Data & Metadata Alignment

Renée J. Miller
University of Toronto

INFINT 2009

Bertinoro

Talk Overview

- Background - brief (selective) history
  - Entity resolution (data matching)
  - Schema mapping
- Bringing data and metadata alignment together
  - Ontologies
- The ILIADS method
  - First stab at these issues
- Experimental evaluation
  - What works well and what doesn't

From Webster…

Main Entry: align·ment Variant(s): also aline·ment \ə-līn-mənt\ Function: *noun* Date: 1790 1: the act of aligning or state of being aligned; *especially* : the proper positioning or state of adjustment of parts (as of a mechanical or electronic device) in relation to each other

Information alignment: the process of *finding*, *modeling* and *using* the correspondences or connections that place information artifacts in relation to each other

Project Origins

- Alignment of research programs and background
  - Professor Lise Getoor, Univ. Maryland, College Park
    - Machine Learning
    - Expert in entity-resolution, statistical learning
  - Myself
    - Data Management
    - Schema mapping, data exchange

Data Alignment (Matching)

- Detecting instances (tuples or records) that represent the same (semantically related) real world entity
- Very well studied problem with many names: data matching, deduplication, merge/purge problem, entity disambiguation, duplicate detection, record matching, identity uncertainty, instance identification, object identification, co-reference resolution, reference reconciliation, record linkage, database hardening, fuzzy matching, entity resolution…..

Example: Citation Data

L. Breiman, L. Friedman, and P. Stone, (1984). Classification and Regression. Wadsworth, Belmont, CA.

Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. Classification and Regression Trees. Wadsworth and Brooks/Cole, 1984.

R. Agrawal, R. Srikant. Fast algorithms for mining association rules in large databases. In VLDB-94, 1994.

Rakesh Agrawal and Ramakrishnan Srikant. Fast Algorithms for Mining Association Rules In Proc. Of the 20th Int'l Conference on Very Large Databases, Santiago, Chile, September 1994.

Example: Customer Data

- Name: Preetam Maloor
  Address: 18055 Cottage Garden Dr, Germantown, MD

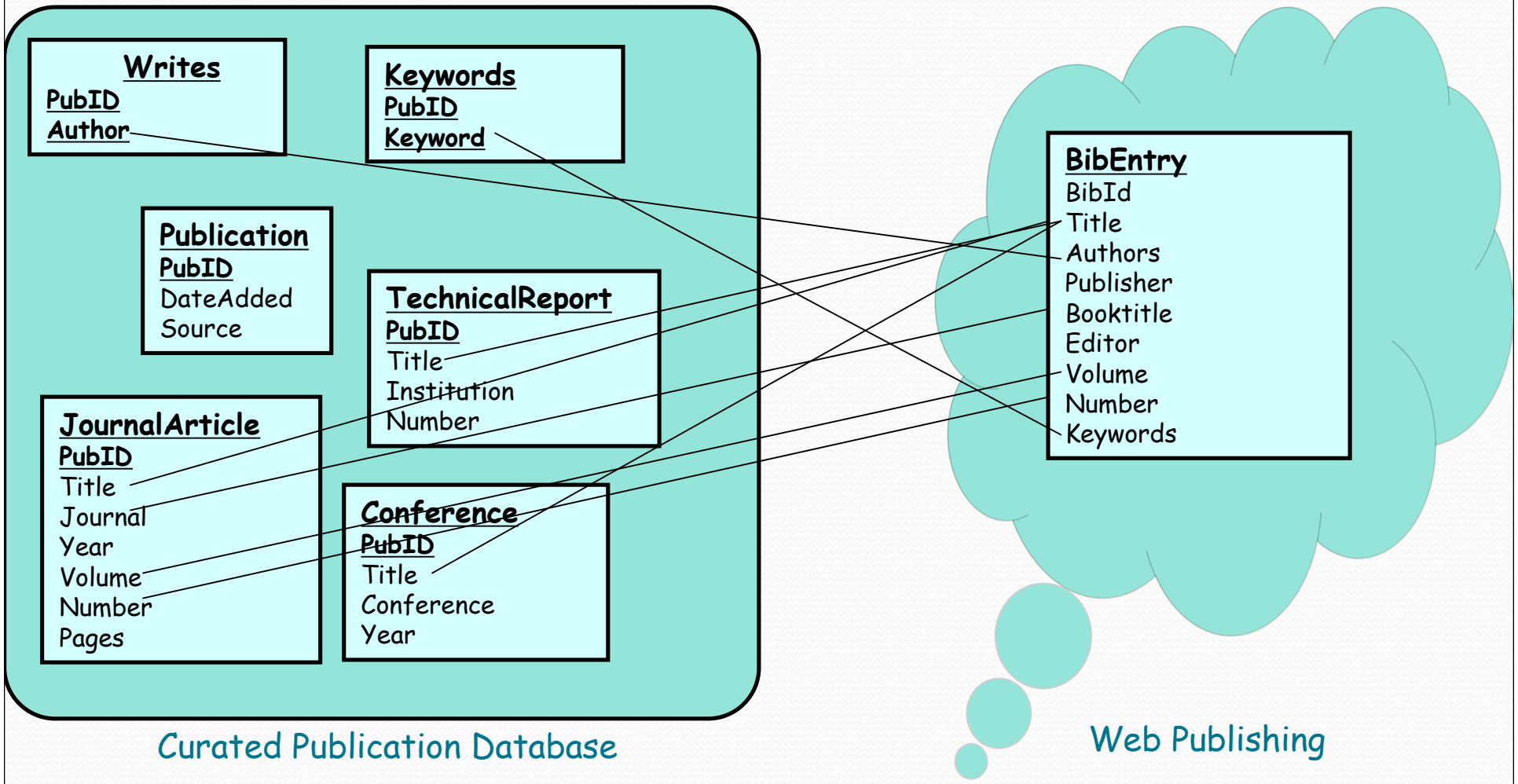
- Name: Maloor, P
  Marital Status: Single
  Occupation: Research Engineer

- Name: Preetam A. Maloor
  Type of Housing: Rent
  Location: Germantown, MD

Schema Alignment (Matching)

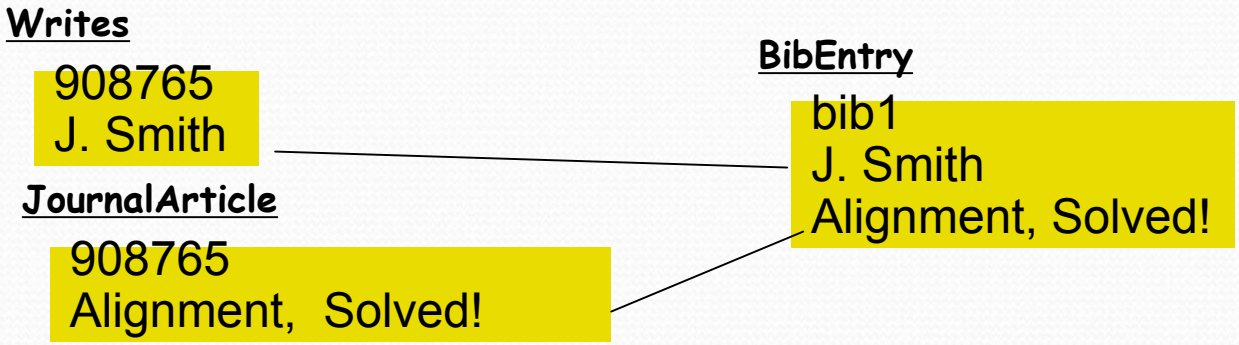
- Detecting schema objects (e.g., attributes) that are semantically related
- Very well studied problem leverages many different inference techniques
- Both data and schema matching make use of similarity measures based on:
  - Lexical similarity, structural similarity, semantic similarity, etc.

Schema Matching Example



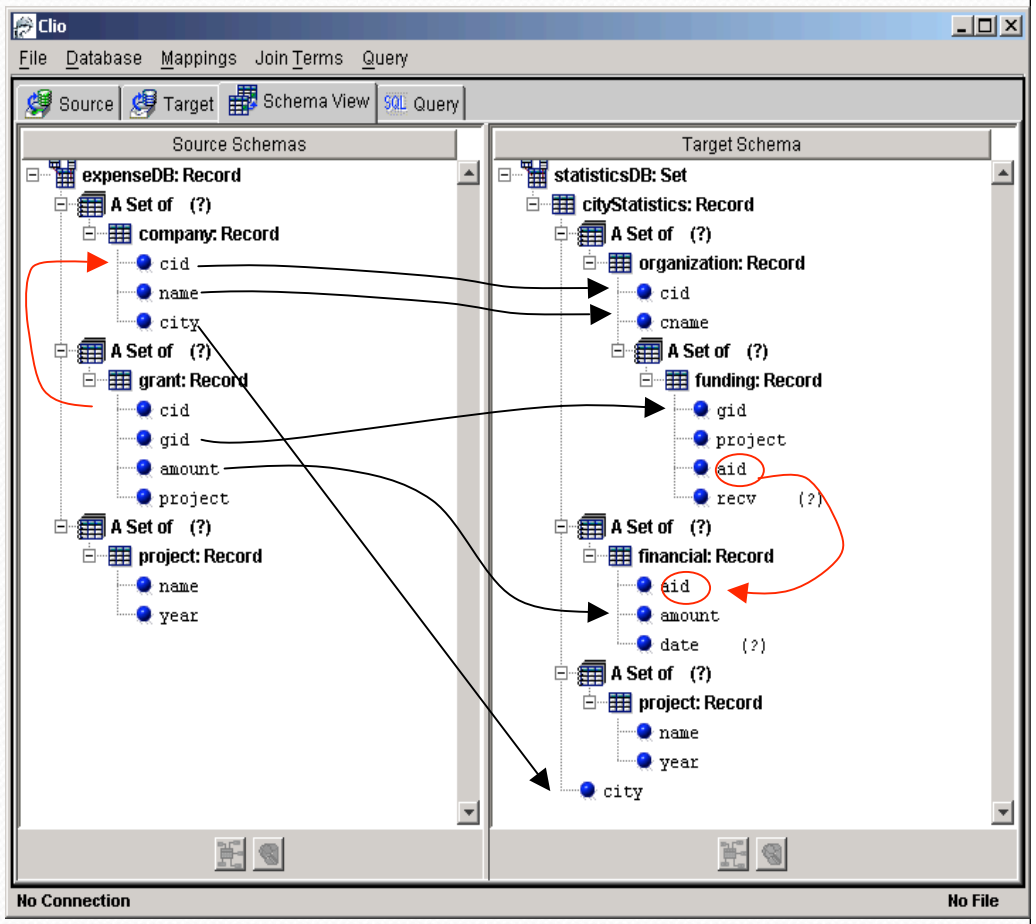
Interpreting Matches

- Suppose we want to translate data from source to target (*data exchange*) or query target data using source schema (*data integration*)
- Matches do not give sufficient information to transform data (maintaining semantics of the data!)
  - How is a BibEntry tuple created?



Mapping Creation (Clio)

- Leverage attribute **matches**
  - User friendly
  - Automatic discovery
- Preserve data **semantics**
  - discover data associations
  - use constraints and schema structure
- Model **incompleteness**
  - generate new values for data exchange
- Produce correct **grouping**

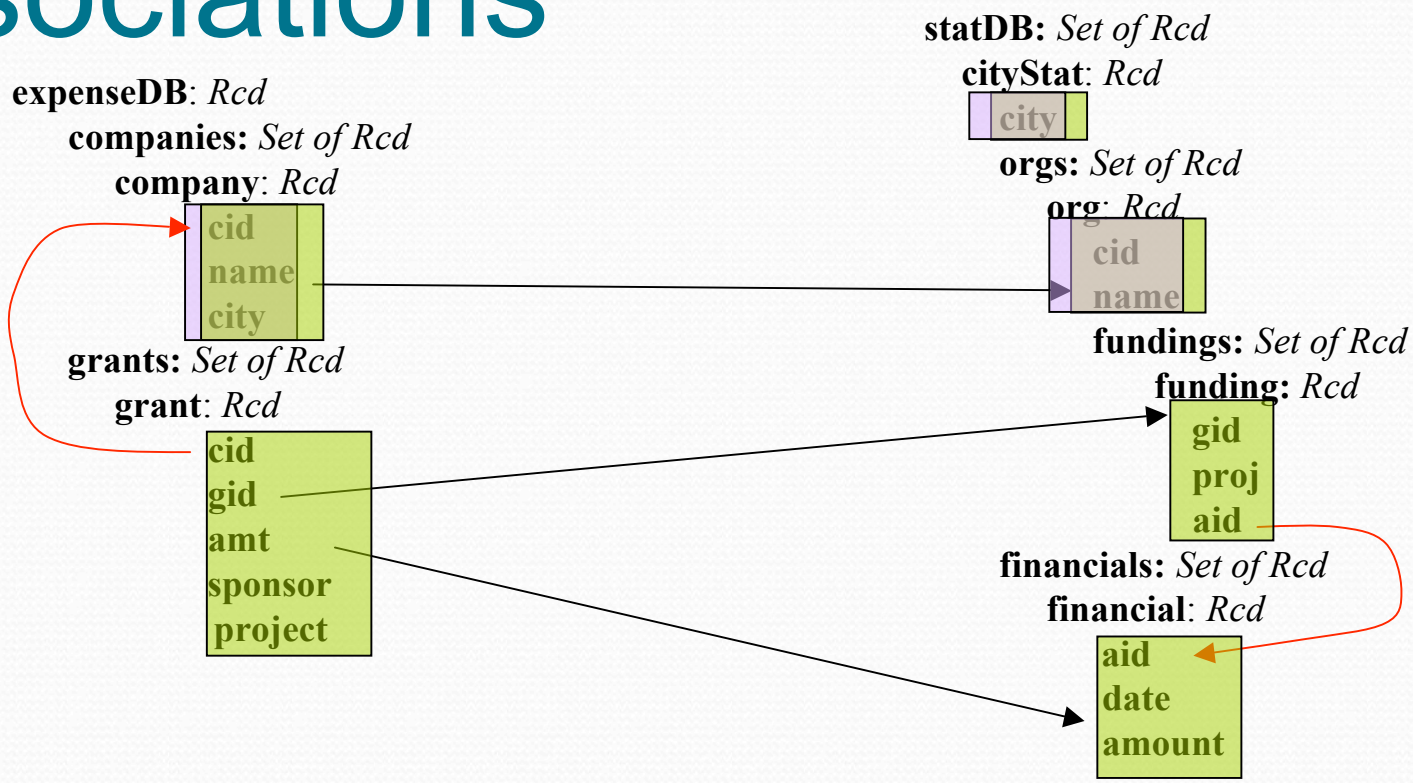


Many meanings of a line: *STBenchmark* Alexe, Tan, Velegrakis, VLDB08

Mapping Generation

- Mapping: $\forall \mathbf{x}\ \varphi_s(\mathbf{x}) \rightarrow \exists \mathbf{y}\ \psi_t(\mathbf{x}, \mathbf{y})$
  - Query on source contained in query on target ($\varphi_s \subseteq \psi_t$)
- How do we find source (φ_s) and target (ψ_t) queries?
  - Use **chase** [Maier, Mendelzon, Sagiv 79] to find connections within the schemas.
  - Originally defined to solve inference problem for relational dependencies.
  - We use it to generate possible alternative representations of information (associations).
    - Generalized to nested-relational model [Popa et al, VLDB02]
    - Generalized to discover grouping and correlation semantics [Fuxman et al, VLDB 06]

Associations



company(CID,N,C)

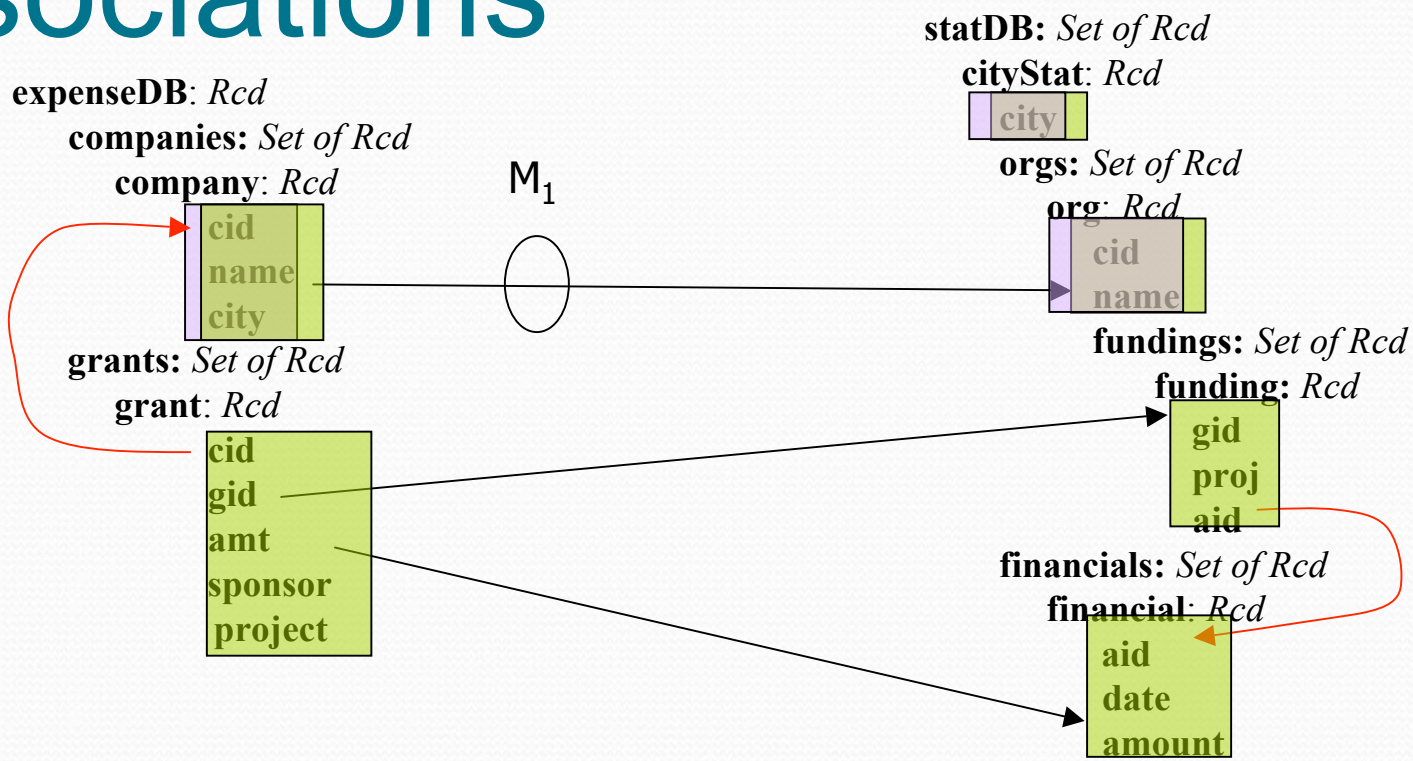
company(CID,N,C),grant(CID,GID,A,S,P)

cityStat(C,Os),Os(CID,N)

cityStat(C,Os),Os(CID,N),
funding(GID,P,AID),
financial(AID,D,A)

Red Arrows are referential constraints like keyrefs in XML

Associations

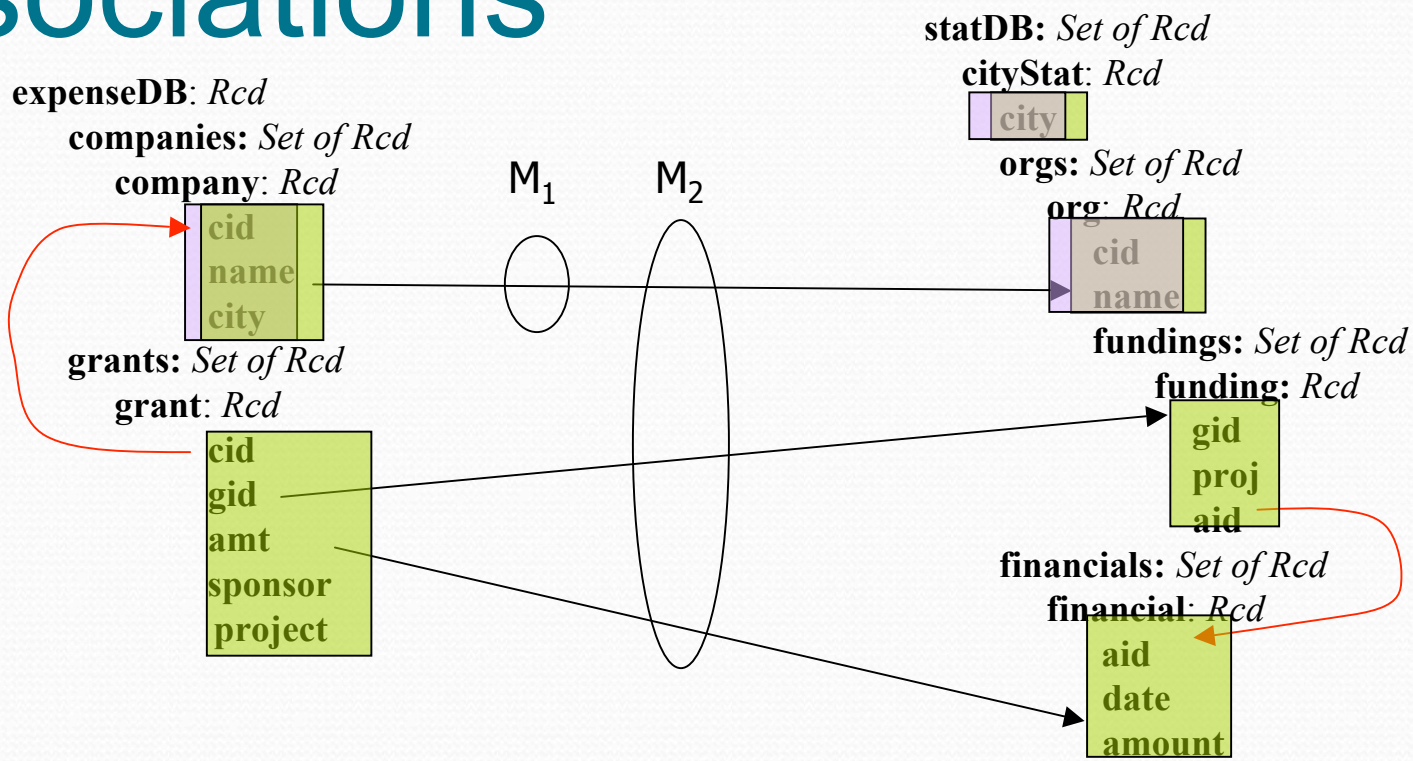


company(CID,N,C) → ∃C′,CID′,Os cityStat(C′,Os),Os(CID′,N)

company(CID,N,C),grant(CID,GID,A,S,P)

cityStat(C,Os),Os(CID,N),
funding(GID,P,AID),
financial(AID,D,A)

Associations



company(CID,N,C) → ∃C′,CID′,Os cityStat(C′,Os),Os(CID′,N)

company(CID,N,C),grant(CID,GID,A,S,P) → ∃C′,CID′,Os, P′,AID,D cityStat(C′,Os),Os(CID′,N), funding(GID,P′,AID′), financial(AID′,D,A)

Talk Overview

- Background - brief (selective) history
- **Bringing data and metadata alignment together**
- The ILIADS method
- Experimental evaluation

Need Wholistic approaches

- Alignment
  - Separate tools used in data and metadata
    - Advantages to be found by leveraging different tools
  - Alignment results often separated from their modeling and use in integration
    - Alignment discovery needs to be aware of how alignments will be modeled and used

Information alignment: the process of *finding*, *modeling* and *using* the correspondences or connections that place information artifacts in relation to each other

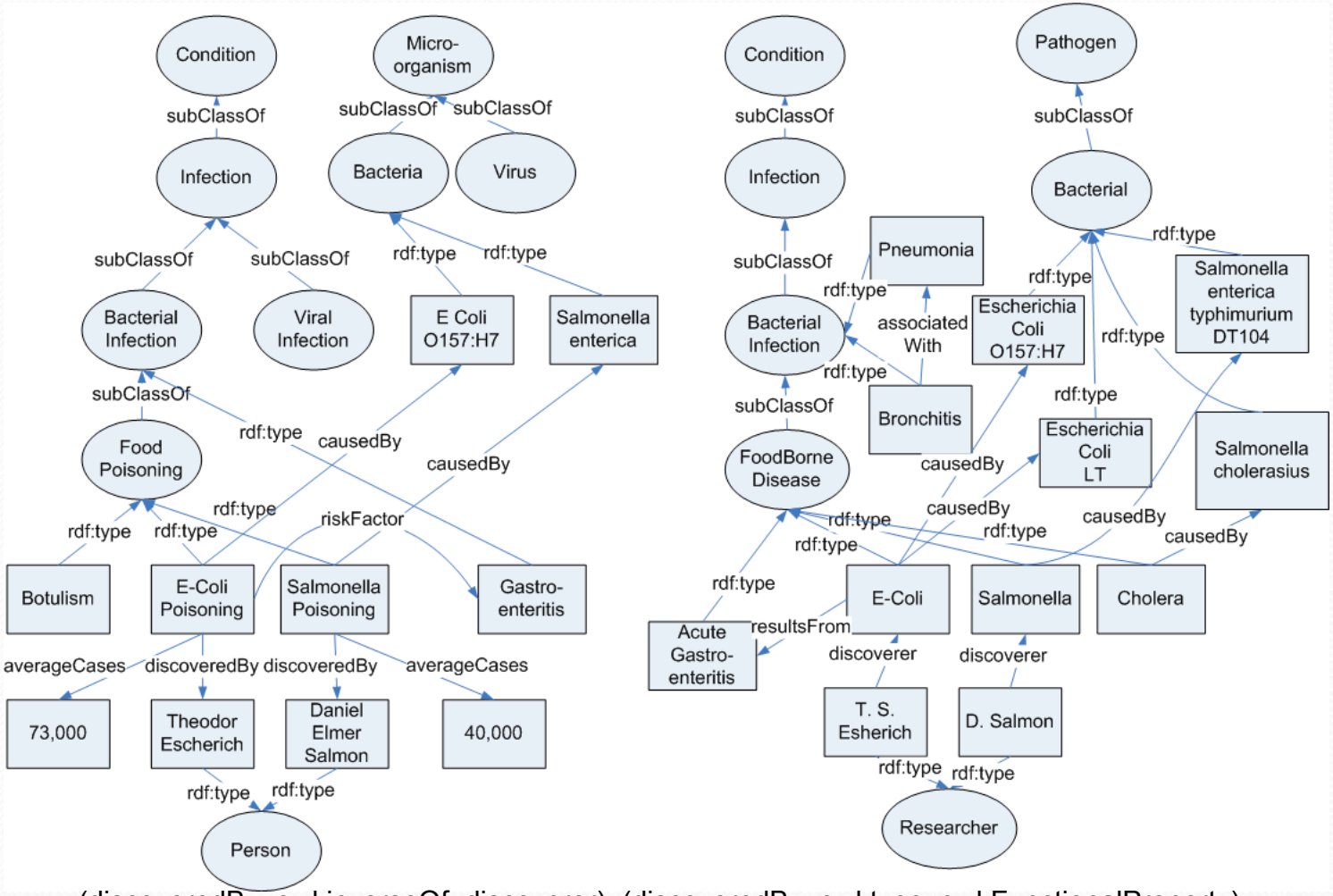
Ontology Alignment

- Ontology
  - Data (instances)
  - Structure
    - Properties
    - Inheritance hierarchy
  - Constraints (axioms)
- Alignment
  - Goal is to leverage entire ontology for better alignment

Defining the terms

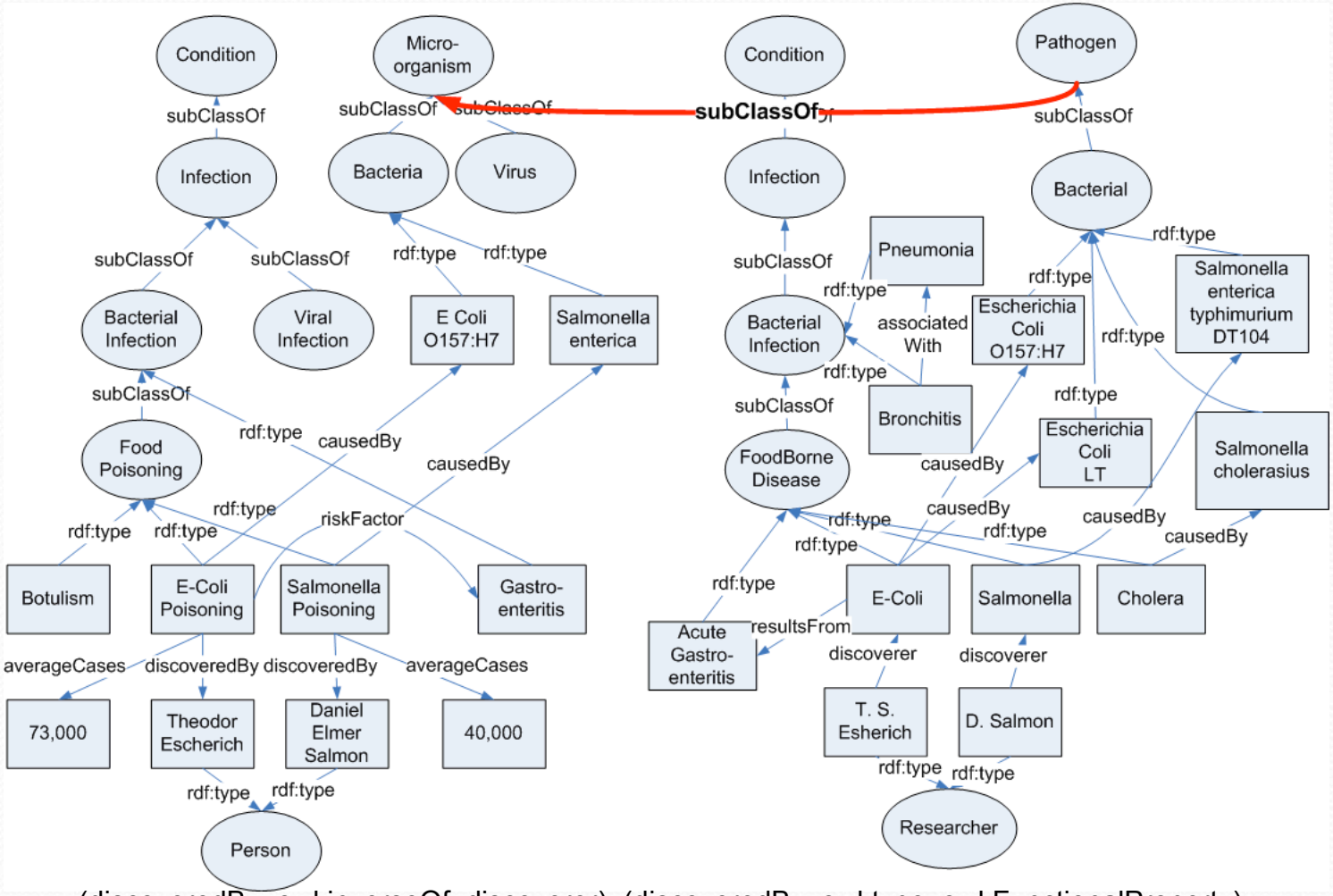
- **Entity:** everything that has an URI identifier (plus literals)
- **Ontology**: software artifact consisting of classes, instances, facts, axioms
- **Alignment:** given two ontologies, find relationships between their respective entities
- **Integration**: merge two ontologies under a set of alignments to obtain a consistent result

Example OWL Lite ontologies



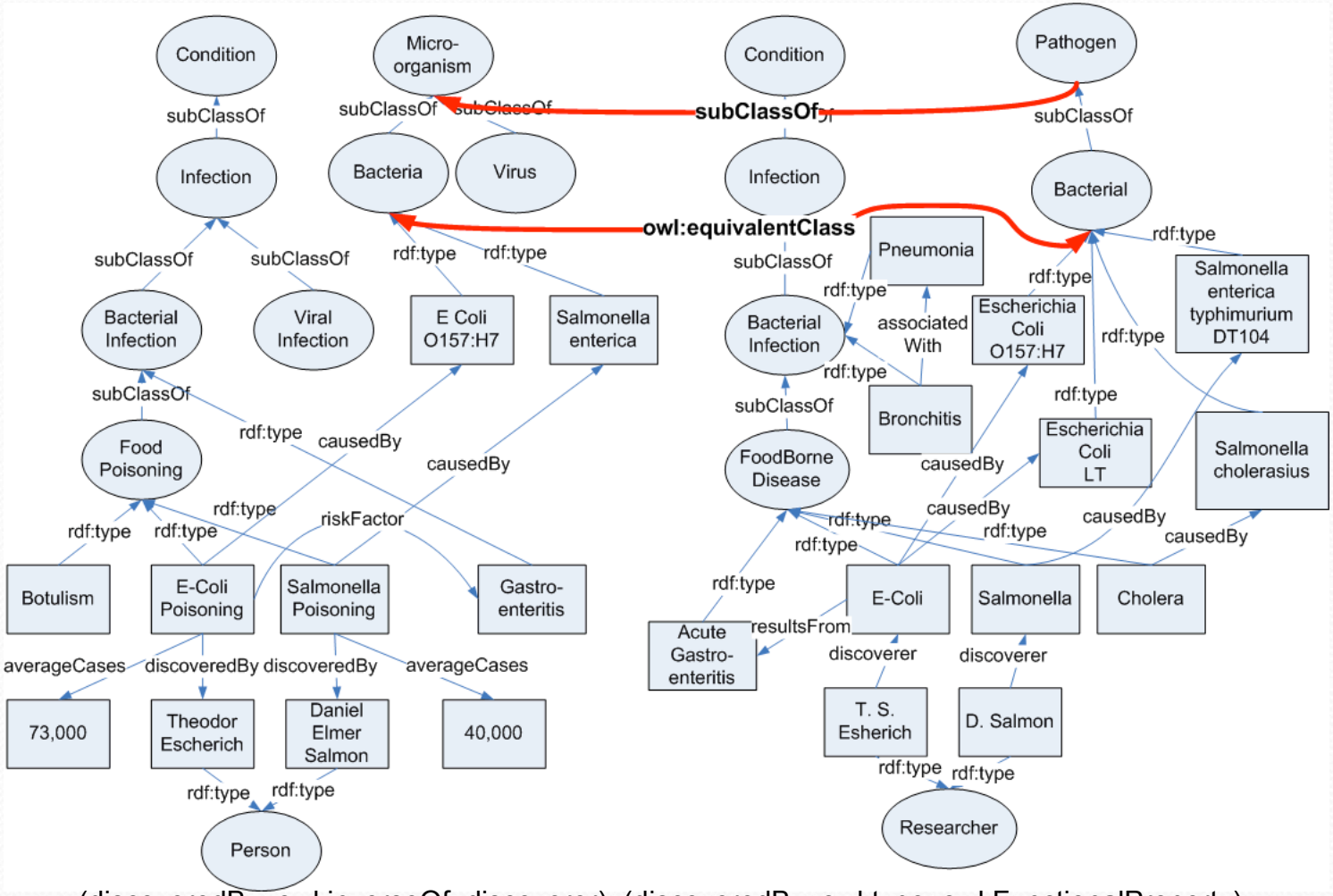
(discoveredBy, owl:inverseOf, discoverer); (discoveredBy, owl:type, owl:FunctionalProperty)
(discoveredBy, owl:inverseOf, discoverer); (associatedWith, owl:type, owl:TransitiveProperty)
(resultsF rom, rdfs:subPropertyOf, associatedWith)

Example alignments



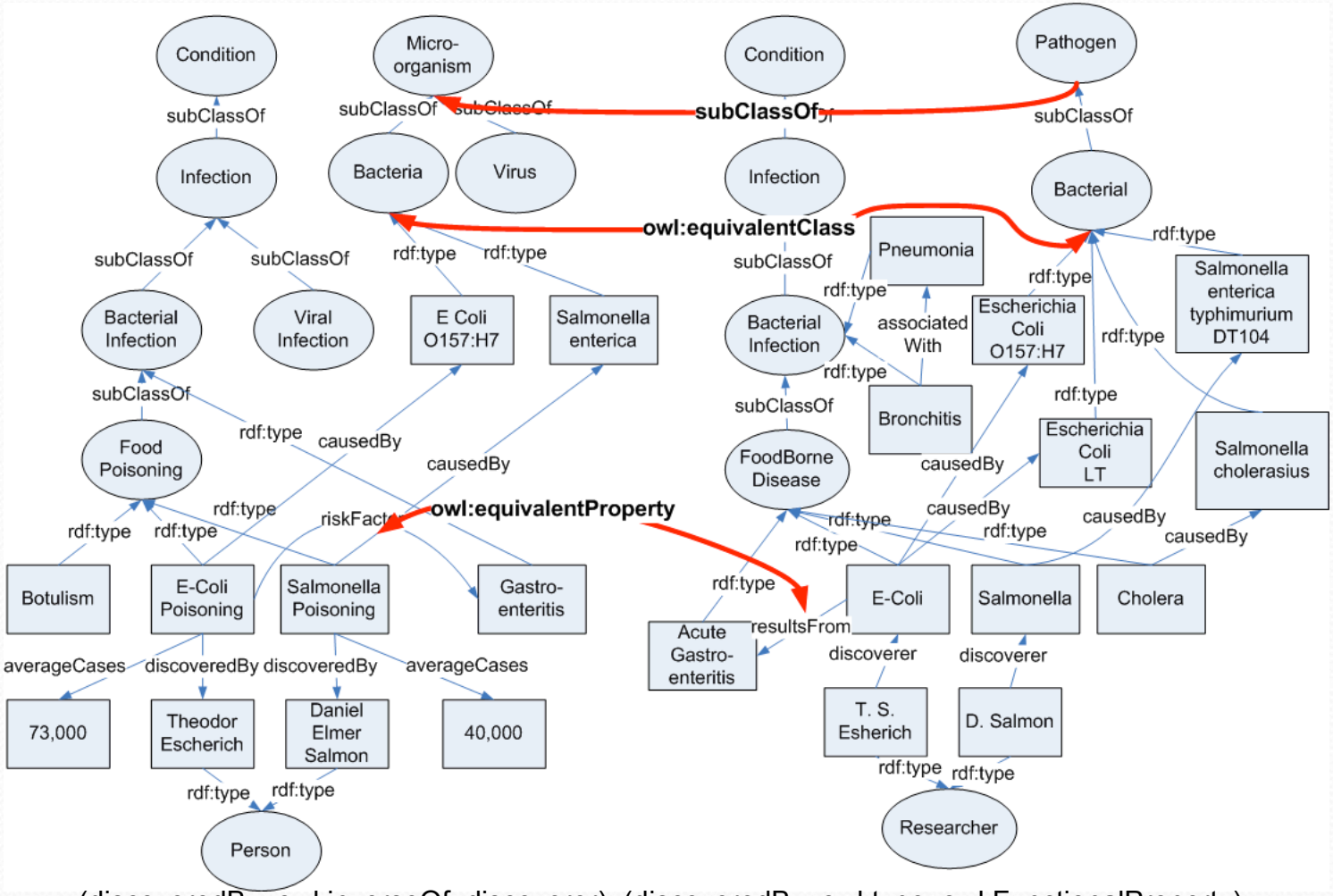
(discoveredBy, owl:inverseOf, discoverer); (discoveredBy, owl:type, owl:FunctionalProperty)
(discoveredBy, owl:inverseOf, discoverer); (associatedWith, owl:type, owl:TransitiveProperty)
(resultsF rom, rdfs:subPropertyOf, associatedWith)

Example alignments



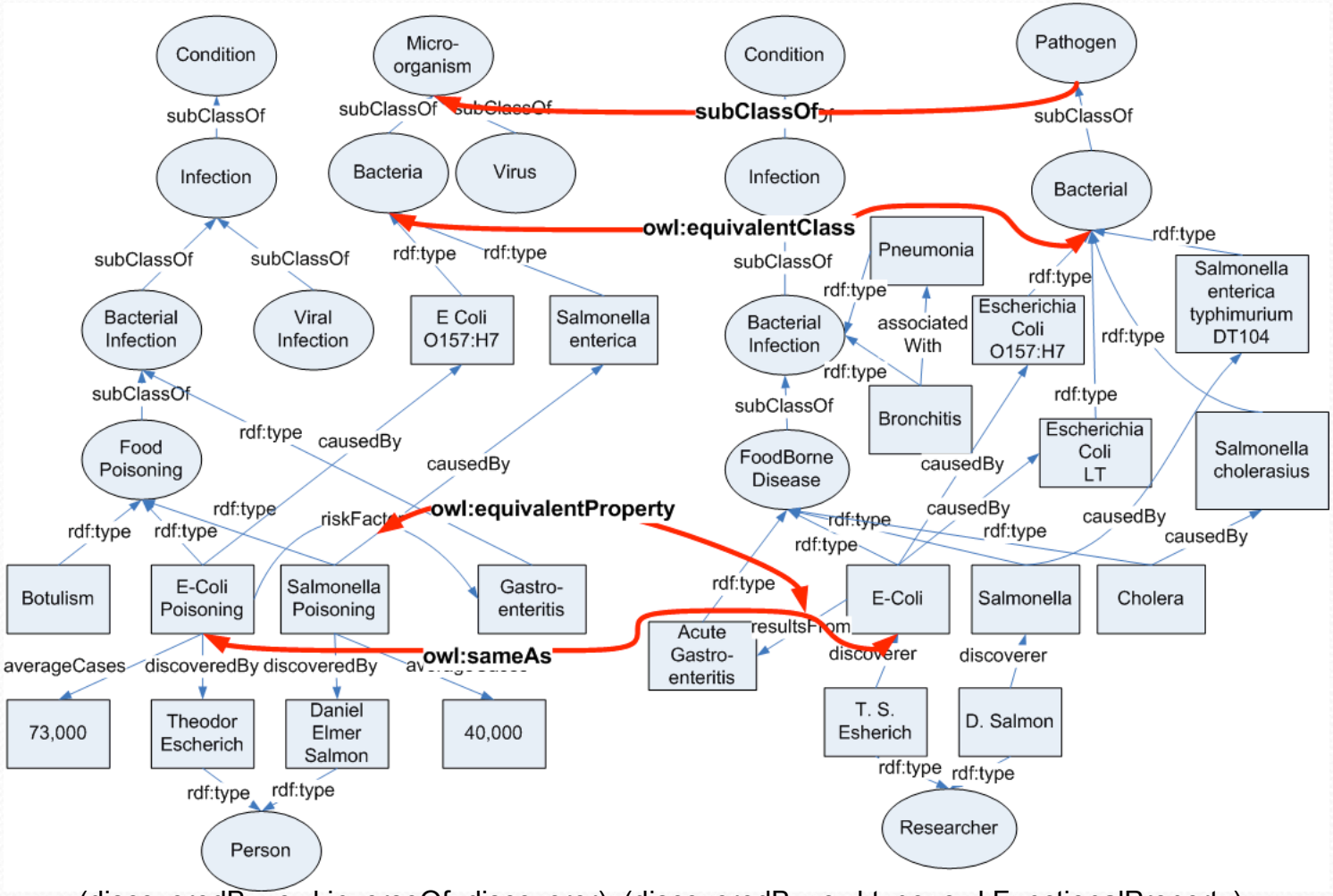
(discoveredBy, owl:inverseOf, discoverer); (discoveredBy, owl:type, owl:FunctionalProperty)
(discoveredBy, owl:inverseOf, discoverer); (associatedWith, owl:type, owl:TransitiveProperty)
(resultsF rom, rdfs:subPropertyOf, associatedWith)

Example alignments



(discoveredBy, owl:inverseOf, discoverer); (discoveredBy, owl:type, owl:FunctionalProperty)
(discoveredBy, owl:inverseOf, discoverer); (associatedWith, owl:type, owl:TransitiveProperty)
(resultsF rom, rdfs:subPropertyOf, associatedWith)

Example alignments



(discoveredBy, owl:inverseOf, discoverer); (discoveredBy, owl:type, owl:FunctionalProperty)
(discoveredBy, owl:inverseOf, discoverer); (associatedWith, owl:type, owl:TransitiveProperty)
(resultsF rom, rdfs:subPropertyOf, associatedWith)

Inference in OWL (Lite)

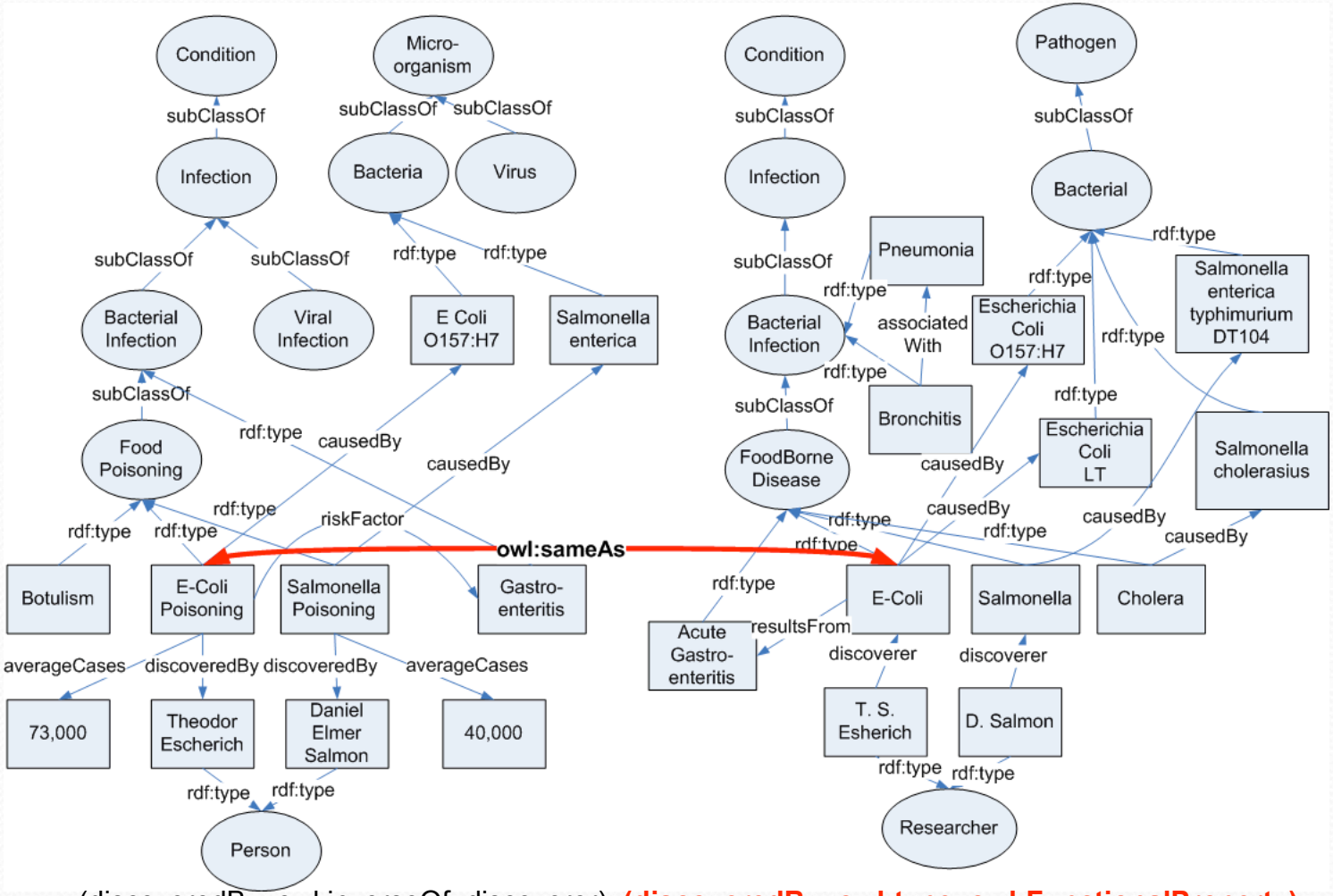
- A tableau-based method
- Example tableau rule:

$$\frac{(p\ \text{owl:inverseOf}\ p')\quad (o_1\ p\ o_2)}{(o_2\ p'\ o_1)}$$

- Example inconsistency:

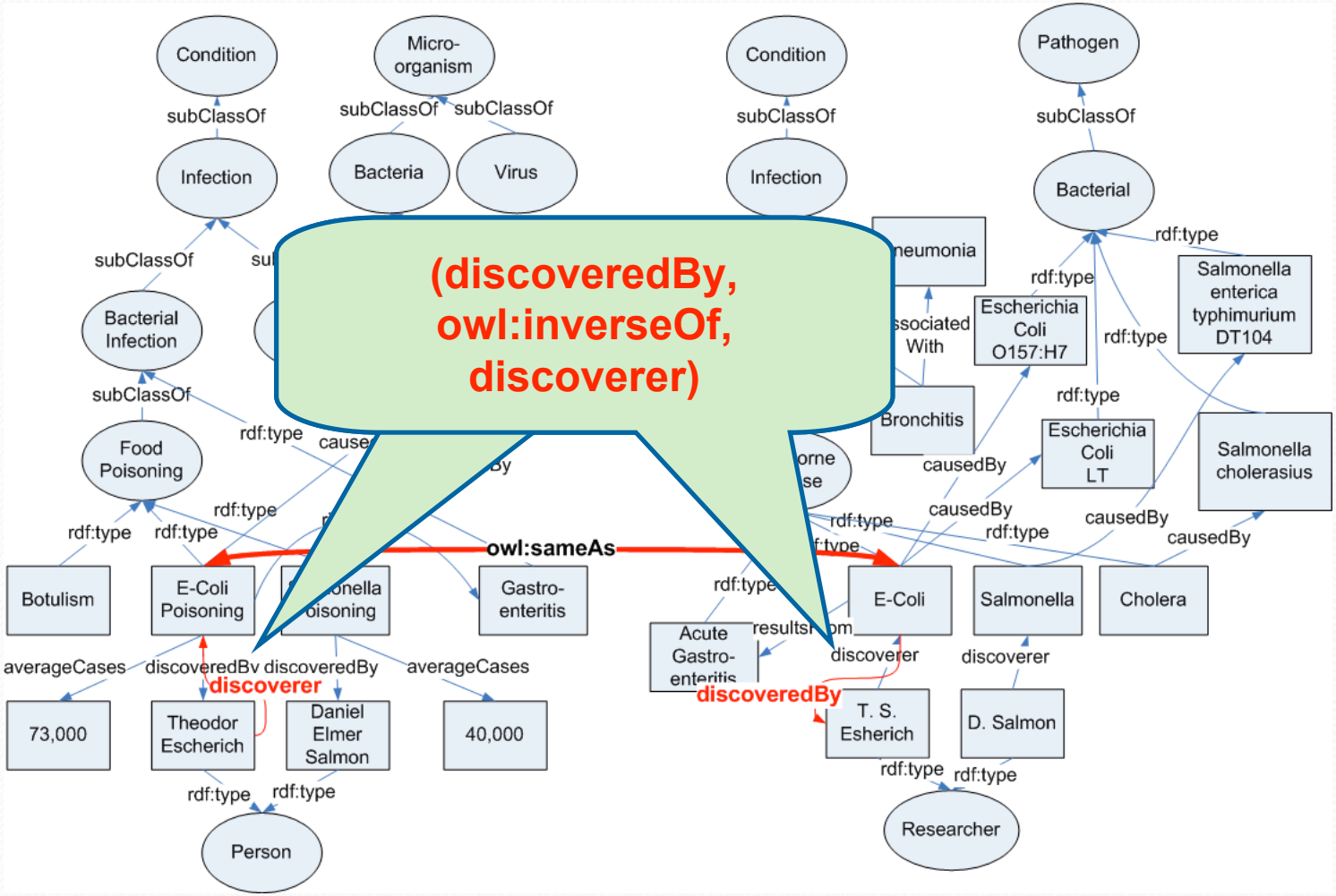
$$\frac{(o_1\ \text{owl:sameAs}\ o_2)\ (o_2\ \text{owl:differentFrom}\ o_1)}{\perp}$$

Example inference

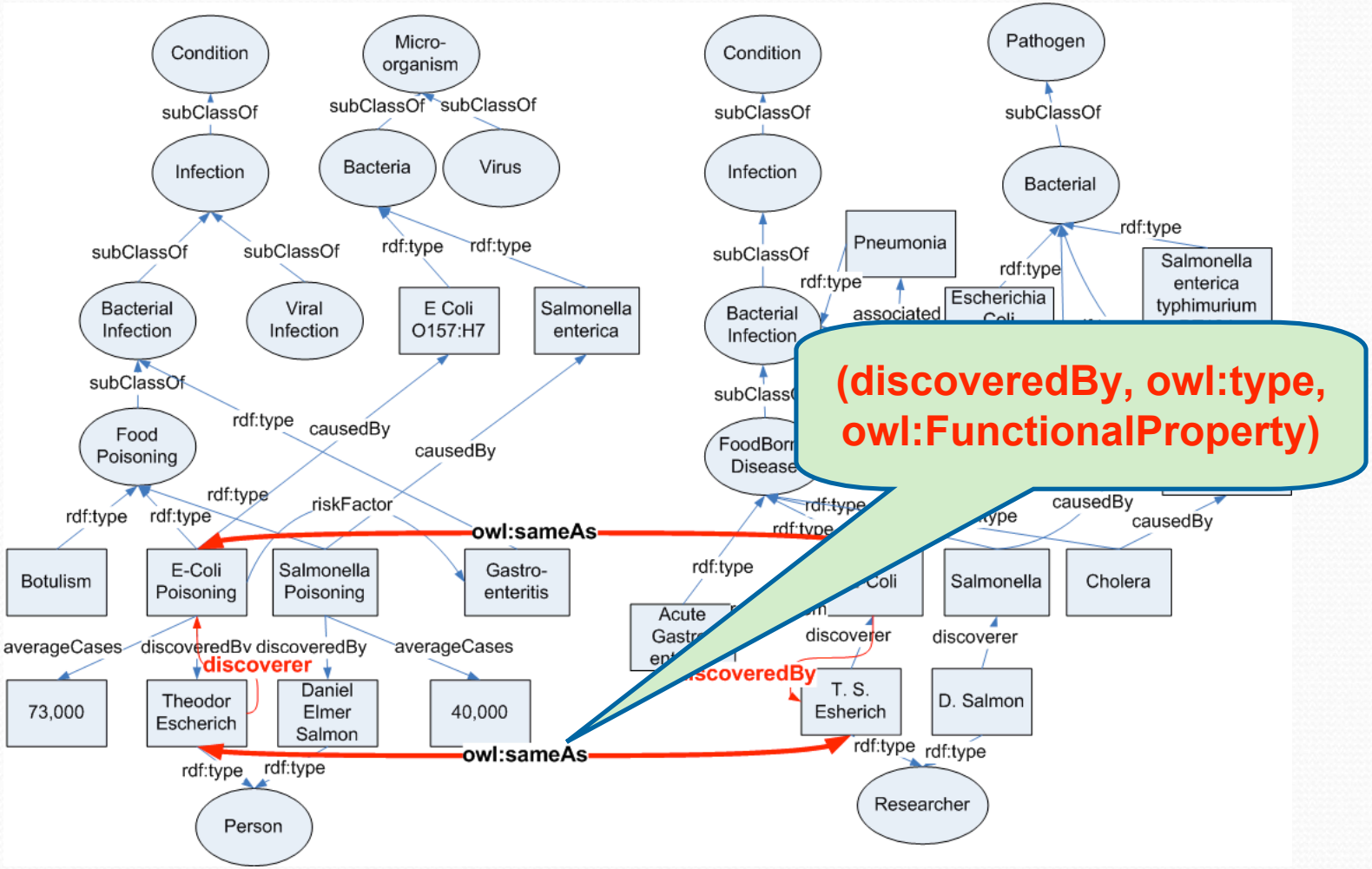


(discoveredBy, owl:inverseOf, discoverer); **(discoveredBy, owl:type, owl:FunctionalProperty)** **(discoveredBy, owl:inverseOf, discoverer)**; (associatedWith, owl:type, owl:TransitiveProperty) (resultsF rom, rdfs:subPropertyOf, associatedWith)

Example inference


(discoveredBy, owl:inverseOf, discoverer)
Condition
Micro-organism
Condition
Pathogen
subClassOf
subClassOf
subClassOf
subClassOf
subClassOf
Infection
Bacteria
Virus
Infection
Bacterial
subClassOf
Bacterial Infection
subClassOf
Food Poisoning
rdf:type
rdf:type
rdf:type

Example inference



The alignment problem

- Find a set of triples *(entity$_1$ relation entity$_2$)* where:
  - *entity$_1$, entity$_2$* are entities from the two ontologies
  - *relation* is one of
    - subClassOf, equivalentClass, subPropertyOf, equivalentProperty, sameAs
- For **integration**, the union of the ontologies and the alignment must be **consistent**.

Talk Overview

- Background - brief (selective) history
- Bringing data and metadata alignment together
- **The ILIADS method**
- Experimental evaluation

ILIADS - Ontology Alignment

- Produce Ontology alignments by
  - using instance similarity measures and
  - using logical inference (e.g., in OWL) to estimate quality of alignment

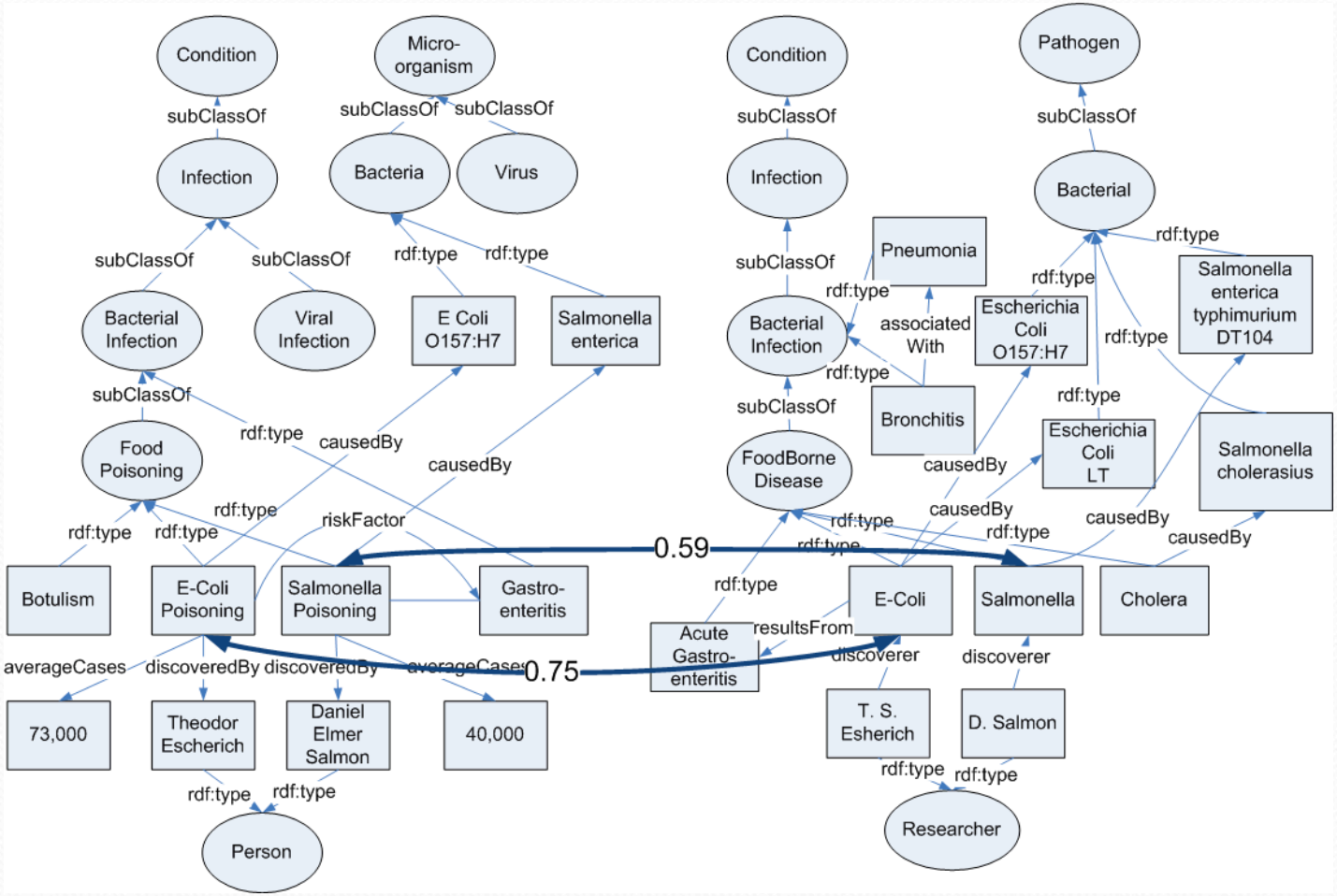
- Parameterize the method such that
  - it can be adapted for a wide variety of inputs
  - the parameters can be adjusted with minimal effort based on the input ontologies

SIGMOD 2007 - Octavian Udrea, Lise Getoor, Renée J. Miller

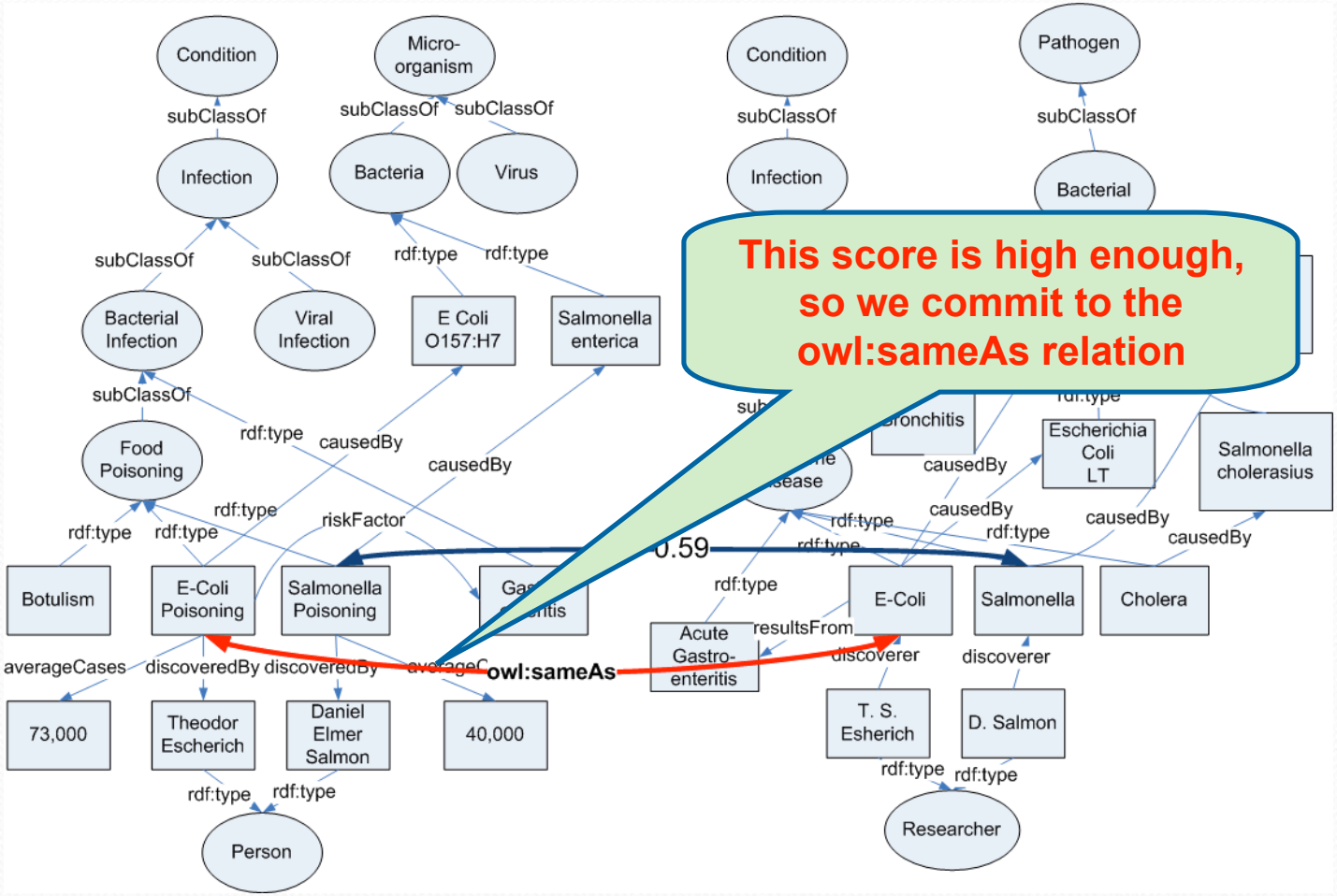
Aligning Ontologies

- Ideally, alignment should be treated as an optimization problem
  - Choose candidate pairs to maximize an ontology-level similarity measure
  - Unfeasible in practice (open question)
- To approximate, existing tools use locally computed similarity measures
  - Often, this means the “big picture” of the search space is ignored

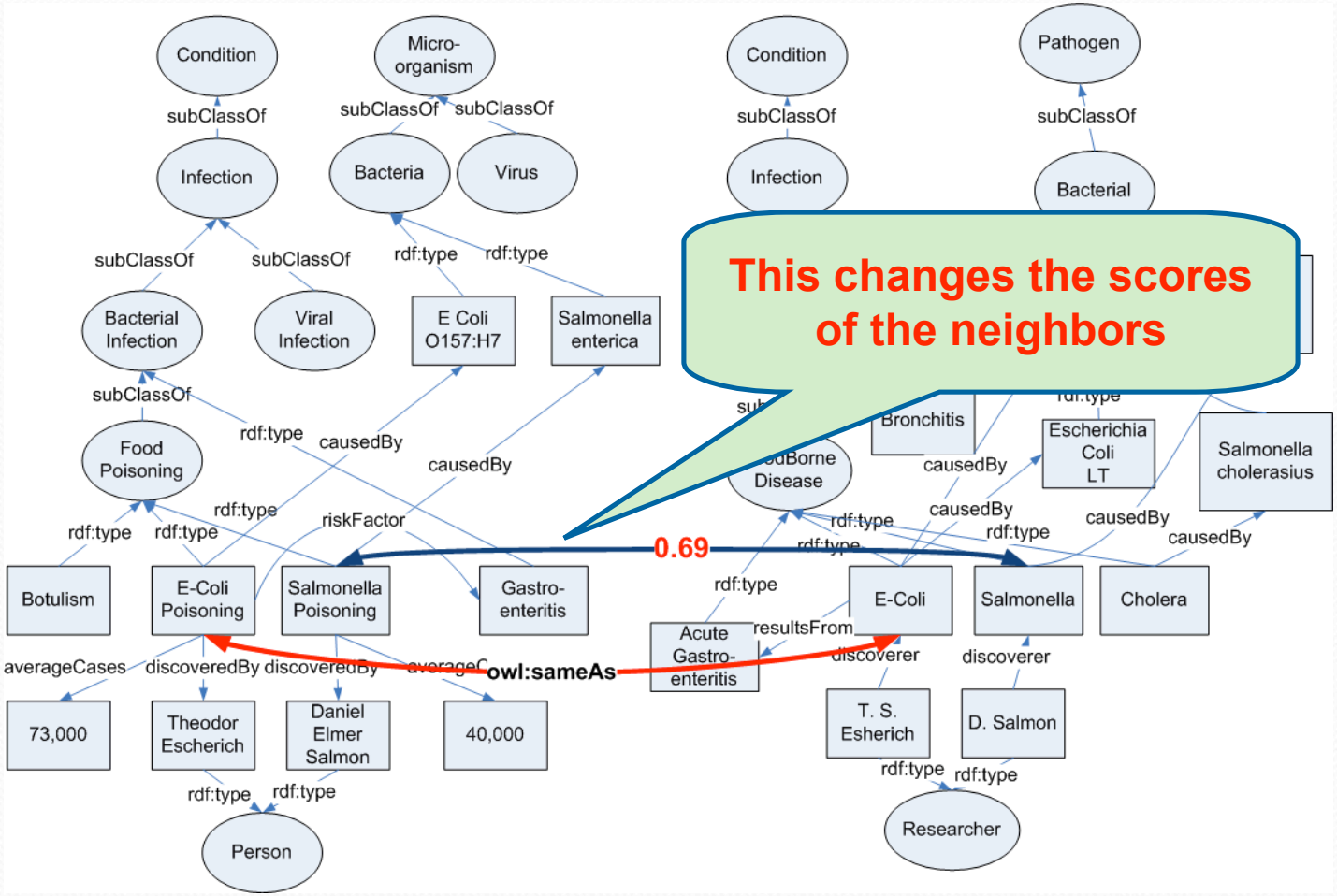
Incremental methods



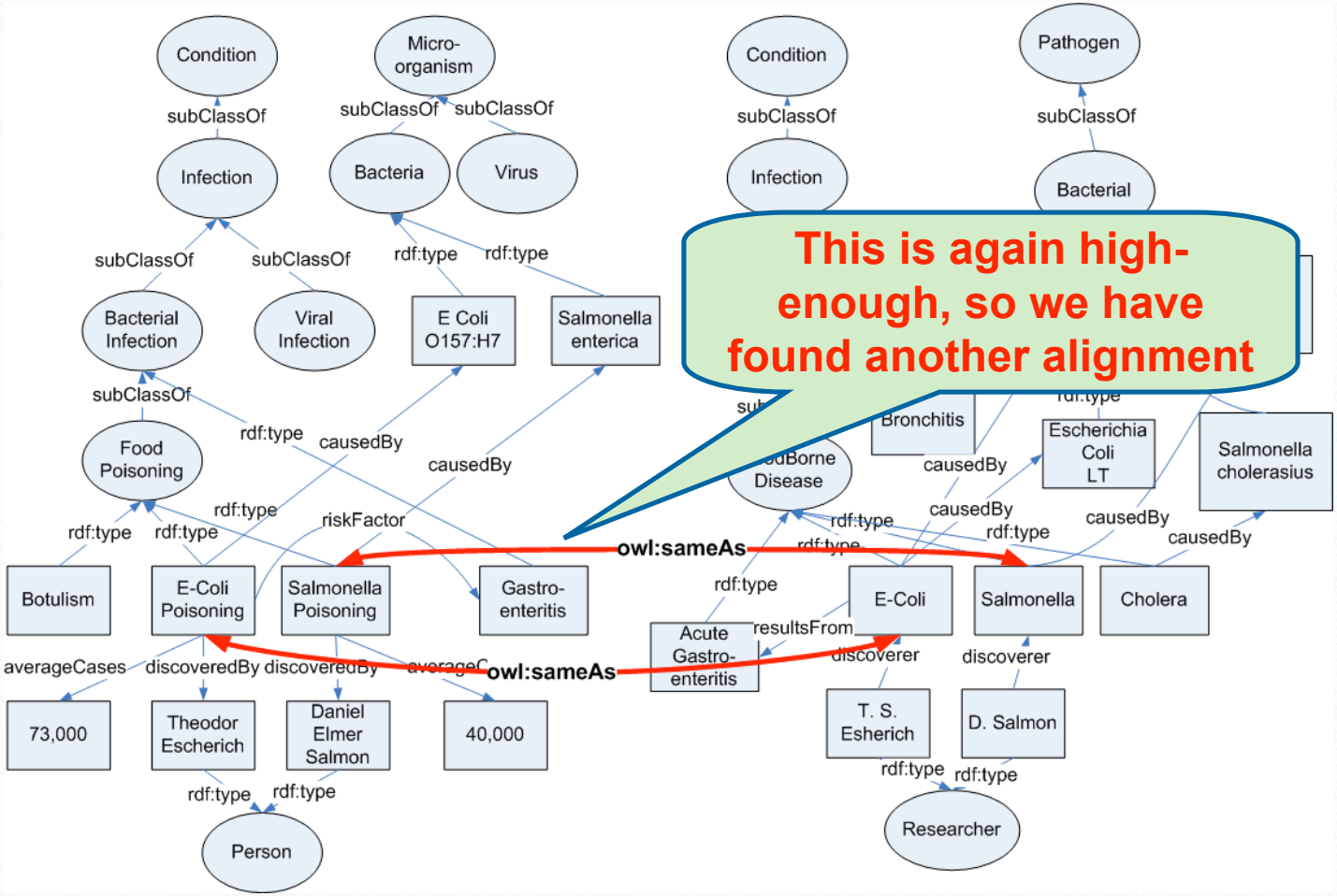
Incremental methods



Incremental methods



Incremental methods



The core of ILIADS

- Compute alignment candidates based on well established methods
  - Lexical, structural, extensional similarity
- In addition, evaluate how “good” a candidate pair is based on the logical consequences of asserting the alignment
  - We call this “inference similarity”
  - Essentially a look-ahead that estimates the impact of the alignment on the global similarity score

The ILIADS algorithm

repeat until no more candidates

1. Compute local similarities
2. Select *promising* candidates
3. **For each** candidate
   a. Perform N inference steps
   b. Update score with the inference similarity
4. Select the candidate with the best score

end

Computing similarity

repeat until no more candidates

1. **Compute local similarities**
2. **Select *promising* candidates**
3. **For each** candidate
   a. Perform N inference steps
   b. Update score with the inference similarity
4. Select the candidate with the best score

end

- $sim(e,e') = \lambda_x \, sim_{lexical}(e,e') + \lambda_s \, sim_{structural}(e,e') + \lambda_e \, sim_{extensional}(e,e')$
- Lexical similarity: Jaro-Winkler and Wordnet
- Structural similarity: Jaccard for various neighborhoods
- Extensional similarity: Jaccard on extensions
- Select candidates with sim(e,e') above a threshold

Performing inference

repeat until no more candidates

1. Compute local similarities
2. Select *promising* candidates
3. **For each** candidate
   a. **Perform N inference steps**
   b. Update score with the inference similarity
4. Select the candidate with the best score

end

For the candidate pair (e,e'):

- Select an axiom and apply the corresponding rule
- The *logical consequences* are the pairs of entities ($e^{(i)}$, $e^{(j)}$) that have just become equivalent
- Repeat a small number of times (5)

Updated score

repeat until no more candidates

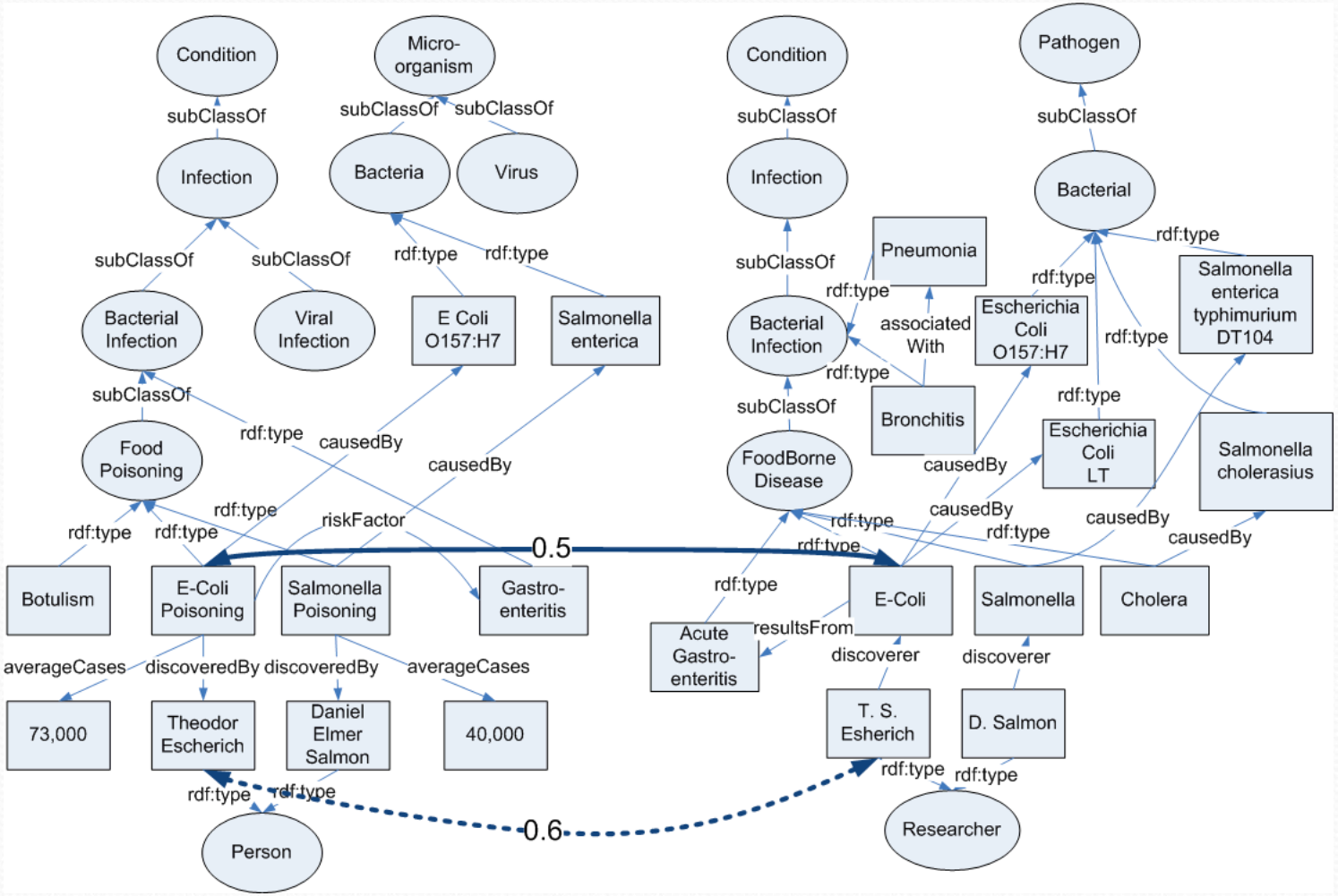
1. Compute local similarities
2. Select *promising* candidates
3. **For each** candidate
   a. Perform N inference steps
   b. **Update score with the inference similarity**
4. Select the candidate with the best score

end

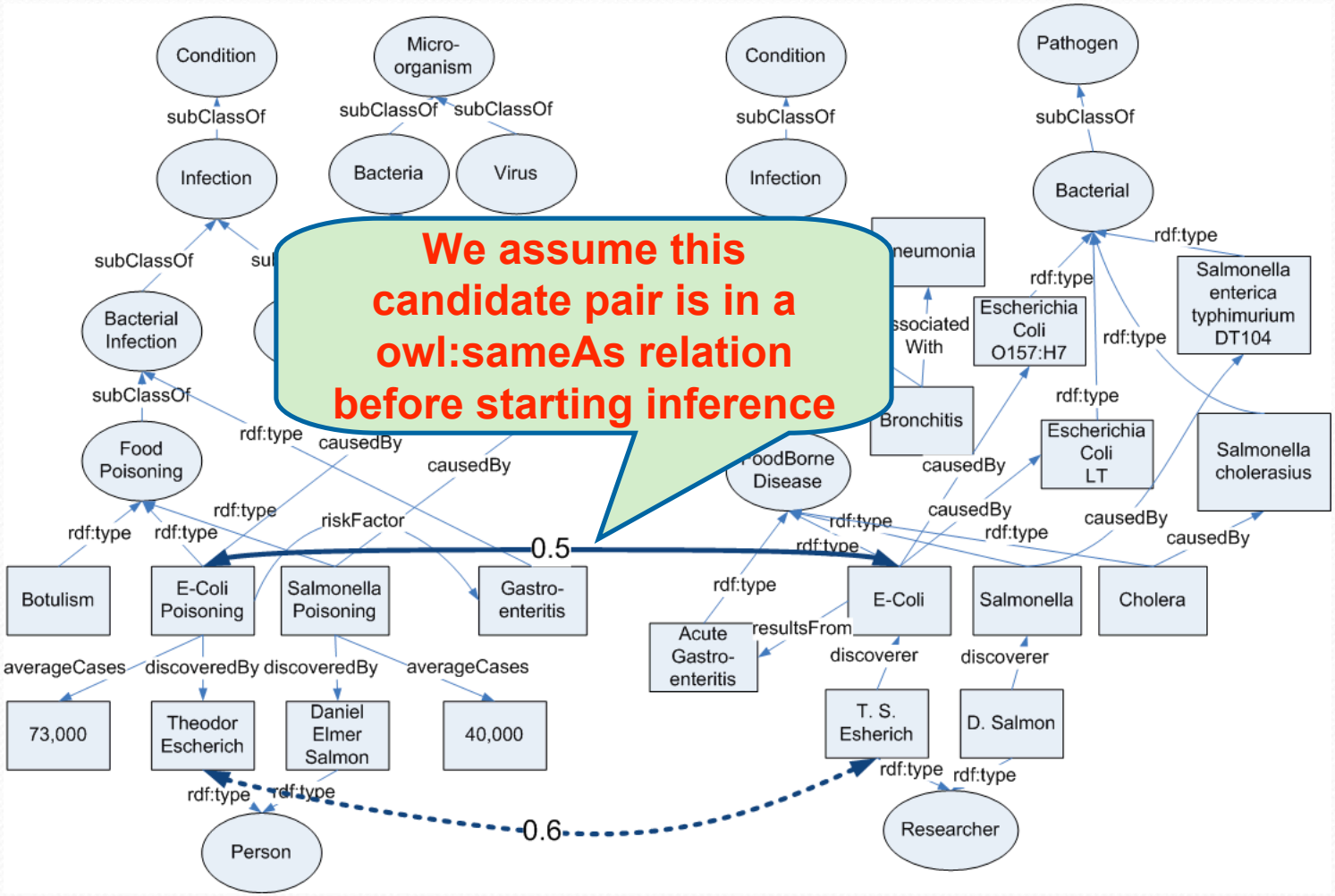
For the candidate pair (e,e'):

- Compute the product P of $sim(e^{(i)}, e^{(j)}) / (1 – sim(e^{(i)}, e^{(j)}))$ over all logical consequences
- $sim_{updated}(e,e') = sim(e,e') * P$

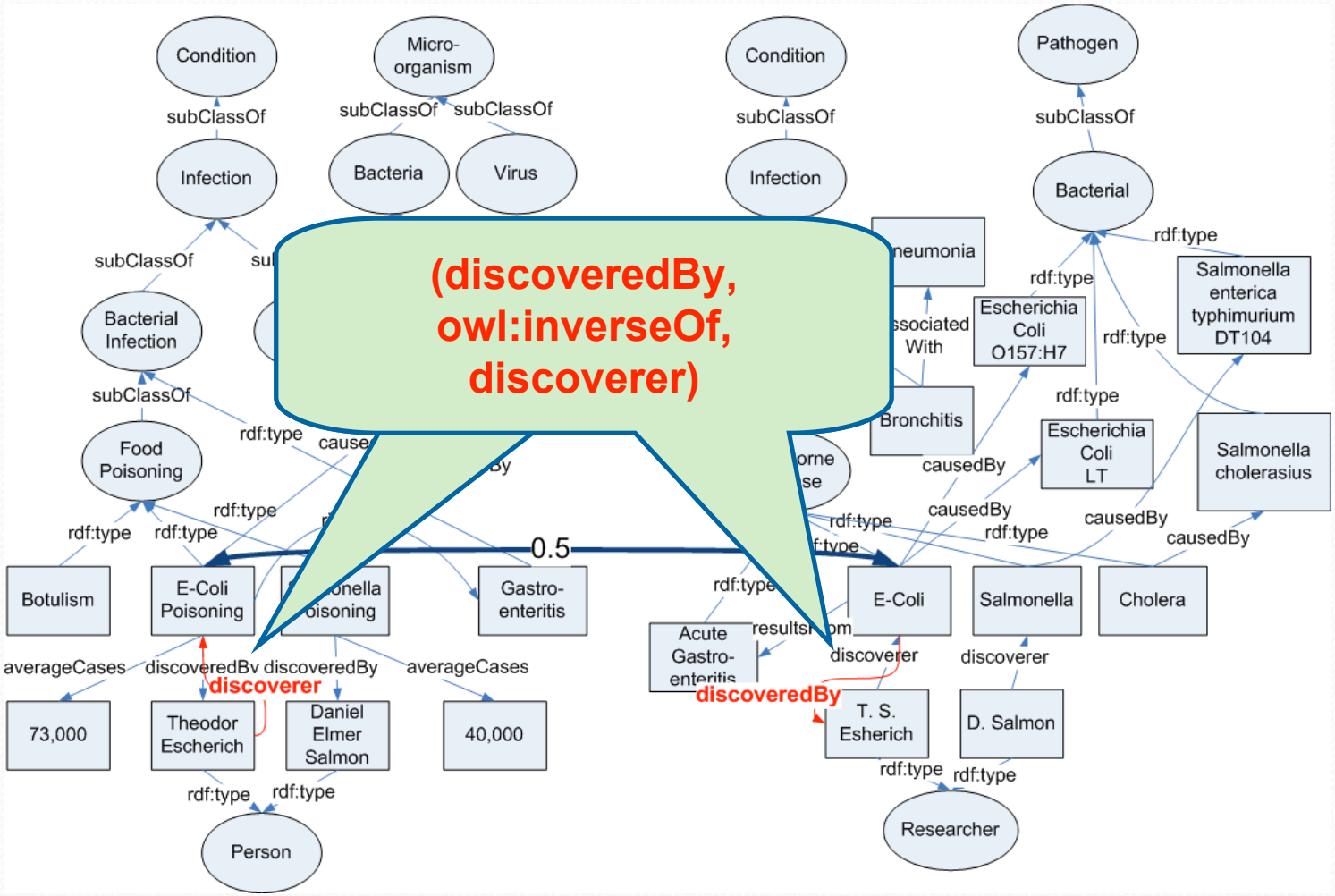
Example inference similarity



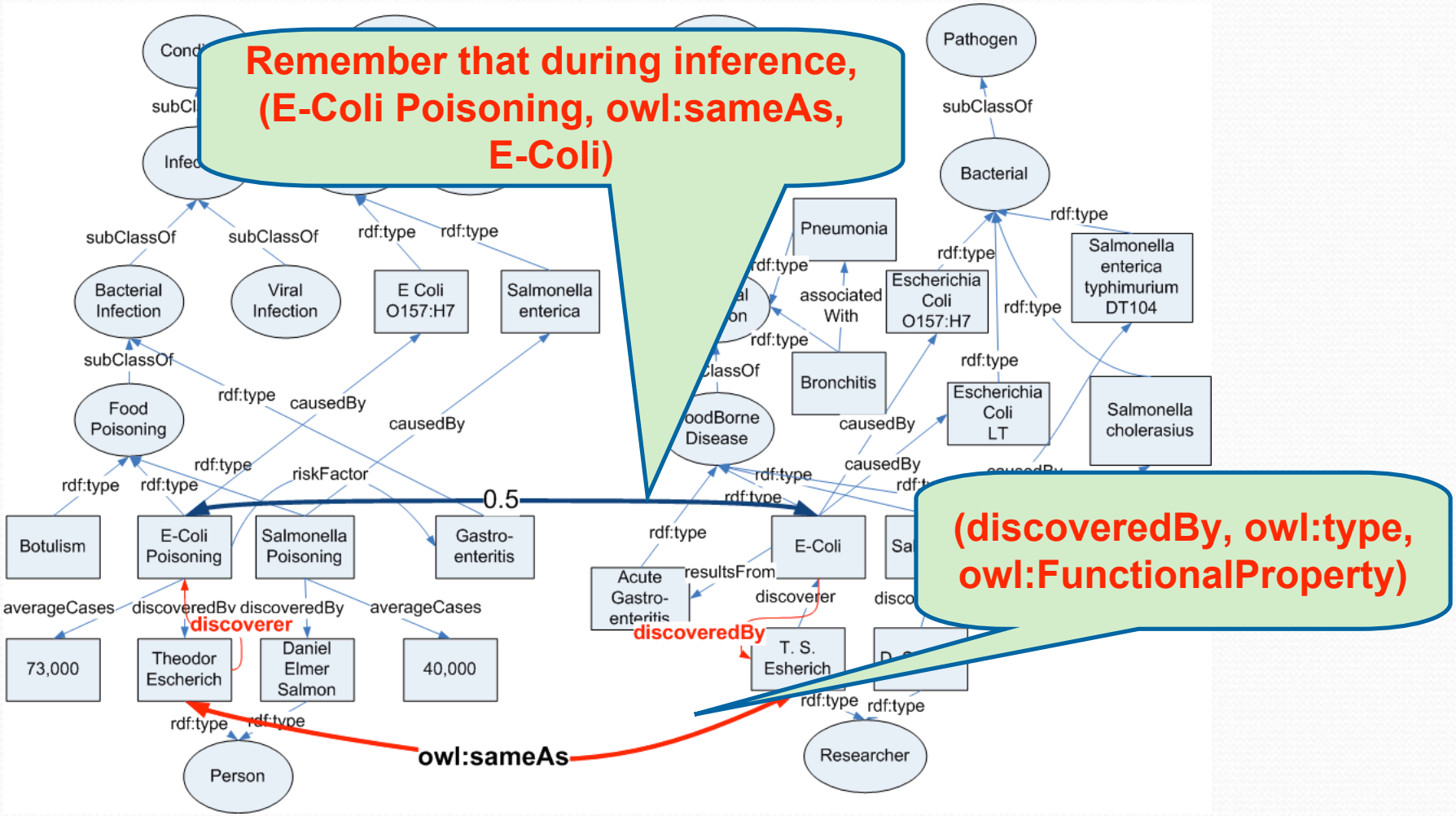
Example inference similarity



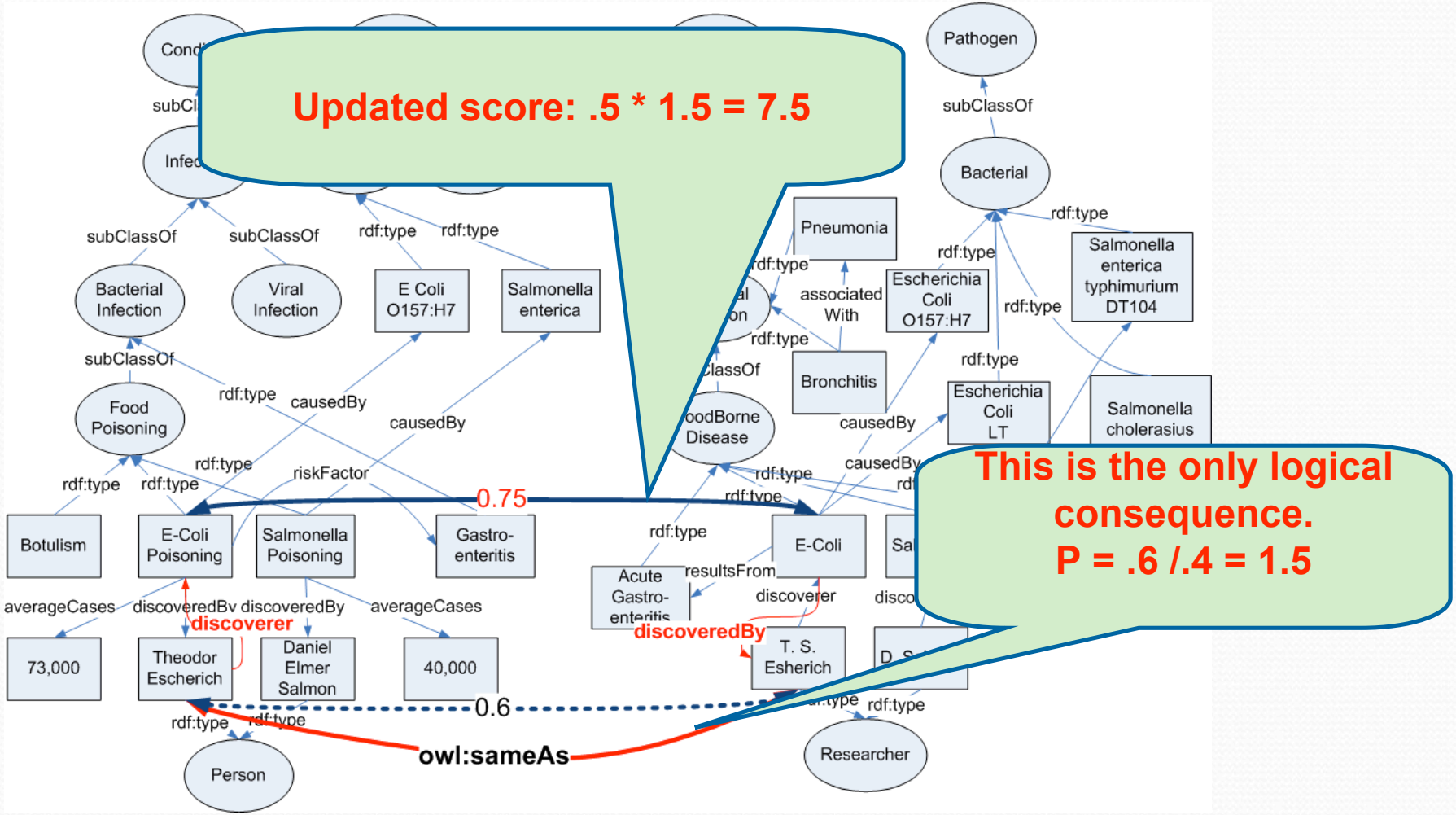
Example inference similarity



Example inference similarity



Example inference similarity



The ILIADS algorithm

- It is still a **local method**
  - Ultimately, it selects the best alignment after each step
- But it estimates the global impact of each alignment better
  - The inference similarity is a look-ahead measure of how good the candidate alignment is

Other issues

- ILIADS may not produce a consistent result
  - Inconsistent ontologies in less than .5% of runs
  - Pellet used to check consistency after ILIADS
- How do we decide between subsumption and equivalence for a pair of entities?
- How do we select the promising candidates?
- How do we choose the axioms to apply in the five inference steps?

Subsumption vs. equivalence

- Deciding whether two entities should subsume each other or be equivalent is not clear-cut
- Simple extensional technique to distinguish between the two cases
  - E.g., measure whether the instances of class c are “almost” the same of those of class c’ => rdfs:equivalentClass
  - If they are a subset, then rdfs:subClassOf

Cluster type selection

- Existing tools use various strategies to generate candidates from classes, individuals or properties
- ILIADS supports:
  - Randomly select from the three types
  - Weighted random (more classes than individuals means classes will be selected more often)
  - Classes first / Individuals first
  - Alternate at each step

Axiom selection policies

- The number of inference steps is small
  - The axioms applied must make a difference
- ILIADS always selects from **relevant** axioms according to a policy:
  - Random
  - Property axioms first (e.g, owl:TransitiveProperty)
  - Class axioms first (e.g., rdfs:subClassOf)
  - Transitive/Inverse/Functional first (since they tend to "generated" sameAs relationships)

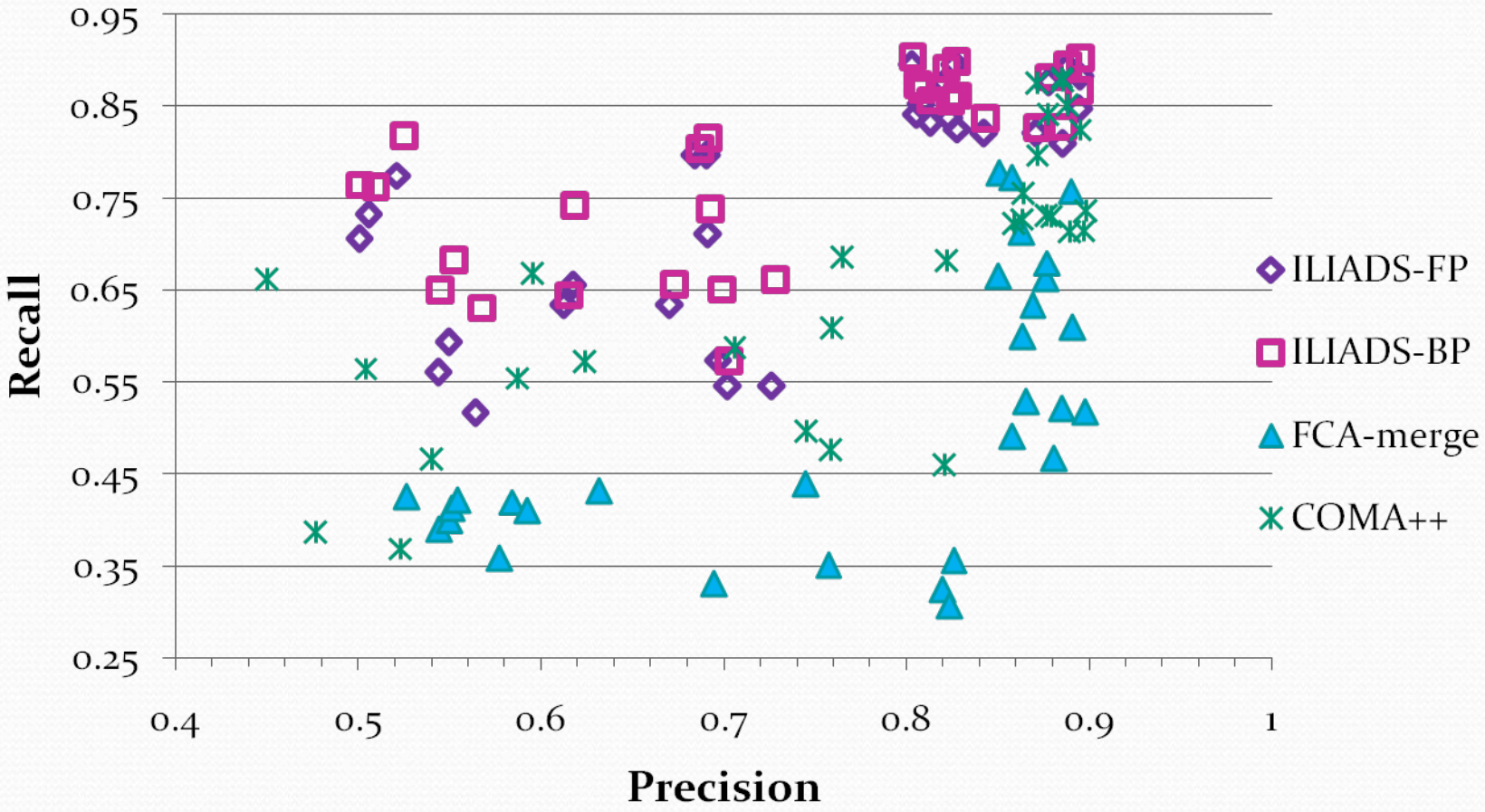
Talk Overview

- Background - brief (selective) history
- Bringing data and metadata alignment together
- The ILIADS method
- **Experimental evaluation**

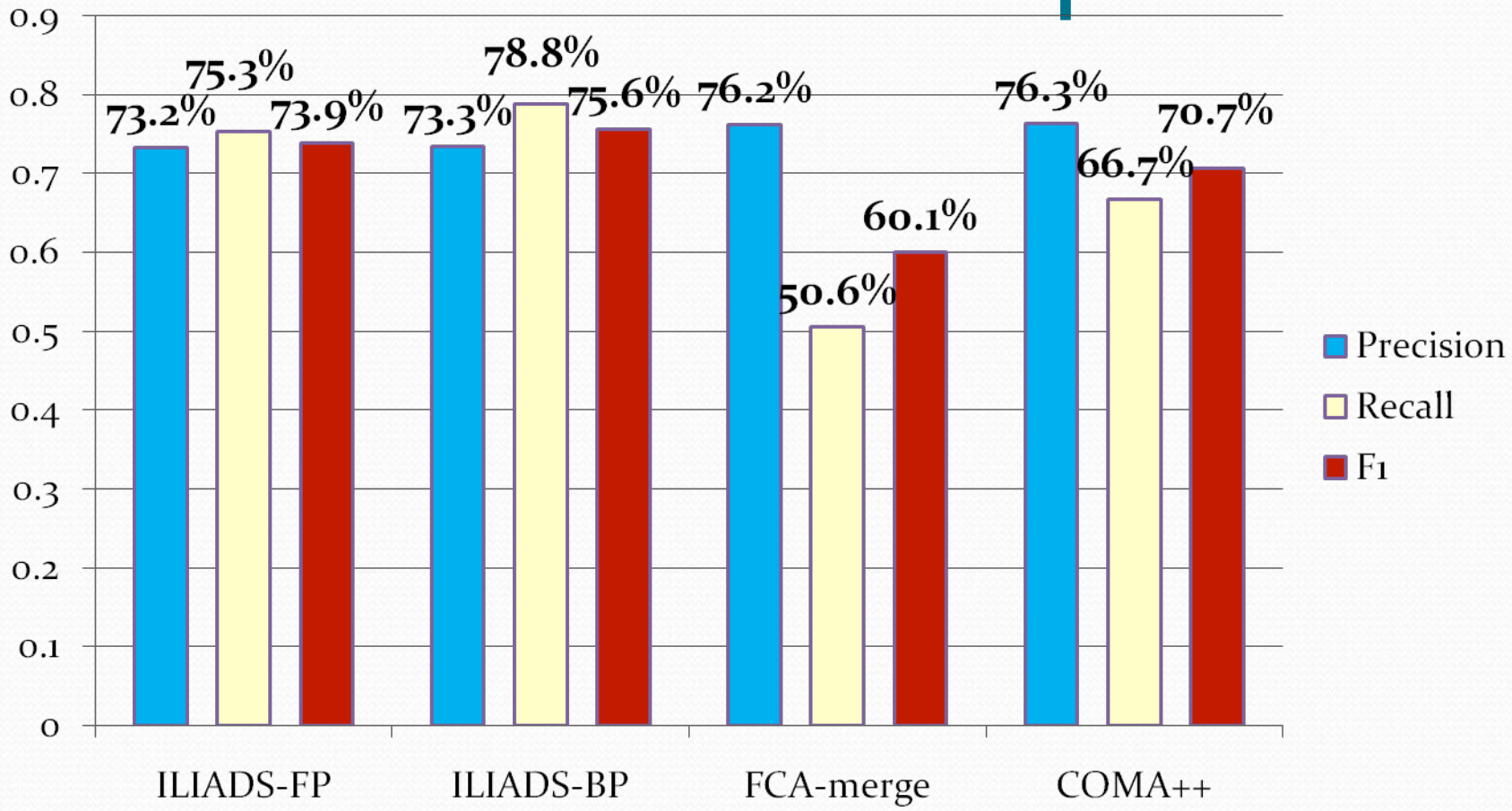
Experimental framework

- Pairs of ontologies (from U. Bremen portal)
  - Ontologies from 194 to over 20000 triples
- Ground truth provided by human reviewers
- Comparison in terms of recall and precision with
  FCA-merge [Stumme, Maedche IJCAI 01]
  COMA++ [Aumueller et al. SIGMOD 05]
- Two versions of the algorithm
  - Best overall average quality ILIADS – FP
  - Best parameters for each pair ILIADS – BP

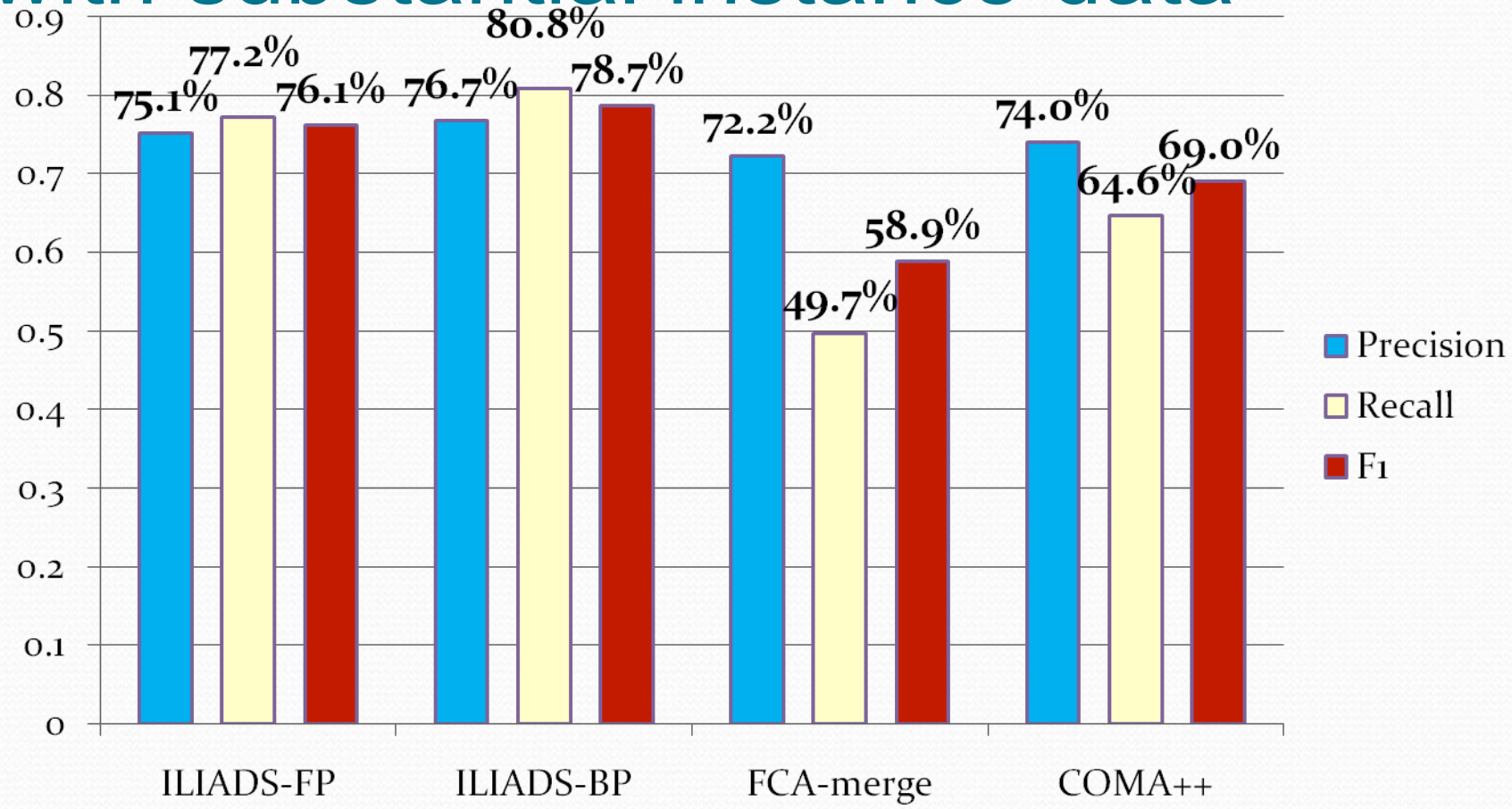
Precision/recall



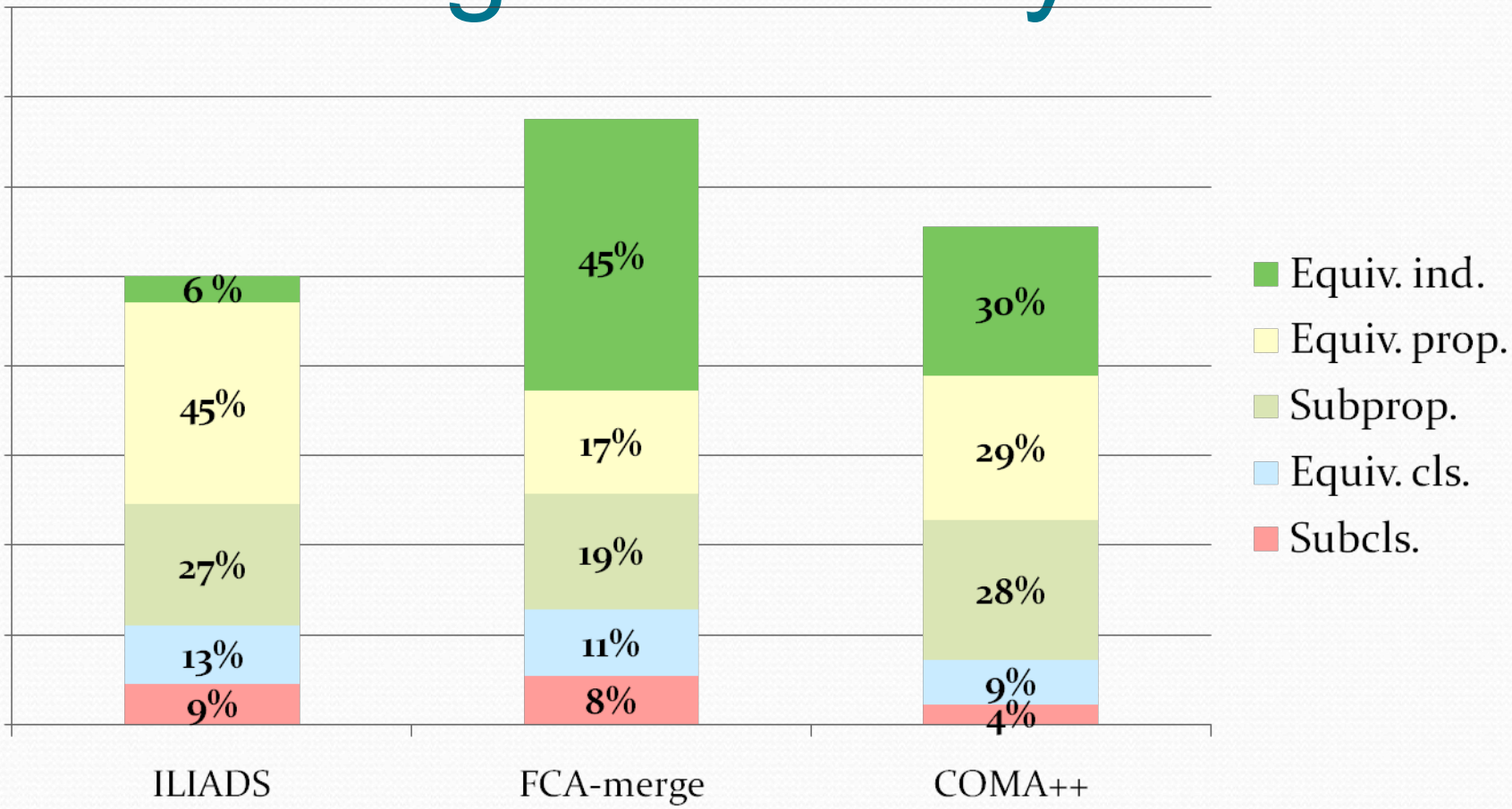
Precision/recall comparison



Precision/recall for ontologies with substantial instance data

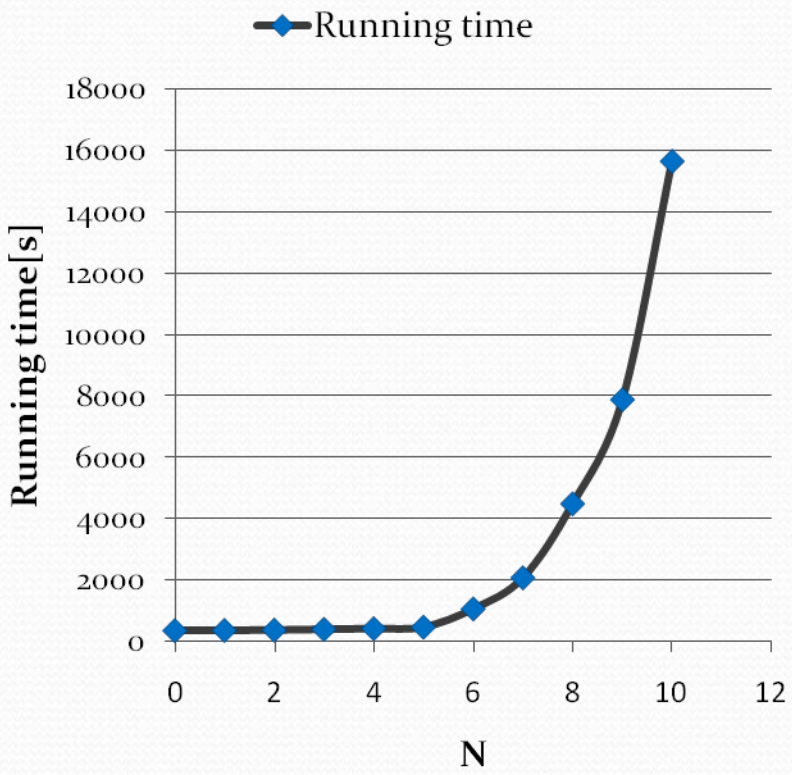


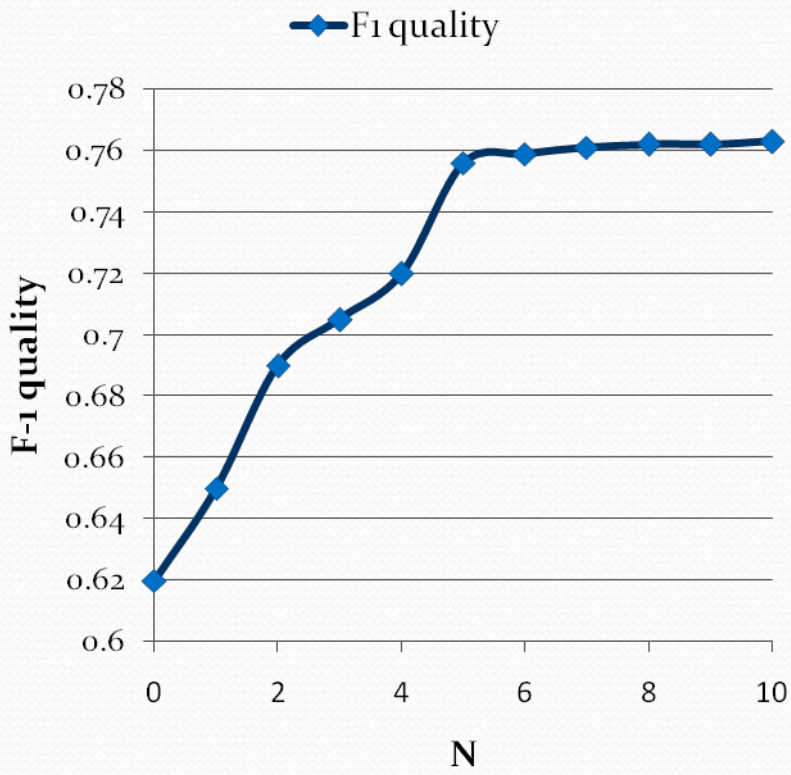
False negative analysis



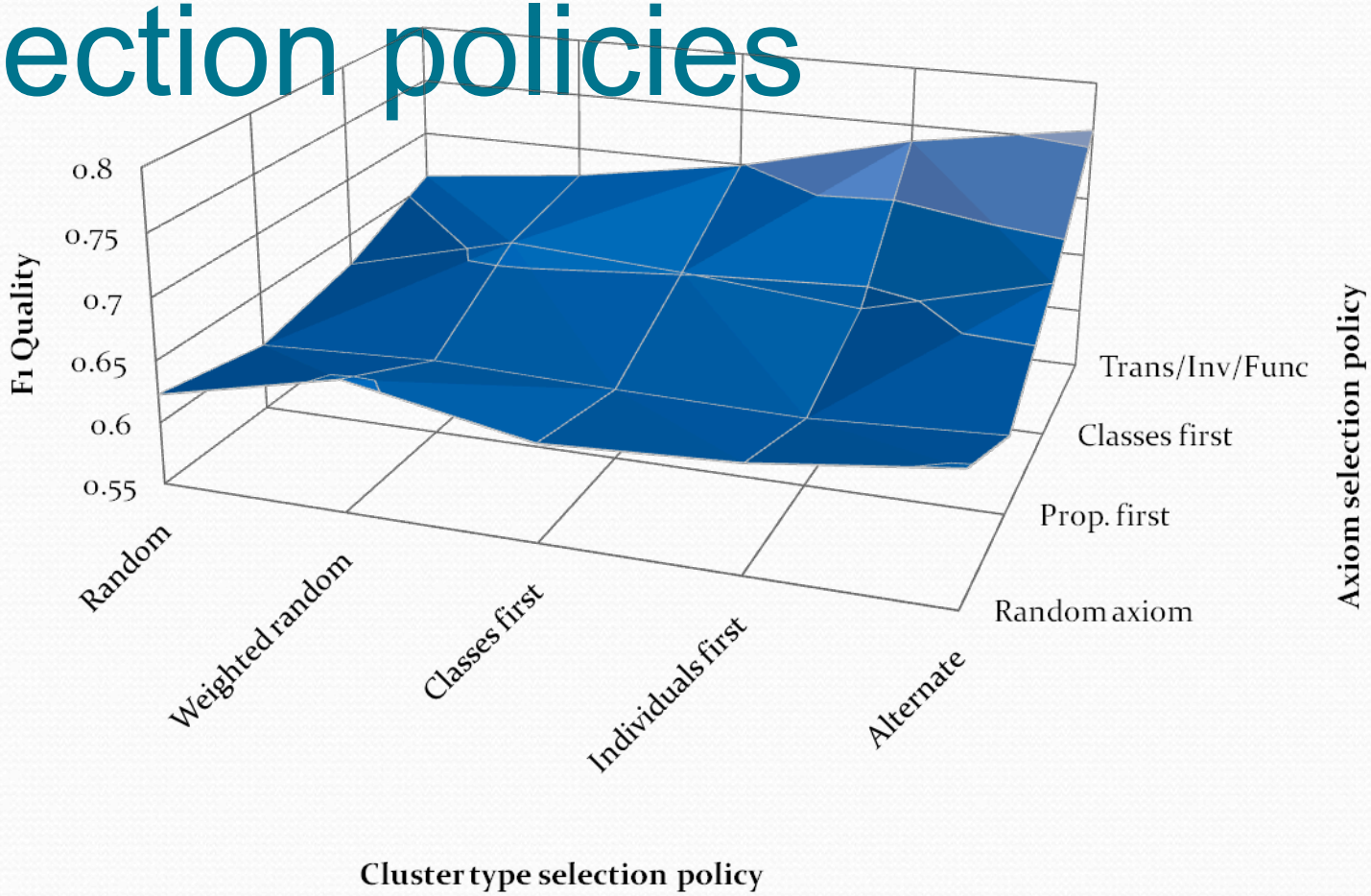
Number of inference steps

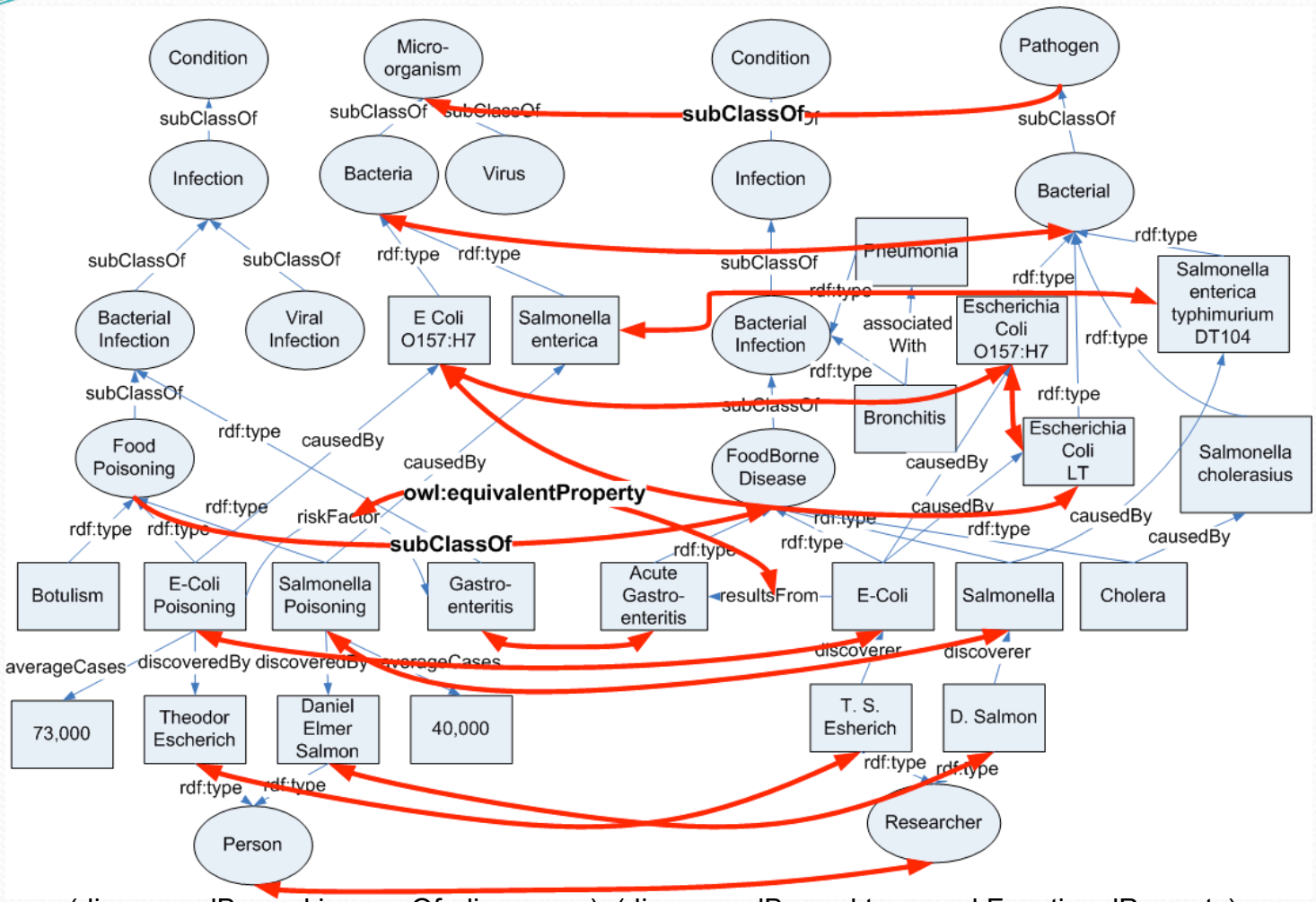
The number of 5 inference steps was chosen as the best compromise between:





Cluster type/axiom selection policies





(discoveredBy, owl:inverseOf, discoverer); (discoveredBy, owl:type, owl:FunctionalProperty)
(discoveredBy, owl:inverseOf, discoverer); (associatedWith, owl:type, owl:TransitiveProperty)
(resultsF rom, rdfs:subPropertyOf, associatedWith)

Choosing the parameters

- The structural similarity coefficients strongly correlate with the average degree of the node
- The structural coefficient for classes correlates with the number of rdfs:subClassOf relationships
- The extensional coefficients correlate with the ratio of instance to classes

Parameter sensitivity

- Structural coefficients are stable around the ILIADS-FP setting for 25 out of 30 pairs
  - The remaining 5 pairs have large differences between their average node degrees
- Extensional coefficients are stable around the ILIADS-FP setting for 21 pairs
  - The remaining 9 pairs have a low ratio of instances to classes (< 1.9)

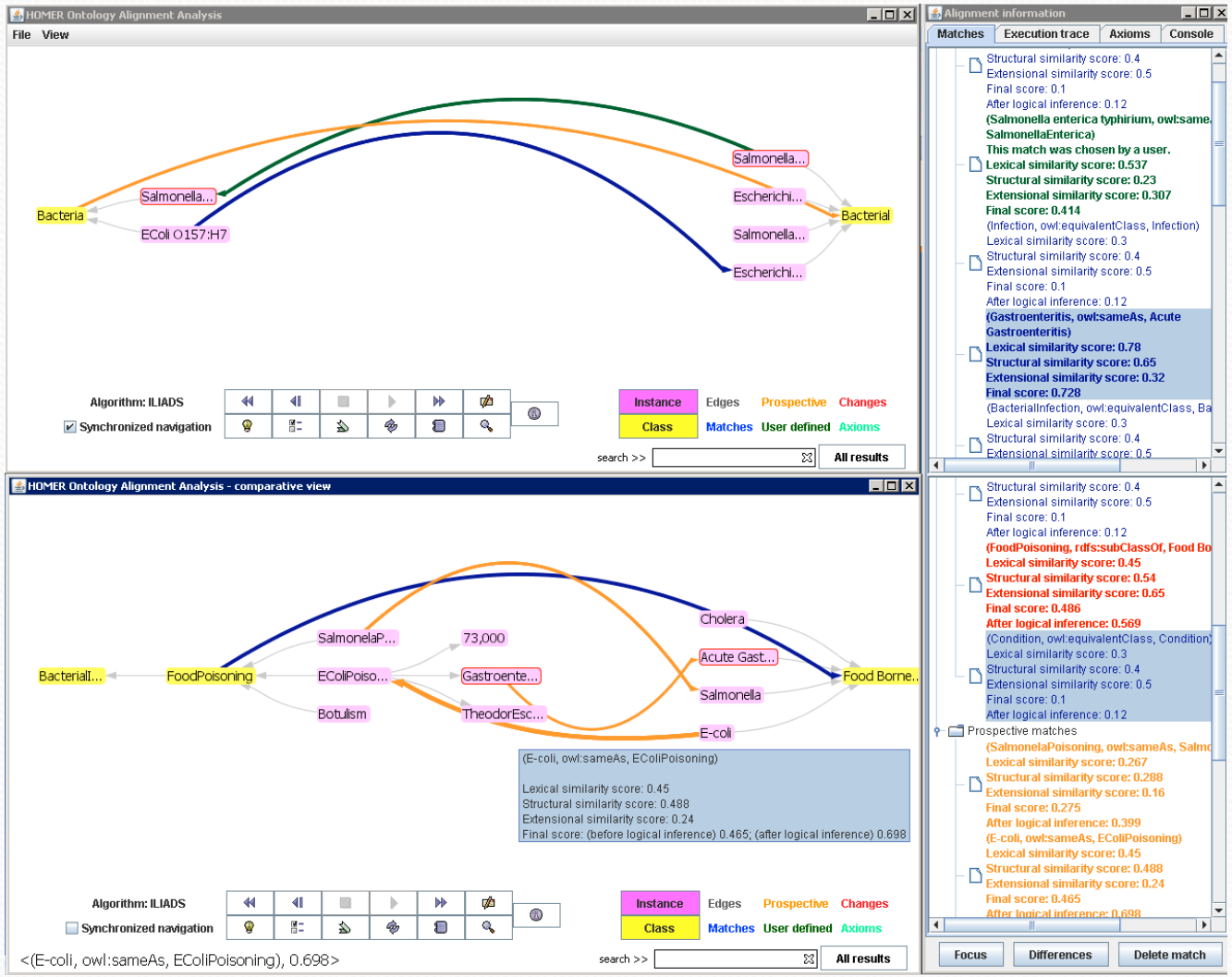
Experimental results summary

- ILIADS has better quality than COMA++ and FCA-merge, with a significant difference for all pairs with substantial instance data
- Matching properties is the major cause of false negatives for all three systems, but ILIADS does better at matching instances
- Structural and extensional coefficients correlate with structural properties and are stable for ontologies with similar structure

ILLIADS Summary

- New algorithm that tightly integrates statistical matching and logical inference to produce better quality alignments
- Found intriguing correlations between structure and matching strategies
- Improvement over existing systems
  - 25% higher quality than FCA-merge,
  - 11% higher recall than COMA++ at comparable precision

HOMER: Tool for Ontology Alignment



http://www.cs.umd.edu/projects/linqs/iliads

Conclusions and Areas for Future Research

Information Alignment: Summary

- The process of finding, modeling and using the correspondences or connections that place information artifacts in relation to each other

Need new, flexible, adaptive methods for information alignment which can take context into account and which can exploit both logical and probabilistic consequences

Open Issues

- New issues of scale in using alignments
- Query-time data and metadata alignment
- Notion of multiple alignments; no single one best
- Need to keep track and make use of lineage
- Need to understand which information is most informative and useful for alignment: data, structure, metadata, etc.
- Need for methods for evaluation and quality measures

Thank you

Renée J. Miller

www.cs.toronto.edu/~miller