- Else terminate and pick fittest LS

- Measure degree of monogamy of links [**NIK+12**]
- Only works for 1-1 relations, e.g., `owl:sameAs`

$$\mathcal{P}(M) = \frac{|\{s|\exists t : (s,t) \in M\}|}{\sum_s |\{t : (s,t) \in M\}|},$$

$$\mathcal{R}(M) = \frac{|\{t|\exists s : (s,t) \in M\}|}{\sum_t |\{s : (s,t) \in M\}|},$$

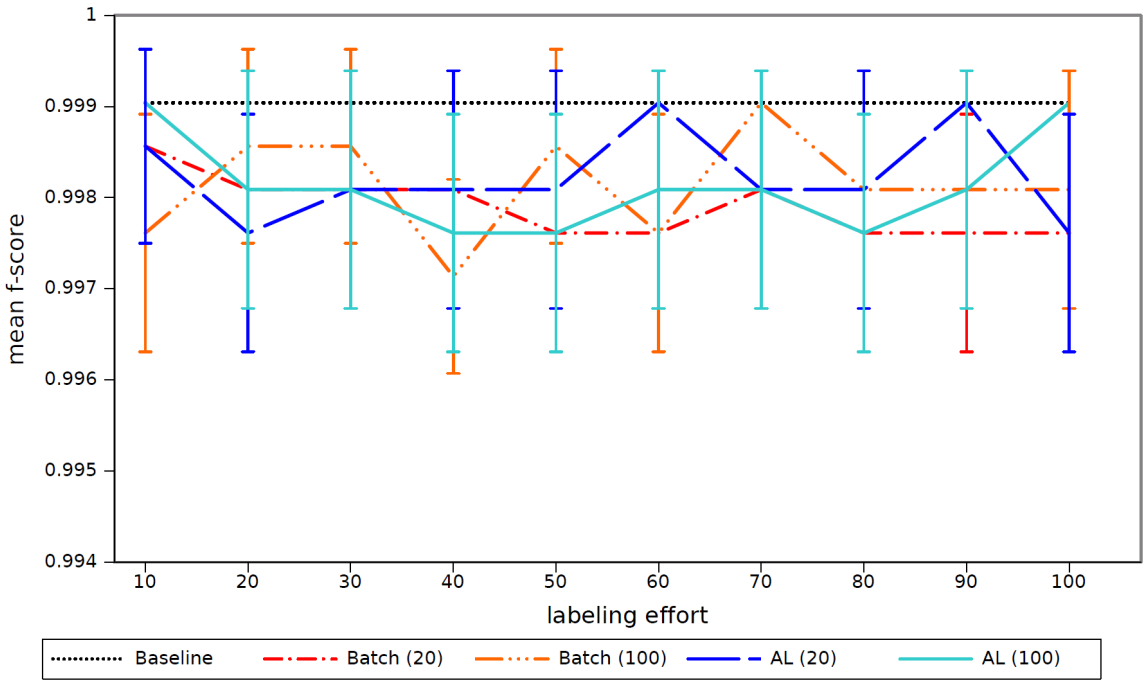
$$\mathcal{F}^{\beta}(M) = (1+\beta^2)\frac{\mathcal{P}_d(M)\mathcal{R}_d(M)}{\beta^2\mathcal{P}_d(M) + \mathcal{R}_d(M)}$$

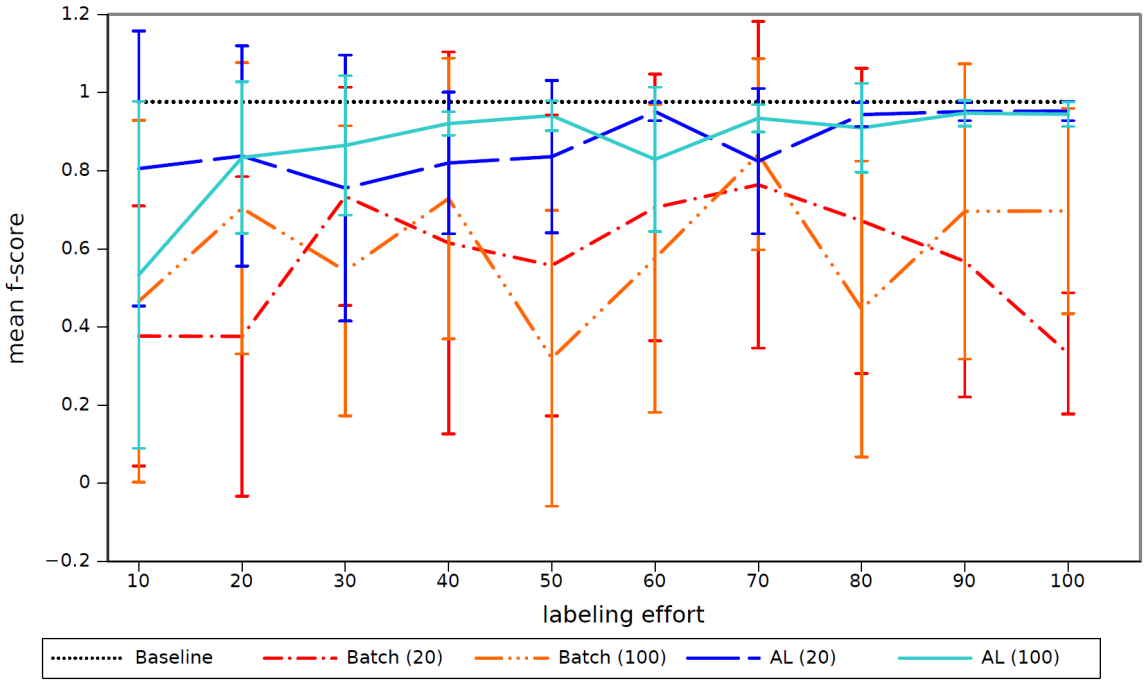


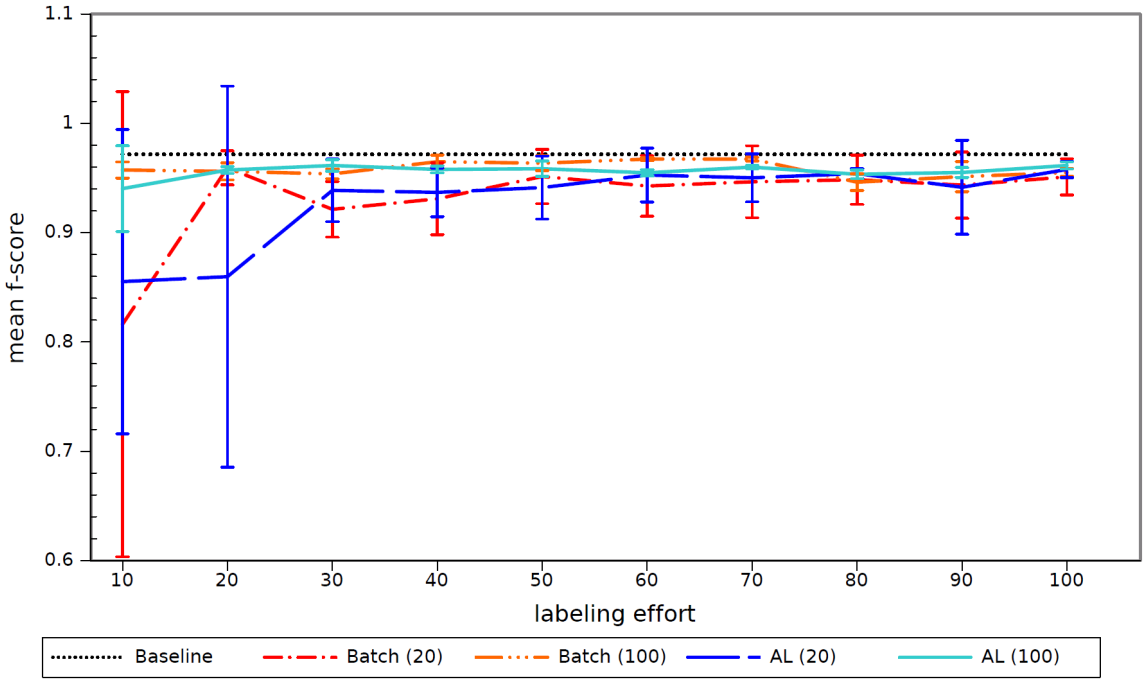
- $\mathcal{P} = 3/4$, $\mathcal{R} = 2/4$, $\mathcal{F} = 3/5$.

Experimental Setup:

- Compared batch learning and genetic programming
- Used 3 different data sets
  1. Dailymed-Drugbank (LATC)
  2. DBpedia-LinkedMDB (LATC)
  3. DBLP-ACM
- Compared different sizes of population (20,100)
- Compared random annotation with active learning
- Mutation and crossover rates = 0.6
- Maximal number of iterations = 50


1
0.999
0.998
0.997
0.996
0.995
0.994
mean f-score
10
20
30
40
50
60
70
80
90
100
labeling effort
Baseline
Batch (20)
Batch (100)
AL (20)
AL (100)

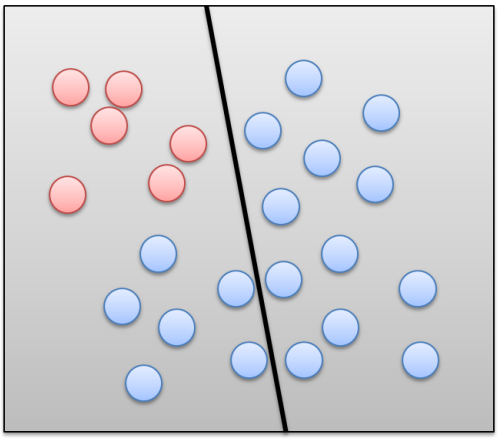

1.2
1
0.8
0.6
0.4
0.2
0
−0.2
mean f-score
10
20
30
40
50
60
70
80
90
100
labeling effort
Baseline
Batch (20)
Batch (100)
AL (20)
AL (100)


1.1
1
0.9
0.8
0.7
0.6
mean f-score
10
20
30
40
50
60
70
80
90
100
labeling effort
Baseline
Batch (20)
Batch (100)
AL (20)
AL (100)

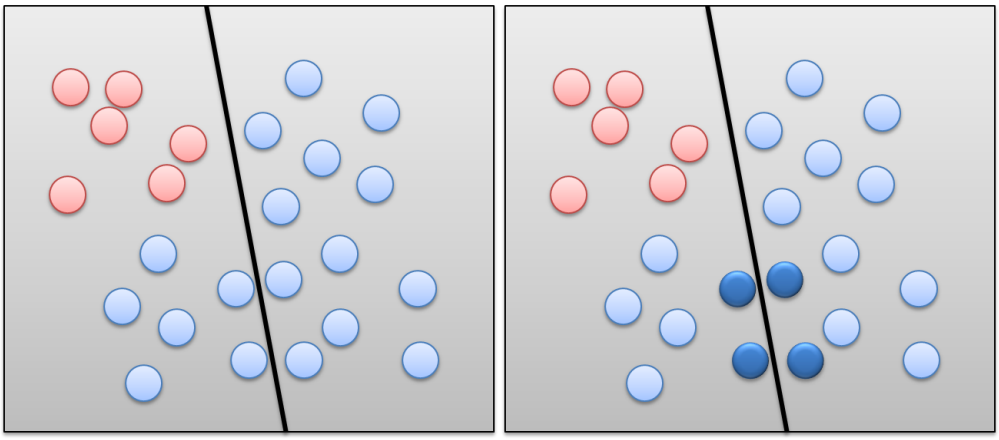
- Larger population leads to
  - Better results, yet
  - Longer runtimes
- For most datasets, population size of 100 seems sufficient for most linked data sets
- EAGLE is more time-efficient than state of the art
  - 337s for ACM-DBLP (n=100) vs.
  - 1553s for Marlin (ADTree)
  - 2196s for Marlin (SVM)
  - 4320s for Febrl (SVM)
- Active learning clearly outperforms random annotation

Table of Contents

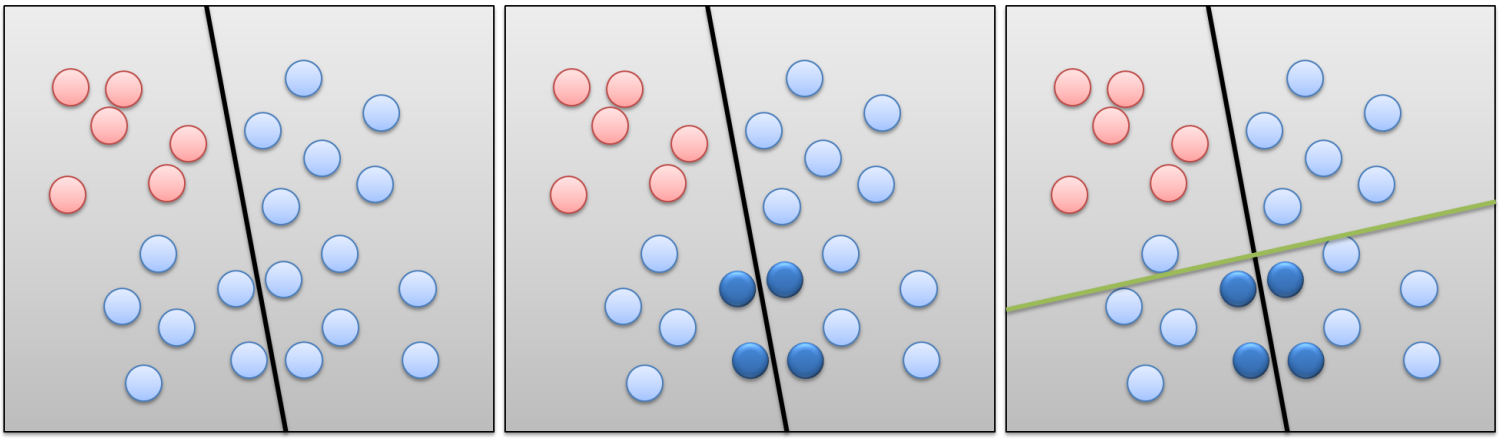
- **Supervised** (**mostly active**, e.g., RAVEN, EAGLE, SILK)
- Unsupervised (e.g., KnoFuss, EUCLID, EAGLE)



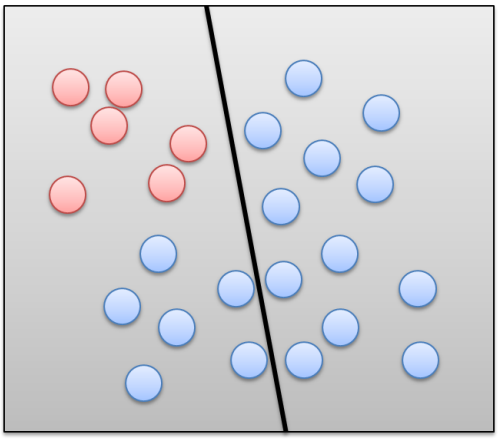
- **Supervised** (**mostly active**, e.g., RAVEN, EAGLE, SILK)
- Unsupervised (e.g., KnoFuss, EUCLID, EAGLE)



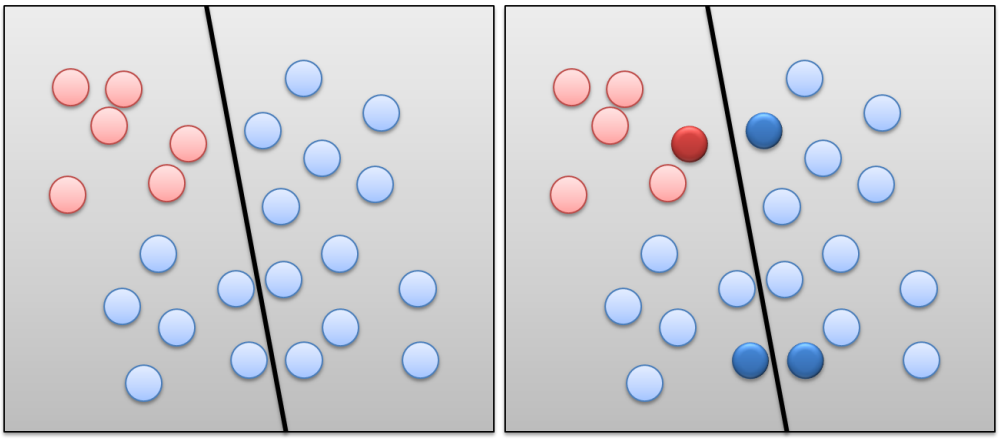
- **Supervised** (**mostly active**, e.g., RAVEN, EAGLE, SILK)
- Unsupervised (e.g., KnoFuss, EUCLID, EAGLE)



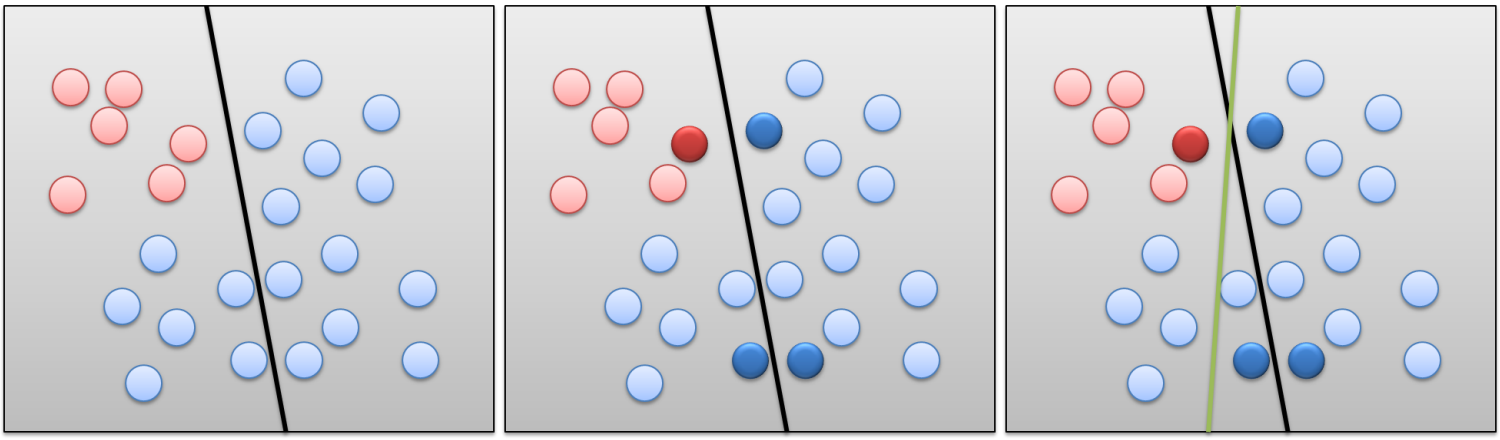
- **Supervised** (**mostly active**, e.g., RAVEN, EAGLE, SILK)
- Unsupervised (e.g., KnoFuss, EUCLID, EAGLE)



- **Supervised** (**mostly active**, e.g., RAVEN, EAGLE, SILK)
- Unsupervised (e.g., KnoFuss, EUCLID, EAGLE)



- **Supervised** (**mostly active**, e.g., RAVEN, EAGLE, SILK)
- Unsupervised (e.g., KnoFuss, EUCLID, EAGLE)



Insight

- Choice of right example is key for learning
- So far, only use of informativeness

Insight

- Choice of right example is key for learning
- So far, only use of informativeness

Question

- Can we do better by using more information?

Insight

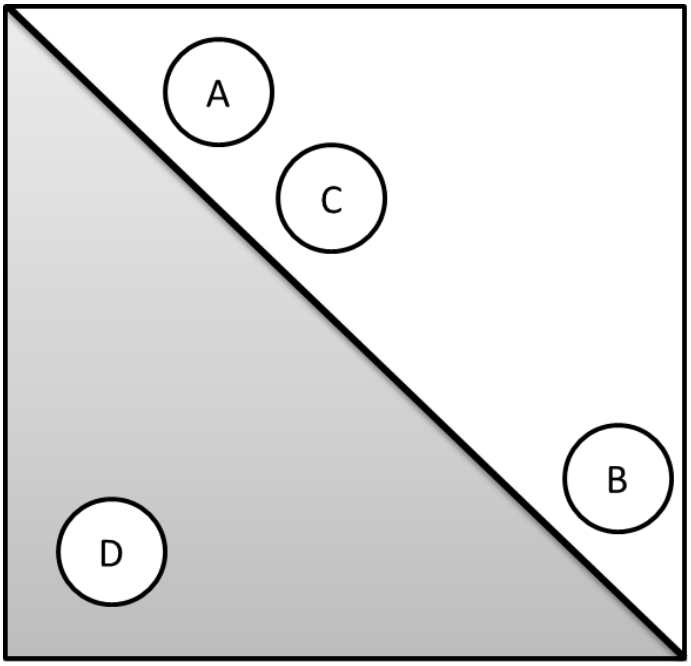
- Choice of right example is key for learning
- So far, only use of informativeness

Question

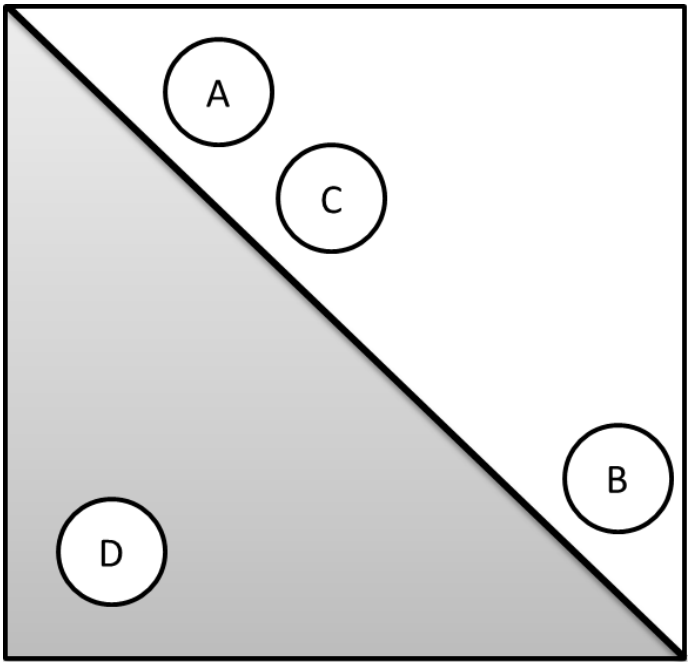
- Can we do better by using more information?
- **Higher F-measure**
- **Often slower**

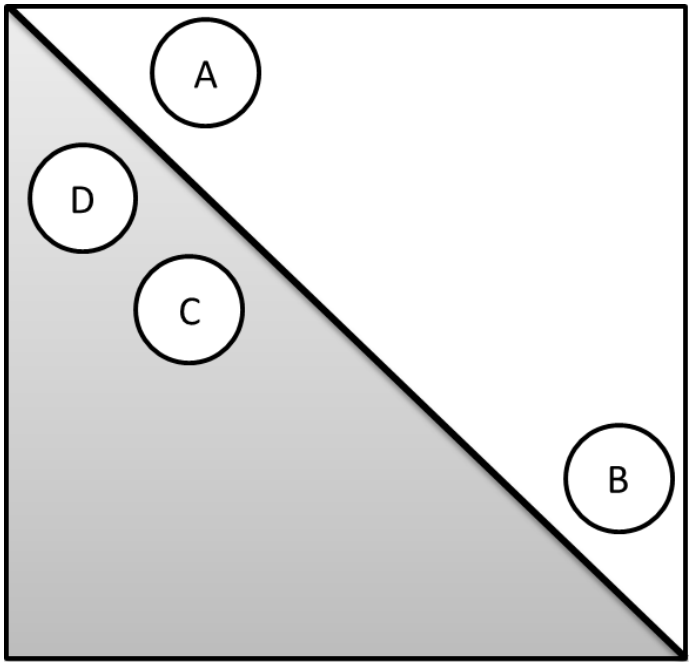
- Use similarity of link candidates when selecting most informative examples

- Use similarity of link candidates when selecting most informative examples



- Use similarity of link candidates when selecting most informative examples



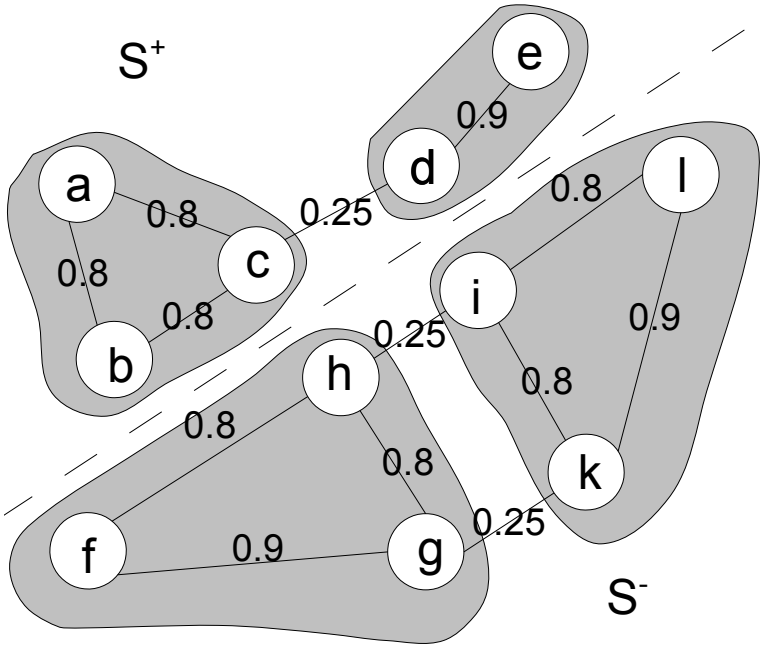


- Link candidate $x = (s, t)$ can be regarded as vector $(\sigma_1(x), \ldots, \sigma_n(x)) \in [0, 1]^n$.
- Similarity of link candidates x and y:

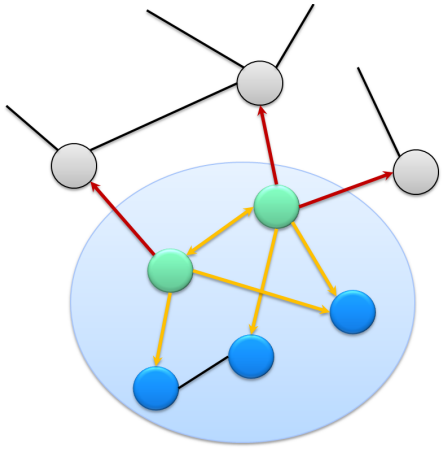
$$sim(x, y) = \frac{1}{1 + \sqrt{\sum_{i=1}^{n} (\sigma_i(x) - \sigma_i(y))^2}}. \quad (1)$$

- Allows exploiting both intra- and inter-class similarity

- **Rationale**: Use intra-class similarity
- **Approach**
  - Cluster elements of S^+ and S^- independently
  - Choose one element per cluster as representative
  - Present oracle with most informative representatives

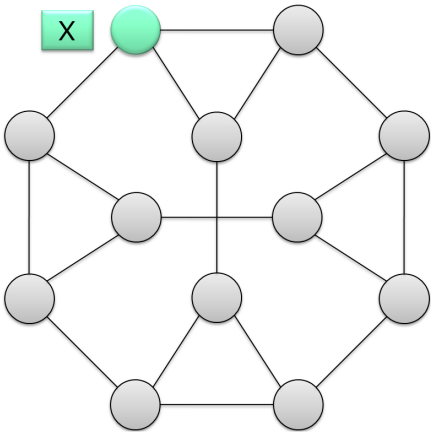


- $G = (V, E, \omega)$ with $V = S^+$ or $V = S^-$
- $\omega(x, y) = sim(x, y)$
- Keep best ec edges for each $x \in V$

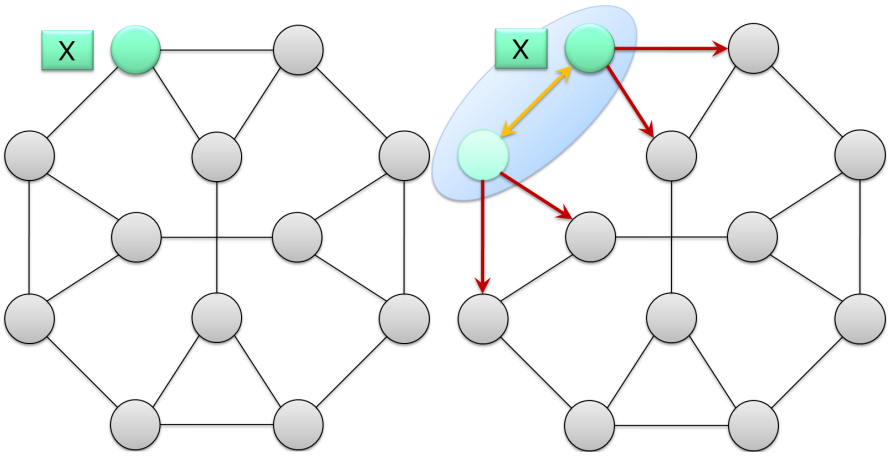


- Seed-based algorithm
- Goal: Maximize borderflow ratio $bf(X) = \frac{\Omega(b(X),X)}{\Omega(b(X),n(X))}$

- Seed-based algorithm
- Goal: Maximize borderflow ratio $bf(X) = \frac{\Omega(b(X),X)}{\Omega(b(X),n(X))}$

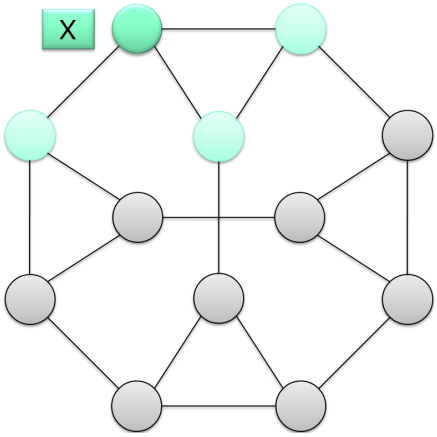


- Seed-based algorithm
- Goal: Maximize borderflow ratio $bf(X) = \frac{\Omega(b(X),X)}{\Omega(b(X),n(X))}$

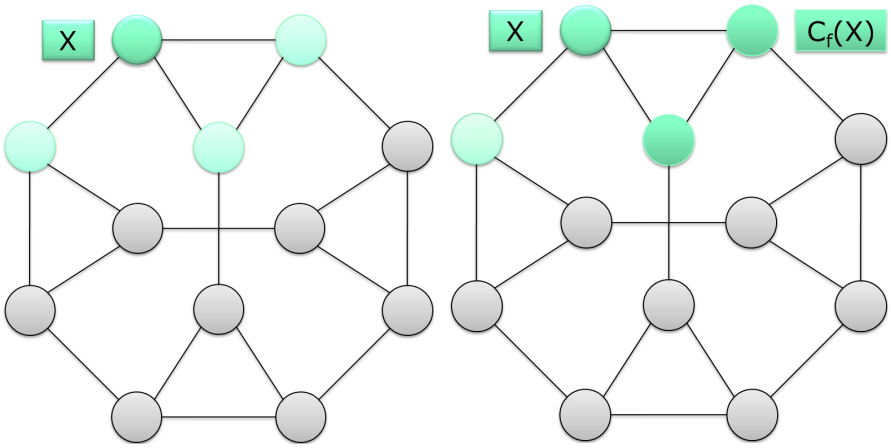


`http://sourceforge.net/projects/cugar-framework/`

- Seed-based algorithm
- Goal: Maximize borderflow ratio $bf(X) = \frac{\Omega(b(X),X)}{\Omega(b(X),n(X))}$



- Seed-based algorithm
- Goal: Maximize borderflow ratio $bf(X) = \frac{\Omega(b(X),X)}{\Omega(b(X),n(X))}$

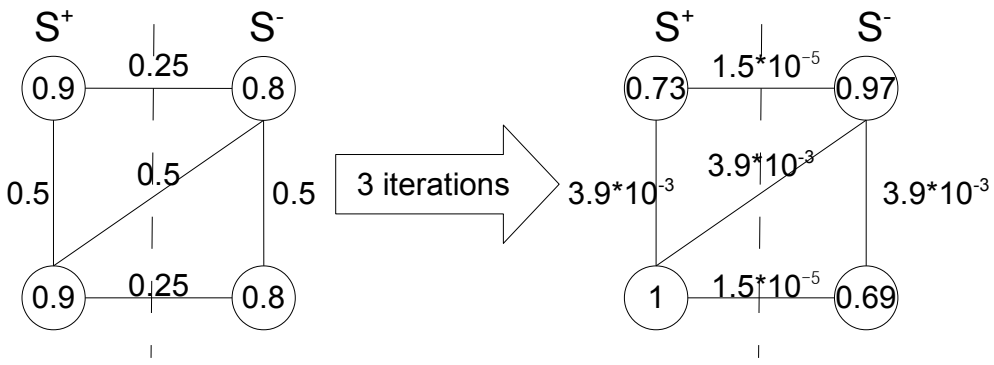


`http://sourceforge.net/projects/cugar-framework/`

- **Rationale**: Use both inter- and intra-class similarity
- **Approach**
  - $\mathcal{M}_0 : m_{ij} = sim(x_i, x_j)$ with $(x_i, x_j) \in (S^+ \cup S^-)^2$
  - $\mathcal{A}_0 : a_i = ifm(x_i)$

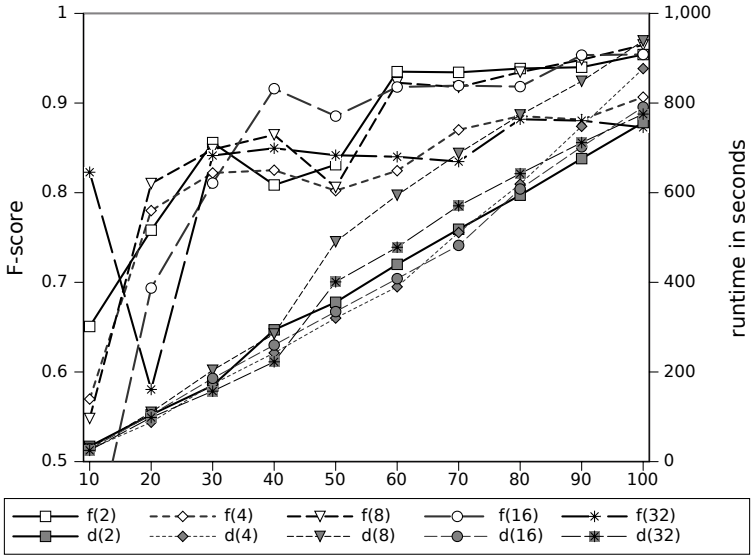
- **Rationale**: Use both inter- and intra-class similarity
- **Approach**
  - $\mathcal{M}_0 : m_{ij} = sim(x_i, x_j)$ with $(x_i, x_j) \in (S^+ \cup S^-)^2$
  - $\mathcal{A}_0 : a_i = ifm(x_i)$
  - $\mathcal{A}_t = \mathcal{A}_{t-1} + \mathcal{M}_{t-1}\mathcal{A}_{t-1}$ (**spread activation**)
  - $\mathcal{A}_t = \mathcal{A}_t / \max(\mathcal{A}_t)$ (**normalize**)
  - $\mathcal{M}_t = \mathcal{M}_{t-1}^{(r)}$ (**weight decay**)

- **Rationale**: Use both inter- and intra-class similarity
- **Approach**
  - $\mathcal{M}_0 : m_{ij} = sim(x_i, x_j)$ with $(x_i, x_j) \in (S^+ \cup S^-)^2$
  - $\mathcal{A}_0 : a_i = ifm(x_i)$
  - $\mathcal{A}_t = \mathcal{A}_{t-1} + \mathcal{M}_{t-1}\mathcal{A}_{t-1}$ (**spread activation**)
  - $\mathcal{A}_t = \mathcal{A}_t / \max(\mathcal{A}_t)$ (**normalize**)
  - $\mathcal{M}_t = \mathcal{M}_{t-1}^{(r)}$ (**weight decay**)

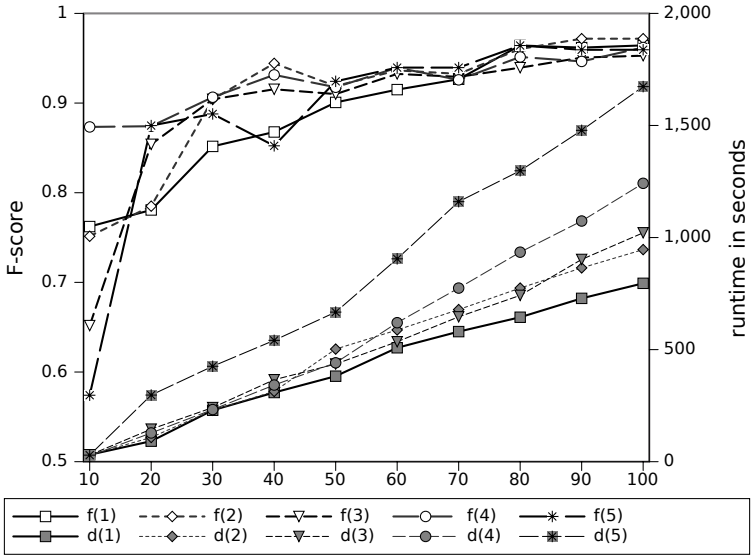


- Used EAGLE as active learning approach
  - Mutation and crossover rate = 0.6
  - Selection rate = 0.7
  - Not deterministic ⇒ Ran each experiment 5 times
  - 5 queries to oracle per iteration
  - 10 iterations overall
  - 2 populations sizes: 20 and 100
  - 50 generations between iterations
- Two real-world and three synthetic datasets
- Single thread of a server (JDK1.7, Ubuntu 10.0.4, AMD Opteron 2GHz, 2GB/Experiment)

- Ran experiments on DBLP-ACM
- Population = 20
- $r \in \{2, 4, 8, 16, 32\}$



- Ran experiments on DBLP-ACM
- Population = 20
- $ec \in \{1, 2, 3, 4, 5\}$

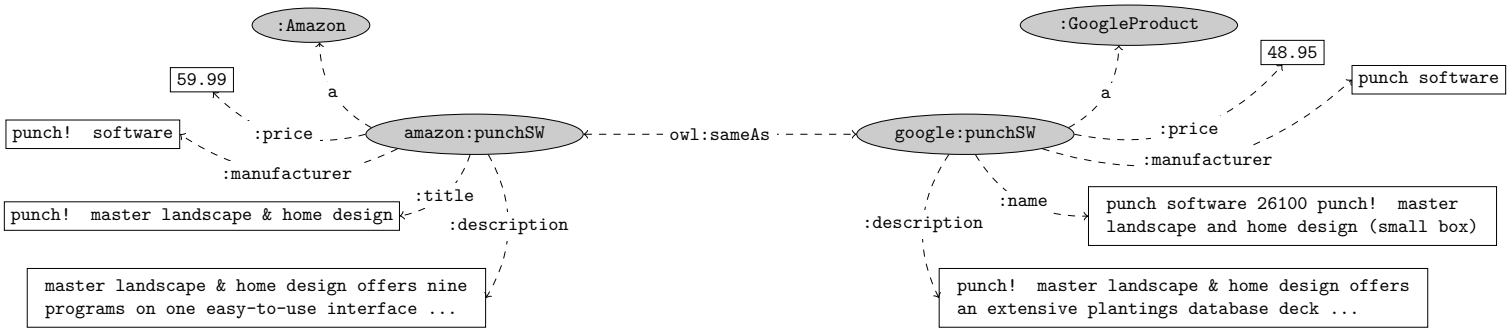


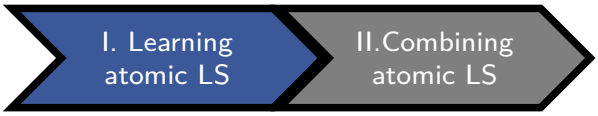
- Population = 100, final values
- Better results, yet unclear when to use *WD* or *CL*

DataSet	**EAGLE**	**WD**	**CL**
Abt	0.19±0.04	**0.25±0.04**	0.23±0.04
DBLP	0.91±0.03	**0.96±0.01**	0.96±0.02
Person1	0.86±0.02	**0.89±0.01**	0.81±0.18
Person2	0.74±0.03	0.71±0.08	**0.77±0.03**
Restaurant	**0.89±0.0**	0.86±0.02	**0.89±0.0**

Table of Contents

- Need for automatic LD for evolving datasets
- Mostly positive examples on the Web of Data
- Negative examples rarely to be found
- Missing links cannot be regarded as negative examples (Open World Assumption)




I. Learning
atomic LS
II.Combining
atomic LS

Goal: Derive a set of initial atomic LS

1. Compute the subset of properties with sufficient coverage
2. Return as many mappings as property pairs with highest possible F-measure

(*euclidean*(x.price, y.price), 0.90)

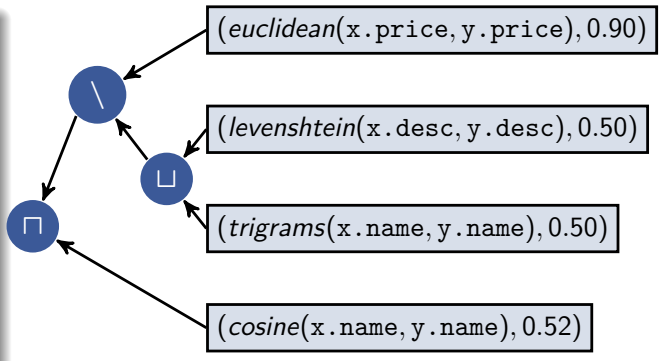
(*levenshtein*(x.desc, y.desc), 0.50)

(*trigrams*(x.name, y.name), 0.50)

(*cosine*(x.name, y.name), 0.52)

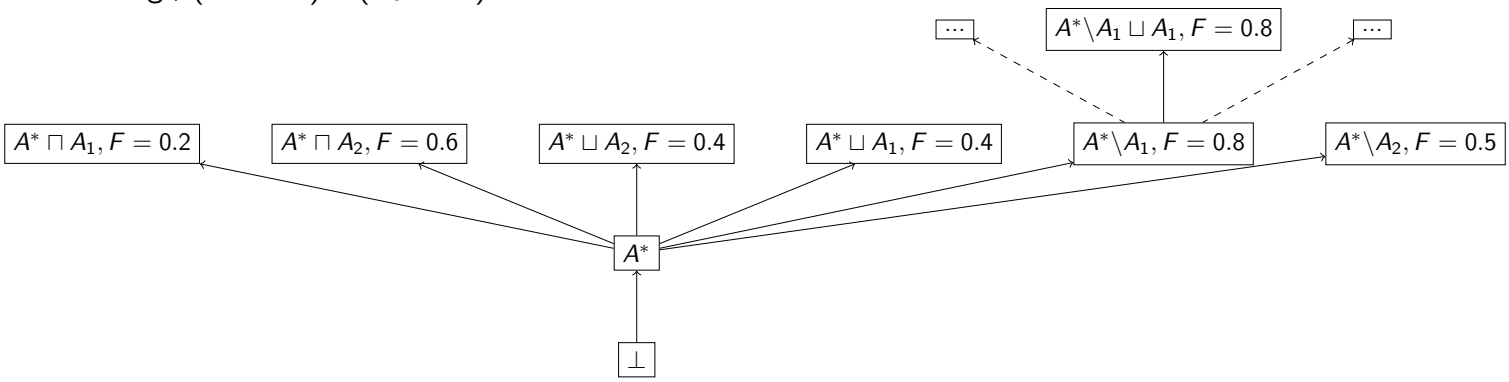
Goal: Derive a set of complex LS

1. Input: set of atomic LSs
2. Use $\sqcap$, $\sqcup$, $\setminus$ to append further atomic LS
3. Compute complex LS by using an approach based on generalisation operators
4. Perform an iterative search through a solution space based on a score function
5. WOMBAT uses F-measure as the score function



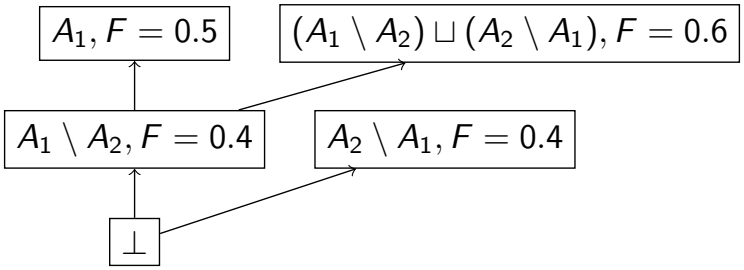
- The simple operator (φ)
  - is not a refinement operator
  - allows efficient implementation
  - can not reach all specifications
    e.g., $(A_1 \sqcup A_2) \sqcap (A_3 \sqcup A_4)$

$$\varphi(L)=\begin{cases}\bigcup_{i=1}^{n} A_i = A^* & \text{if } L=\bot \\ \left(\bigcup_{i=1}^{n} L \sqcup A_i\right) \cup \left(\bigcup_{i=1}^{n} L \sqcap A_i\right) \cup \left(\bigcup_{i=1}^{n} L \setminus A_i\right) & \text{otherwise}\end{cases}$$



- The complete operator (ψ)
  - uses a more sophisticated expansion strategy
  - allows learning arbitrarily nested LS
  - is an upward complete refinement operator
  - is improved using pruning

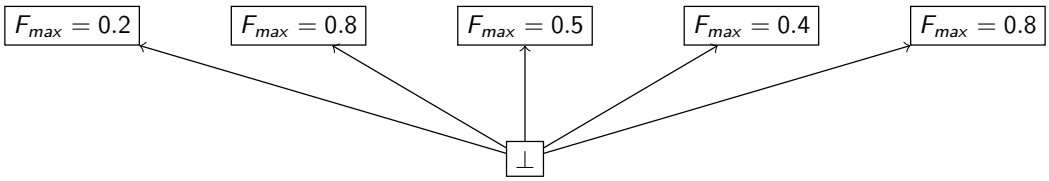
$$\psi(L)=\begin{cases}\{A_{i_1}\setminus A_{j_1}\sqcap\cdots\sqcap A_{i_m}\setminus A_{j_m}\,|\,A_{i_k},A_{j_k}\in\mathbf{A} \\ \quad \text{for all } 1\leq k\leq m\} & \text{if } L=\bot \\ \{L\sqcup A_i\setminus A_j\,|\,A_i\in\mathbf{A},A_j\in\mathbf{A}\} & \text{if } L\in A \\ \{L_1\}\cup\{L\sqcup A_i\setminus A_j\,|\,A_i\in\mathbf{A},A_j\in\mathbf{A}\} & \text{if } L=L_1\setminus L_2 \\ \{L_1\sqcap\cdots\sqcap L_{i-1}\sqcap L'\sqcap L_{i+1}\sqcap\cdots\sqcap L_n\,|\,L'\in\psi(L_i)\} & \\ \quad\cup\{L\sqcup A_i\setminus A_j\,|\,A_i\in\mathbf{A},A_j\in\mathbf{A}\} & \text{if } L=L_1\sqcap\cdots\sqcap L_n (n\geq 2) \\ \{L_1\sqcup\cdots\sqcup L_{i-1}\sqcup L'\sqcup L_{i+1}\sqcup\cdots\sqcup L_n\,|\,L'\in\psi(L_i)\} & \\ \quad\cup\{L\sqcup A_i\setminus A_j\,|\,A_i\in\mathbf{A},A_j\in\mathbf{A}\} & \text{if } L=L_1\sqcup\cdots\sqcup L_n (n\geq 2)\end{cases}$$



ψ is an upward refinement operator

The set of links generated by a child node is a superset of or equal to the set of links generated by its parent

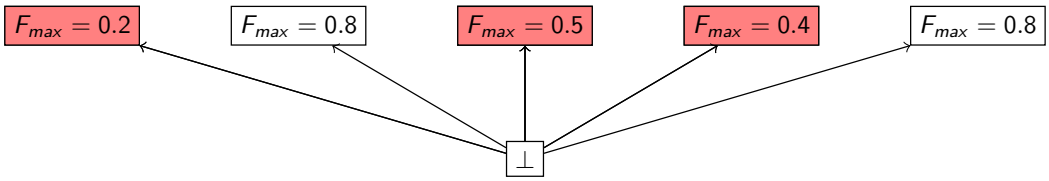
- r_{max} is bounded by the most general constructable LS
- p_{max} is bounded as false positives cannot disappear during generalisation
- $F_{max} = \frac{2 p_{max} r_{max}}{p_{max} + r_{max}}$
- Prune all nodes in the search tree with $F_{max} < F_{best}$



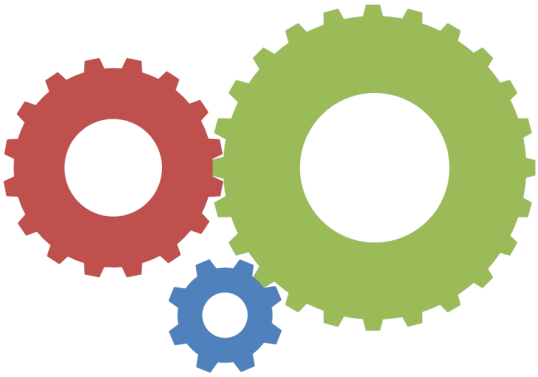
ψ *is an upward refinement operator*

The set of links generated by a child node is a superset of or equal to the set of links generated by its parent

- r_{max} is bounded by the most general constructable LS
- p_{max} is bounded as false positives cannot disappear during generalisation
- $F_{max} = \frac{2 p_{max} r_{max}}{p_{max} + r_{max}}$
- Prune all nodes in the search tree with $F_{max} < F_{best}$



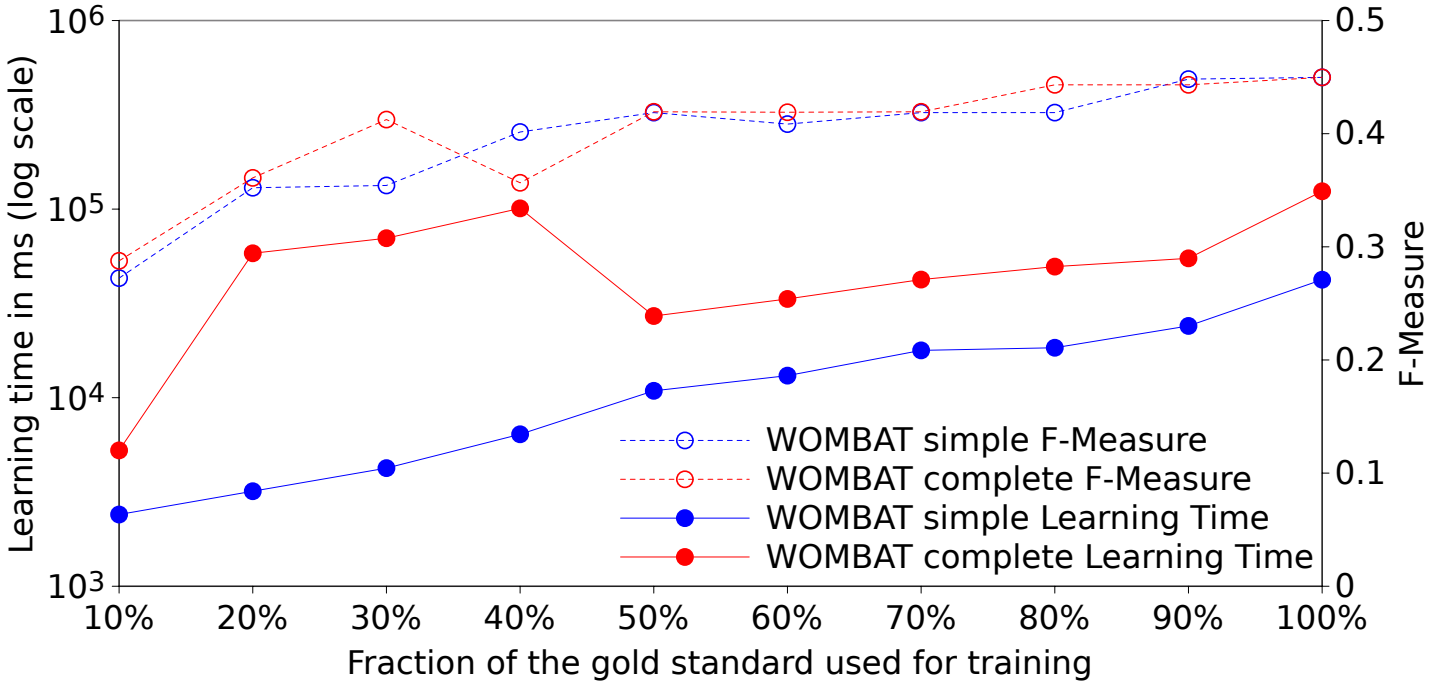
- 8 benchmark datasets (5 real-world, 3 synthetic)
- 2.80 GHz PC running OpenJDK 64-Bit Server 1.7.0_75 on Ubuntu 14.04.2 LTS
- 7 GB RAM
- WOMBAT
  - Similarity measures: `jaccard`, `trigrams`, `cosine` and `qgrams`
  - Termination: $F = 1$ or max number of refinement tree depth of 10
  - Properties coverage threshold = 0.6



Wombat

Evaluation: 10-Fold Cross Validation F-Measure

Dataset	Wombat Simple	Wombat Complete	Euclid Linear	Euclid Conjunction	Euclid Disjunction	Eagle
Person 1	**1.00**	**1.00**	0.64	0.97	**1.00**	0.99 ± 0.004
Person 2	**1.00**	0.99	0.22	0.78	0.96	0.94 ± 0.032
Restaurants	**0.98**	0.97	0.97	0.97	0.97	0.97 ± 0.024
DBLP-ACM	0.97	**0.98**	**0.98**	**0.98**	**0.98**	$\mathbf{0.98 \pm 0.007}$
Abt-Buy	0.60	0.61	0.06	0.06	0.52	$\mathbf{0.65 \pm 0.025}$
Amazon-GP	0.70	0.67	0.59	0.71	**0.73**	0.71 ± 0.033
DBP-LMDB	0.99	**1.00**	0.99	0.99	0.99	0.99 ± 0.004
DBLP-GS	**0.94**	**0.94**	0.90	0.91	0.91	0.93 ± 0.006
Average	**0.90**	**0.90**	0.67	0.80	0.88	$\mathbf{0.90 \pm 0.017}$


Learning time in ms (log scale)
F-Measure
Fraction of the gold standard used for training
WOMBAT simple F-Measure
WOMBAT complete F-Measure
WOMBAT simple Learning Time
WOMBAT complete Learning Time
10^6
10^5
10^4
10^3
0.5
0.4
0.3
0.2
0.1
0
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%

Dataset	10%	20%	30%	40%	50%	60%	70%	80%
Person 1	1.57	2.13	1.85	2.13	2.13	2.13	2.13	2.13
Person 2	1.29	1.29	1.57	1.57	1.57	1.57	1.57	1.57
Restaurant	1.17	1.45	1.17	1.45	1.45	1.45	1.45	1.45
DBLP-ACM	6.23	5.58	6.79	6.85	6.85	6.85	6.79	6.79
Abt-Buy	3.38	3.00	3.00	3.39	3.39	3.39	1.79	3.39
Amazon-GP	1.14	1.38	1.33	1.37	1.38	1.45	1.54	1.59
DBP-LMDB	1.00	1.86	2.86	1.86	1.86	2.33	2.36	2.36
DBLP-GS	1.79	1.93	2.01	2.36	2.45	1.66	2.44	2.26

Pruning factor: $\frac{\text{number of searched nodes (search tree size + pruned nodes)}}{\text{Max. size of the search tree (2000 nodes in this set of experiments)}}$

Wombat

Evaluation: Training with only 2%

Dataset	Pessimistic	Re-weighted	Simple	Complete
Persons 1	**1.00**	**1.00**	**1.00**	**1.00**
Persons 2	0.97	**1.00**	0.80	0.84
Restaurants	0.95	0.94	**0.98**	0.88
DBLP-ACM	0.93	**0.95**	0.94	0.94
Amazon-GP	0.39	0.43	**0.53**	0.45
Abt-Buy	0.36	**0.37**	**0.37**	0.36
Average	0.77	**0.78**	0.77	0.74

- Conclusion
  - Presented WOMBAT, the first approach to learn LS from positive examples
  - WOMBAT is based on generalisation over the space of LS
  - Presented 2 operators to achieve this goal
  - Evaluated WOMBAT against SOTA
  - WOMBAT outperforms SOTA by 11% on average
- Future work
  - Parallelize WOMBAT
  - Try more aggressive pruning techniques for better scalability
  - Apply active learning strategies
  - Unsupervised WOMBAT

Table of Contents

- Large number of challenges to learning accurate specifications
  1. Reduce labeling effort
     ⇒ Active learning
  2. Learn complex specifications
     ⇒ Genetic programming
  3. Learn specifications efficienty
     ⇒ See previous slides
- Challenges include
  1. Determinism
  2. Deep learning
  3. Self-checking
  4. . . .

HOBBIT
Holistic Benchmarking
of Big Linked Data

This work was supported by grants from the EU H2020 Framework Programme provided for the project HOBBIT (GA no. 688227).

Link Discovery Tutorial

Part IV: Benchmarking

Mohamed Ahmed Sherif[1], Irini Fundulaki[2], Axel-Cyrille Ngonga Ngomo[1]

(1) Institute for Applied Informatics, Germany
(2) FORTH, Greece

May 28th, 2017
Portoroz, Slovenia.

Instance matching research has led to the development of various systems.

- What are the problems that I wish to solve?
- What are the relevant key performance indicators?
- What is the behavior of the existing engines w.r.t. the key performance indicators?

Which are the tool(s) that I should use for my data and for my use case?

Benchmarks exist

- To allow adequate measurements of systems
- To provide evaluation of engines for real (or close to real) use cases

Provide help

- Designers and Developers to assess the performance of their tools
- Users to compare the different available tools and evaluate suitability for their needs
- Researchers to compare their work to others

Leads to improvements:

- Vendors can improve their technology
- Researchers can address new challenges
- Current benchmark design can be improved to cover new necessities and application domains

Instance Matching Benchmark comprises of

- **Datasets:** The raw material of the benchmarks. These are the source and the target dataset that will be matched together to find the links
- **Test Cases:** Address heterogeneities (structural, value, semantic) of the datasets to be matched
- **Gold Standard (Ground Truth / Reference Alignment)**: The "correct answer sheet" used to judge the completeness and soundness of the instance matching algorithms
- **Metrics**: The performance metric(s) that determine the systems behaviour and performance

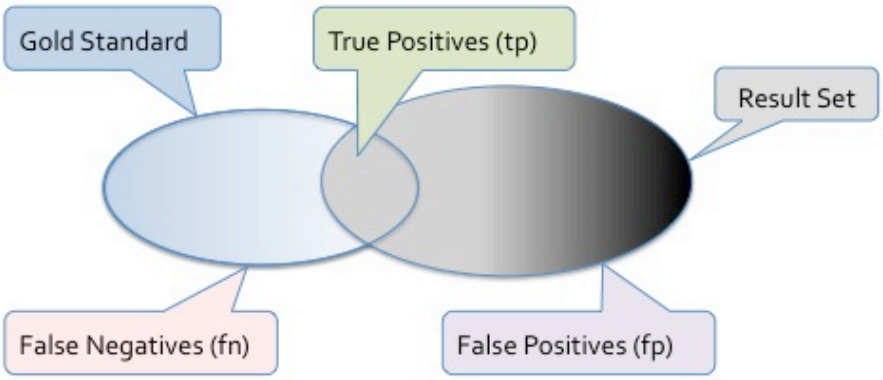
- **Nature**
  - **Real Datasets:** Widely used datasets from a domain of interest
    - Realistic conditions for heterogeneity problems
    - Realistic distributions
    - Error prone, hard to create Reference Alignment
  - **Synthetic Datasets:** Produced with a data generator (that hopefully produces data with interesting characteristics
    - Fully controlled test conditions
    - Accurate, Easy to create Reference Alignments
    - Unrealistic distributions
    - Systematic heterogeneity problems
- **Schema**
  - Datasets to be matched have the **same** or **different schemas**
- **Domain**
  - Datasets come from the **same** or **different** domains

- **Value**
  - Name style abbreviations, Typographical errors, change format (date/gender/number), synonym change, language change (multilinguality)
- **Structural**
  - Change property depth, Delete/add property, split property values, transformation of object/data to data/object type property
- **Semantics**
  - class deletion/modification, invert property assertions, change class/property hierarchy, assert class disjointness
- **Combinations of Variations**

Gold Standard

- The "correct answer sheet" used to judge the completeness and soundness of the instance matching algorithms
- Characteristics
  - Existence of errors / missing alignments
  - Representation: `owl:sameAs` and `skos:exactMatch`

- Precision $P = \frac{tp}{(tp+fn)}$
- Recall $R = \frac{tp}{(tp+fp)}$
- F-measure $F = 2 \times P \times \frac{R}{(P+R)}$



Systematic Procedure	matching tasks should be reproducible and the execution must be comparable
Availability	benchmark should be available
Quality	precise evaluation rules and high quality ontologies must be provided
Equity	evaluation process should not privilege any system
Dissemination	benchmark should be used to evaluate instance matching systems
Volume	dataset size
Gold Standard	gold standard should exist and be as accurate as possible

Instance matching techniques have, until recently, been benchmarked in an ad-hoc way.

There is no standard way of benchmarking the performance of the systems, when it comes to Linked Data.

- IM benchmarks have been mainly driven forward by the **Ontology Alignment Evaluation Initiative (OAEI)**
  - organizes annual campaign for ontology matching since 2005
  - hosts independent benchmarks
- In 2009, OAEI introduced the **Instance Matching (IM) Track**
  - focuses on the evaluation of different instance matching techniques and tools for Linked Data

Instance Matching Benchmarks

- Bechmark Generators
- Synthetic Benchmarks
- Real Benchmarks

Semi automatic generator of Instance Matching Benchmarks

- Contributed in the generation of IIMB Benchmarks of OAEI in 2010, 2011 and 2012 Instance Matching Tracks
- Freely available at (`https://code.google.com/p/swing-generator/`)
- All kind of variations supported into the benchmarks except multilinguality
- Automatically produced gold standard

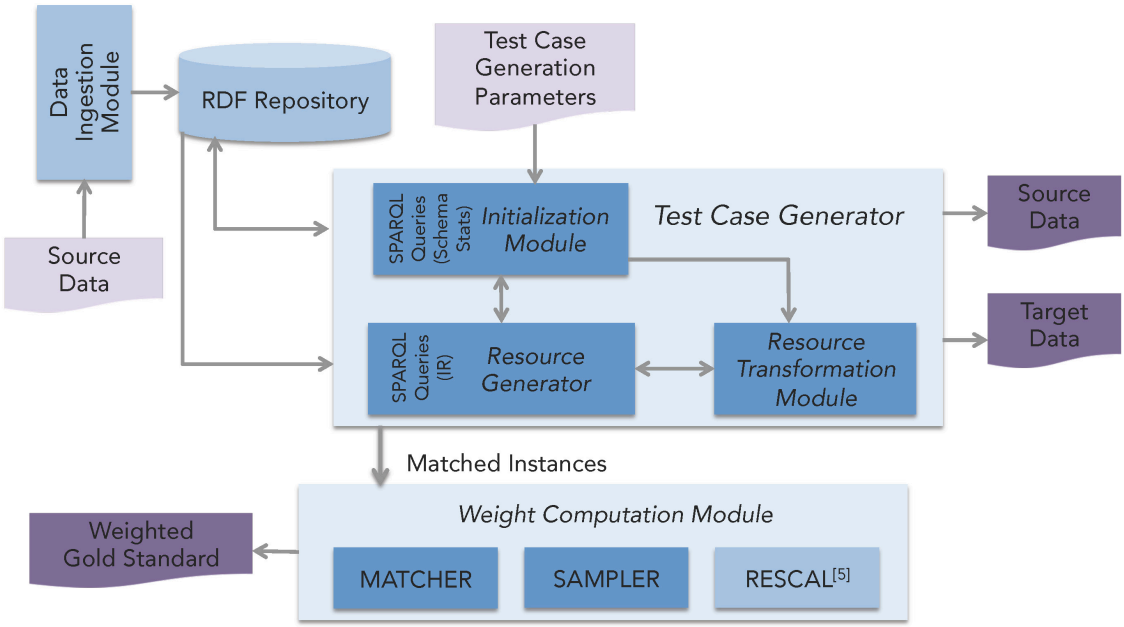
Flexible, generic and domain-independent benchmark generator which takes into consideration RDFS and OWL constructs in order to evaluate instance matching systems.

Lance [SDF+15b]

Lance provides support for:

- Semantics-aware transformations
  - Complex class definitions (union, intersection)
  - Complex property definitions (functional properties, inverse functional properties)
  - Disjointness (properties)
- Standard value and structure based transformations
- Weighted gold standard based on tensor factorization
- Varying degrees of difficulty and fine-grained evaluation metrics

Available at `http://github.com/jsaveta/Lance`


Data Ingestion Module
RDF Repository
Test Case Generation Parameters
Source Data
SPARQL Queries (Schema Stats)
Initialization Module
Test Case Generator
Source Data
Target Data
SPARQL Queries (IR)
Resource Generator
Resource Transformation Module
Matched Instances
Weight Computation Module
Weighted Gold Standard
MATCHER
SAMPLER
RESCAL[5]

Ontology Alignment Evaluation Benchmarks

OAEI RDFT 2013 (French / English)

ID-REC 2014 (English/Italian)

Author –Task 2015 (English/Italian)

Synthetic IM Benchmarks

Overview (1)

	IIMB 2009	IIMB 2010	PR 2010	IIMB 2011	Sandbox 2012	IIMB 2012	RDFT 2013	ID-REC 2014	Author Task 2015
Systematic Procedure	√	√	√	√	√	√	√	√	√
Availability	√	√	√	√	√	-	-	√	√
Quality	√	√	√	√	√	√	√	√	√
Equity	√	√	√	√	√	√	√	√	√
Dissemination	6	3	6	1	3	4	4	5	5
Volume	0.2K	1.4K	0.86K	4K	0.375K	1.5K	0.43K	2.650K	10K
Gold Standard	√	√	√	√	√	√	√	√	√

Synthetic IM Benchmarks

Overview (2)

	IIMB 2009	IIMB 2010	PR 2010	IIMB 2011	Sandbox 2012	IIMB 2012	RDFT 2013	ID-REC 2014	Author Task 2015
Value Variations	√	√	√	√	√	√	√	√	√
Structural Variations	√	√	√	√	-	-	-	+	+
Logical Variations	√	√	-	√	-	√	-	-	-
Multilinguality	-	-	-	-	-	-	√	√	√

	IIMB 2009	IIMB 2010	PR 2010	IIMB 2011	Sandbox 2012	IIMB 2012	RDFT 2013	ID-REC 2014	Author Task 2015
Blind Evaluations	-	-	-	-	-	-	√	√	√
1-n Mappings	-	-	√	-	-	-	√	√	-

Synthetic IM Benchmarks

Overview (3)

	IIMB 2009	IIMB 2010	PR 2010	IIMB 2011	Sandbox 2012	IIMB 2012	RDFT 2013	ID-REC 2014	Author Task 2015	Lance 2015
Systematic Procedure	√	√	√	√	√	√	√	√	√	√
Availability	√	√	√	√	√	-	-	√	√	√
Quality	√	√	√	√	√	√	√	√	√	√
Equity	√	√	√	√	√	√	√	√	√	√
Dissemination	6	3	6	1	3	4	4	5	5	2
Volume	0.2K	1.4K	0.86K	4K	0.375K	1.5K	0.43K	2.650K	10K	$> 1M$
Gold Standard	√	√	√	√	√	√	√	√	√	√

Synthetic IM Benchmarks

Overview (4)

	IIMB 2009	IIMB 2010	PR 2010	IIMB 2011	Sandbox 2012	IIMB 2012	RDFT 2013	ID-REC 2014	Author Task 2015	Lance 2015
Value Variations	√	√	√	√	√	√	√	√	√	√
Structural Variations	√	√	√	√	-	-	-	+	+	+
Logical Variations	√	√	-	√	-	√	-	-	-	+
Multilinguality	-	-	-	-	-	-	√	√	√	√

Synthetic IM Benchmarks

Overview (5)

	IIMB 2009	IIMB 2010	PR 2010	IIMB 2011	Sandbox 2012	IIMB 2012	RDFT 2013	ID-REC 2014	Author Task 2015	Lance 2015
Blind Evaluations	-	-	-	-	-	-	√	√	√	√
1-n Mappings	-	-	√	-	-	-	√	√	-	-

ARS

(OAEI 2009)

Data Interlinking

(OAEI 2010)

Data Interlinking – New York Times

(OAEI 2011)

	ARS	DI 2010	DI 2011
Systematic Procedure	√	√	√
Availability	√	√	-
Quality	√	√	√
Equity	√	√	√
Dissemination	5	2	3
Volume	100K	6K	NA
Gold Standard	√	√	+

Real IM Benchmarks

Overview (2)

	ARS	DI 2010	DI 2011
Value Variations	√	√	√
Structural Variations	√	√	-
Logical Variations	-	-	-
Multilinguality	-	-	-
Blind Evaluations	-	-	-

Multilinguality

OAEI RDFT 2013 (French / English)

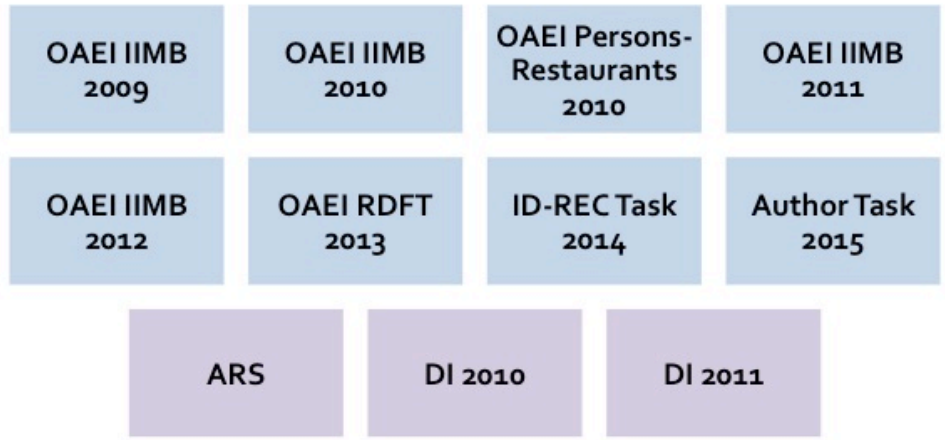
OAEI ID-REC 2014 (English/Italian)

OAEI Author –Task 2015 (English/Italian)

Value Variations

OAEI IIMB 2009	OAEI IIMB 2010	OAEI Persons-Restaurants 2010	OAEI IIMB 2011
Sandbox 2012	OAEI IIMB 2012	OAEI RDFT 2013	ID-REC Task 2014
Author Task 2015	ARS	DI 2010	DI 2011

Structural Variations



Logical Variations

OAEI IIMB 2009

OAEI IIMB 2010

OAEI IIMB 2011

OAEI IIMB 2012

Combinations of Variations

OAEI IIMB 2009

OAEI IIMB 2010

OAEI IIMB 2011

OAEI IIMB 2012

OAEI RDFT 2013

ID-REC Task 2014

Author Task 2015

Scalability

ARS

DI 2010

DI 2011

Scalability & Expressiveness

Open Issues

- Only one benchmark that tackles both, combination of variations and scalability issues
- Not enough IM benchmark using the full expressiveness of RDF/OWL language

- Systems can handle the value variations, the structural variation, and the simple logical variations separately.
- More work needed for complex variations (combination of value, structural, and logical)
- More work needed for structural variations
- Enhancement of systems to cope with the clustering of the mappings (1-n mappings)

- Many instance matching benchmarks have been proposed
- Each of them answering to some of the needs of instance matching systems.
- It is essential to start creating benchmarks that will "show the way to the future"
- Extend the limits of existing systems.

HOBBIT
Holistic Benchmarking
of Big Linked Data

This work was supported by grants from the EU H2020 Framework Programme provided for the project HOBBIT (GA no. 688227).

Link Discovery Tutorial

Part V: Hands-On

Mohamed Ahmed Sherif[1], Irini Fundulaki[2], Axel-Cyrille Ngonga Ngomo[1]

(1) Institute for Applied Informatics, Germany
(2) FORTH, Greece

May 28th, 2017
Portoroz, Slovenia.

Table of Contents

Table of Contents

Semantic Web Dog Food Corpus

Data exposed: Metadata (papers, presentations, people) for several semantic web related conferences and workshops, including the most recent ISWC, ESWC and WWW events.

- Dumps: http://data.semanticweb.org/dumps
- Endpoint: http://data.semanticweb.org/sparql
- DataHub: https://datahub.io/dataset/semantic-web-dog-food

Table of Contents

Task I: Deduplication

Execute the given Configuration File

- Decentralized nature of LOD
- Data contain duplicates
- How to efficiently detect similar resources?

Task I: Deduplication

Execute the given Configuration File

- Decentralized nature of LOD
- Data contain duplicates
- How to efficiently detect similar resources?

Example

Find duplicate authors in Semantic Web Dog Food Corpus?

```
<PREFIX>
  <NAMESPACE>http://www.w3.org/2000/01/rdf-schema#</NAMESPACE>
  <LABEL>rdfs</LABEL>
</PREFIX>
```

LIMES *Configuration File*

SOURCE and TARGET datasets

```
<SOURCE>
    <ID>SDF1</ID>
    <ENDPOINT>semanticDogFood.nt</ENDPOINT>
    <VAR>?x</VAR>
    <PAGESIZE>-1</PAGESIZE>
    <RESTRICTION>?x a foaf:Person</RESTRICTION>
    <PROPERTY>rdfs:label</PROPERTY>
    <TYPE>NT</TYPE>
</SOURCE>
```

```
<METRIC>Levenshtein(x.rdfs:label , y.rdfs:label)</METRIC>
```

LIMES *Configuration File*

ACCEPTANCE and REVIEW

```
<ACCEPTANCE>
    <THRESHOLD>0.9</THRESHOLD>
    <FILE>similarAuthor.nt</FILE>
    <RELATION>ov:similarTo</RELATION>
</ACCEPTANCE>

<REVIEW>
    <THRESHOLD>0.5</THRESHOLD>
    <FILE>similarAuthor_review.nt</FILE>
    <RELATION>owl:sameAs</RELATION>
</REVIEW>
```

```
<OUTPUT>TTL</OUTPUT>
```

Run LIMES

```
java -jar limes-core-1.0.0.jar task1.xml
```

```
<MLALGORITHM>
    <NAME>wombat simple</NAME>
    <TYPE>unsupervised</TYPE>
</MLALGORITHM>
```

Table of Contents

Task II

Find similar publications

1. Find publications
   - with similar keyword,
   - but do not link any publication to itself

1. Find publications
   - with similar keyword,
   - but do not link any publication to itself

1. *Find publications*

- `?x a swrc:InProceedings`
- `?y a swrc:InProceedings`

Task II

Find similar publications

1. Find publications
   - with similar keyword,
   - but do not link any publication to itself

1. *Find publications*

- `?x a swrc:InProceedings`
- `?y a swrc:InProceedings`

2. *with similar keyword, but do not link any publication to itself*

```
MINUS(jaccard(x.swrc:listKeyword , y.swrc:listKeyword)|0.3,
ExactMatch(x.swrc:listKeyword , y.swrc:listKeyword)|1.0)
```

Table of Contents

Run LIMES *GUI*

```
java -jar limes-core-1.0.0.jar -g
```

Table of Contents

This work was supported by grants from the EU H2020 Framework Programme provided for the project HOBBIT (GA no. 688227).