Data & Metadata Alignment

Lise Getoor

University of Maryland, College Park

Tutorial

October 23, 2007

Some Acknowledgements

- Students:

Indrajit Bhattacharya	**Louis Licamele**	**Vivek Sehgal**
Mustafa Bilgic	**Qing Lu**	**Prithvi Sen**
Rezarta Islamaj	**Galileo Namata**	**Elena Zheleva**

- Collaborators:

Chris Diehl	**Hyunmo Kang**	**Lisa Singh**
Tina Eliasi-Rad	**Renee Miller**	**Ben Taskar**
John Grant	**Ben Shneiderman**	**Octavian Udrea**

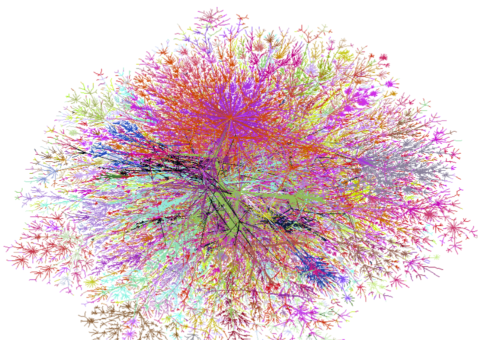
- Funding Sources:

KDD Program

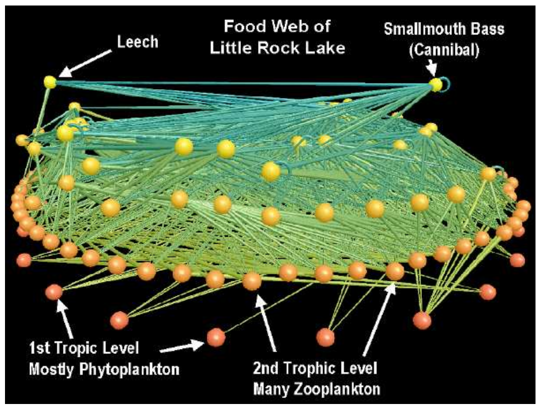
PART I: DATA ALIGNMENT

Graphs and Networks *everywhere*…

- The Web, social networks, communication networks, financial transaction networks, biological networks, etc.



Internet Map, Burch and Cheswick



Food Web, Martinez

Others available at Mark Newman's gallery:
http://www-personal.umich.edu/~mejn/networks/

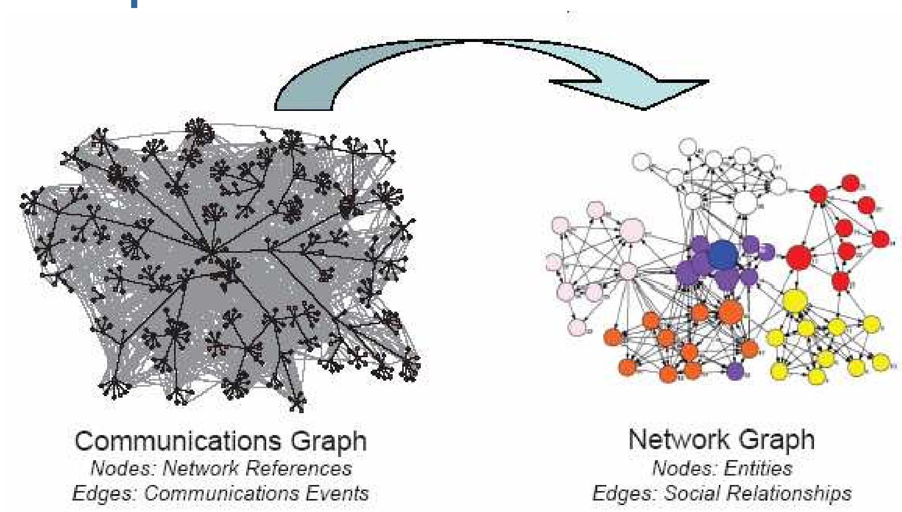
Wealth of Data

- Inundated with data describing networks
- But much of the data is
  - noisy and incomplete
  - at WRONG level of abstraction for analysis

Graph Identification

Graph Alignment

Graph Transformations

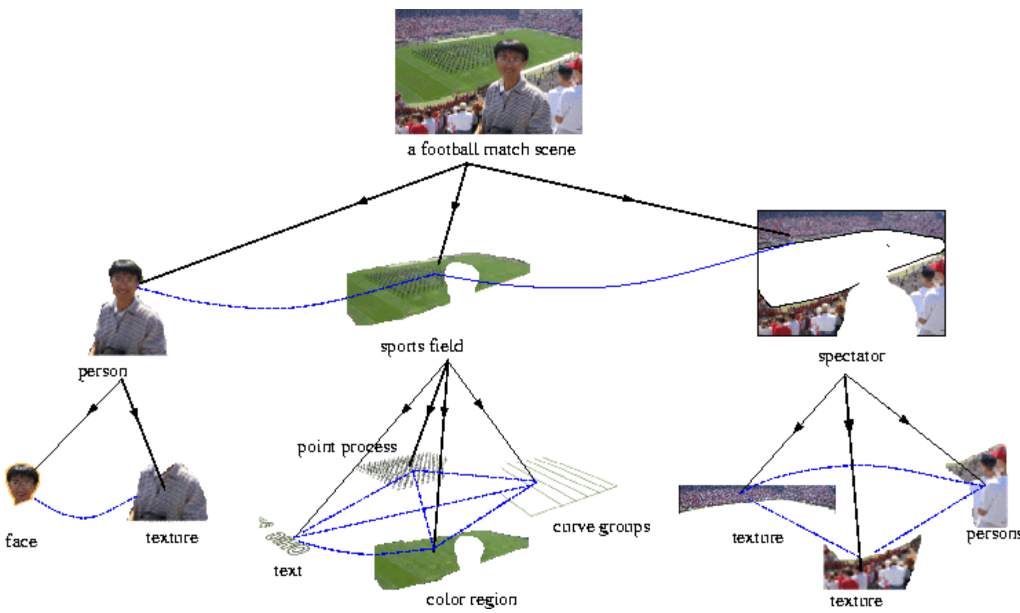


Data Graph $\Rightarrow$ Information Graph

1. Entity Resolution: mapping email addresses to people
2. Link Prediction: predicting social relationship based on communication
3. Collective Classification: labeling nodes in the constructed social network

HP Labs, Huberman & Adamic

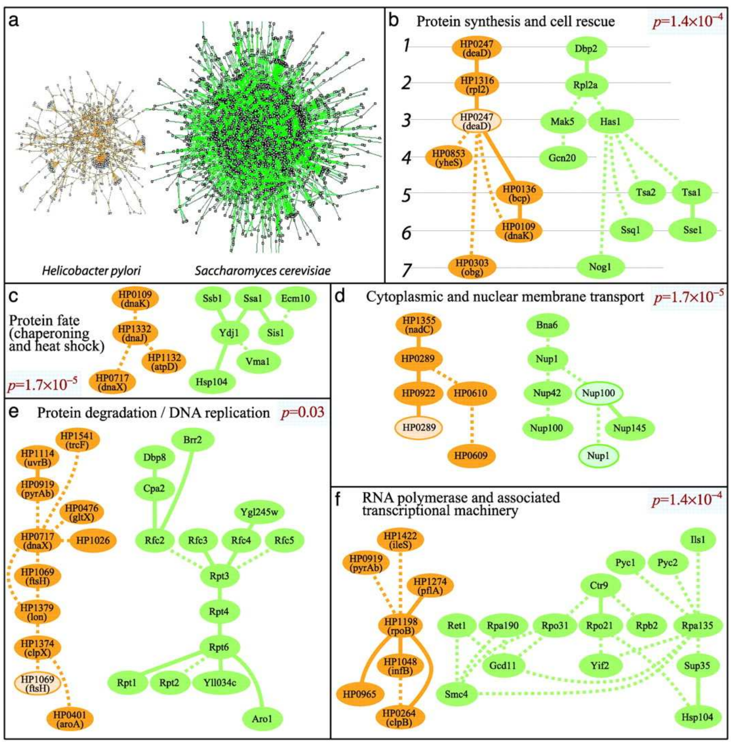
Vision: Image Parsing



Graph Partitioning + Graph Matching

Z.W. Tu, X.R. Chen, A.L. Yuille, and S.C. Zhu, IV05; Lin, Zhu and Wang, IV07

Bio: Protein Network Alignment



Kelley, Brian P. et al. PNAS03

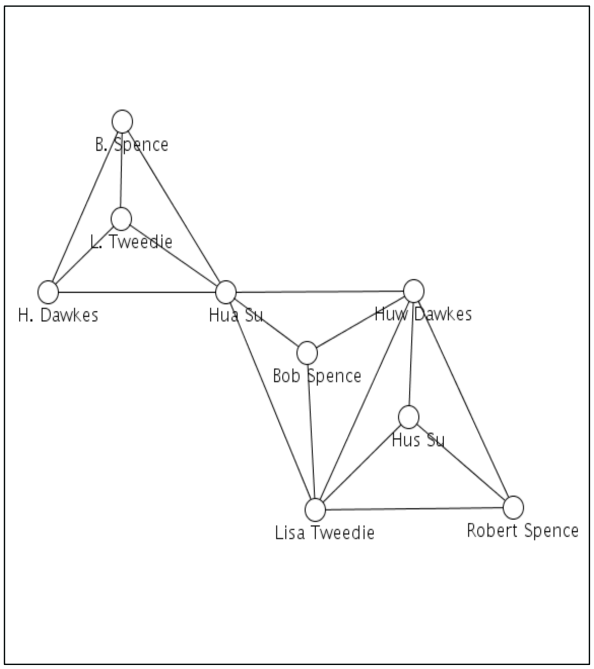
Roadmap

- The Problem
- **The Components**
  - Entity Resolution
  - Collective Classification
  - Link Prediction
  - Putting It All Together
- Open Questions

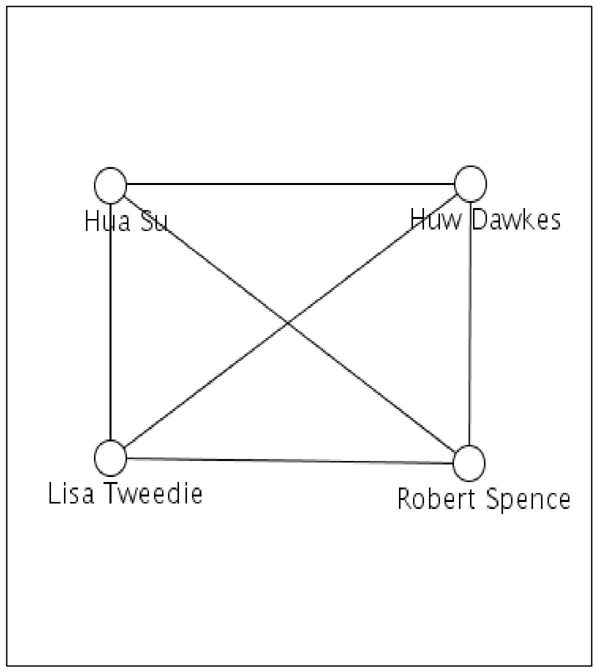
Entity Resolution

- **The Problem**
- Relational Entity Resolution
- Algorithms

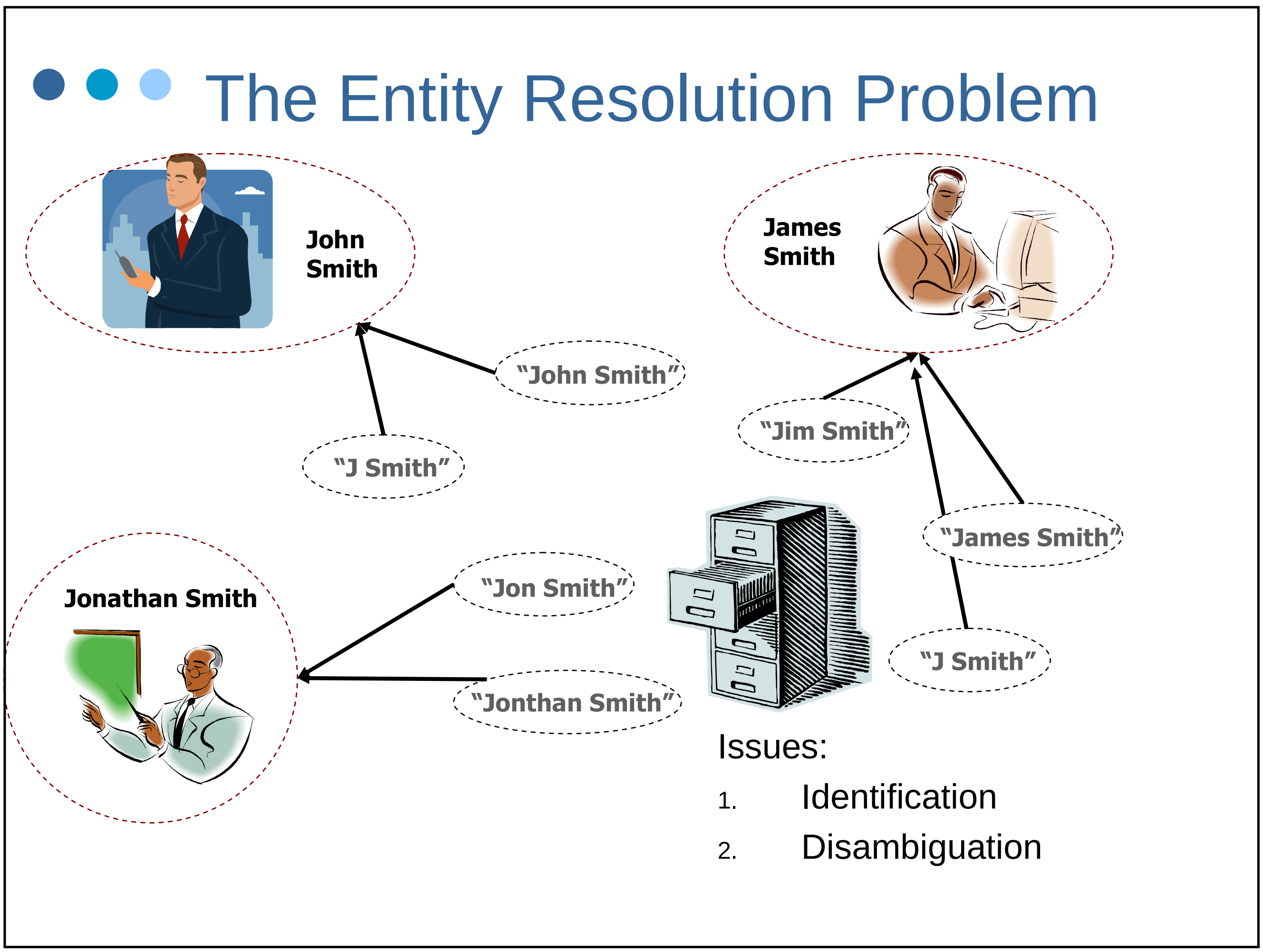
InfoVis Co-Author Network Fragment



before



after


The Entity Resolution Problem
John Smith
James Smith
"John Smith"
"J Smith"
"Jim Smith"
Jonathan Smith
"Jon Smith"
"James Smith"
"J Smith"
"Jonthan Smith"
Issues:
1. Identification
2. Disambiguation

Attribute-based Entity Resolution

Pair-wise classification

"J Smith"	"James Smith"	?
"Jim Smith"	"James Smith"	0.8
"J Smith"	"James Smith"	?
"John Smith"	"James Smith"	0.1
"Jon Smith"	"James Smith"	0.7
"Jonthan Smith"	"James Smith"	0.05

1. Choosing threshold: precision/recall tradeoff
2. Inability to disambiguate
3. Perform transitive closure?

Entity Resolution

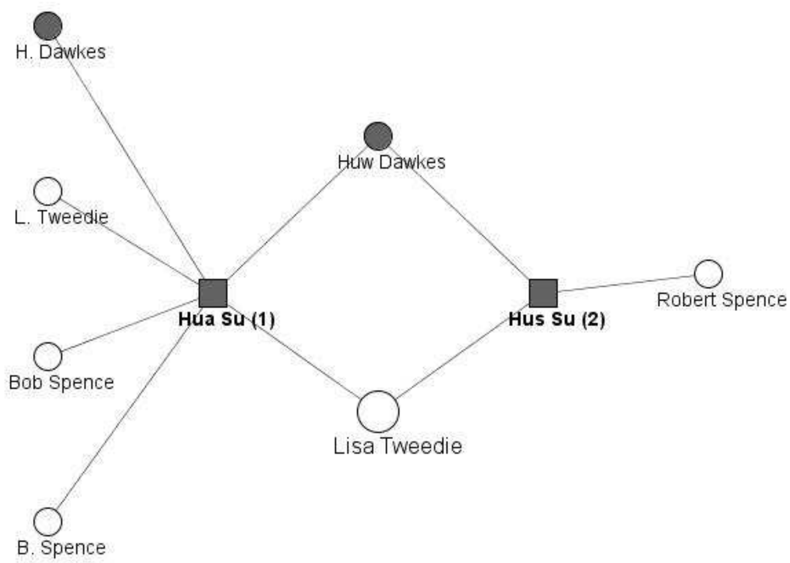
- The Problem
- **Relational Entity Resolution**
- Algorithms

Relational Entity Resolution

- References not observed independently
  - Links between references indicate relations between the entities
  - Co-author relations for bibliographic data
  - To, cc: lists for email
- Use relations to improve identification and disambiguation

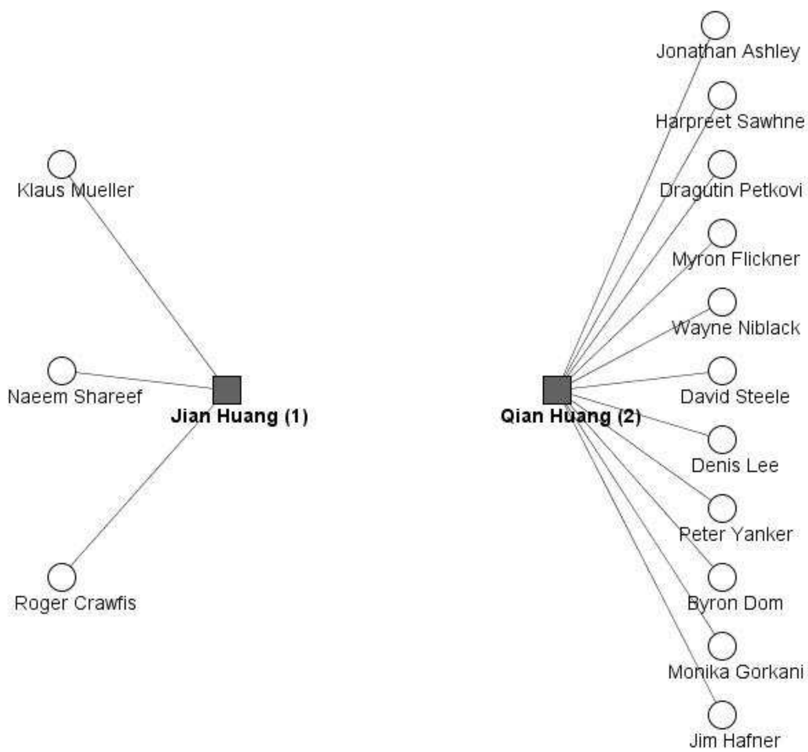
Pasula et al. 03, Ananthakrishna et al. 02, Bhattacharya & Getoor 04,06,07, McCallum & Wellner 04, Li, Morie & Roth 05, Culotta & McCallum 05, Kalashnikov et al. 05, Chen, Li, & Doan 05, Singla & Domingos 05, Dong et al. 05

Relational Identification



Very similar names.
Added evidence from shared co-authors

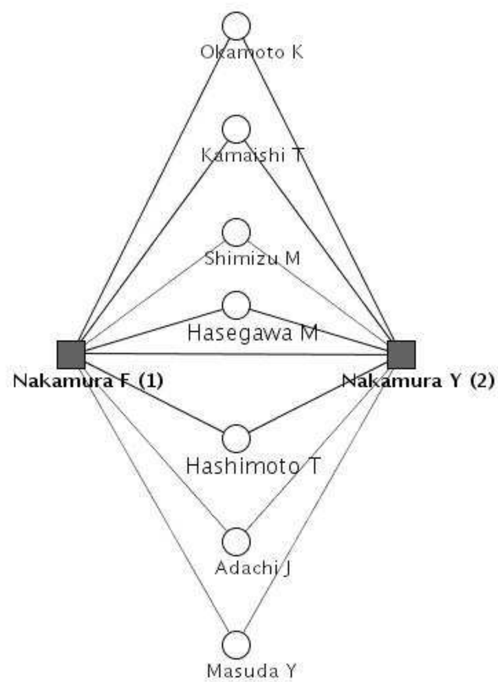
Relational Disambiguation



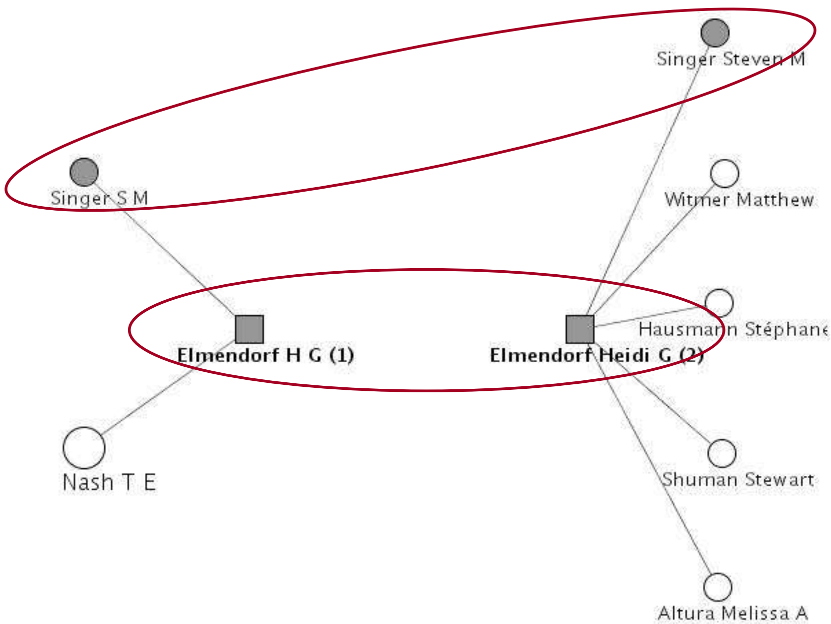
Very similar names but no shared collaborators

Relational Constraints

Co-authors are typically distinct



Collective Entity Resolution



One resolution provides evidence for another => joint resolution

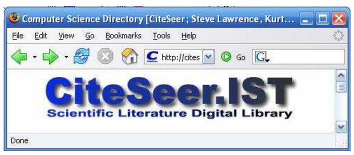
Entity Resolution with Relations

- Naïve Relational Entity Resolution
  - Also compare attributes of related references
  - Two references have co-authors w/ similar names

- **Collective Entity Resolution**
  - Use **discovered entities** of related references
  - Entities cannot be identified independently
  - Harder problem to solve

Entity Resolution

- The Problem
- Relational Entity Resolution
- **Algorithms**
  - **Relational Clustering (RC-ER)**
    - *Bhattacharya & Getoor, DMKD’04, Wiley’06, DE Bulletin’06,TKDD’07*



P1: **“***JOSTLE: Partitioning of Unstructured Meshes for Massively Parallel Machines”*, C. Walshaw, M. Cross, M. G. Everett, S. Johnson

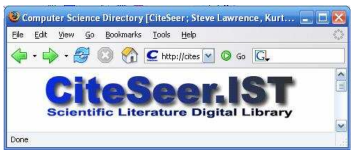
P2: **“***Partitioning Mapping of Unstructured Meshes to Parallel Machine Topologies”*, C. Walshaw, M. Cross, M. G. Everett, S. Johnson, K. McManus

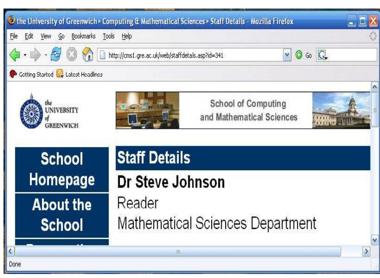
P3: **“***Dynamic Mesh Partitioning: A Unied Optimisation and Load-Balancing Algorithm”*, C. Walshaw, M. Cross, M. G. Everett

P4: **“***Code Generation for Machines with Multiregister Operations”*, Alfred V. Aho, Stephen C. Johnson, Jefferey D. Ullman

P5: **“***Deterministic Parsing of Ambiguous Grammars”*, A. Aho, S. Johnson, J. Ullman

P6: **“***Compilers: Principles, Techniques, and Tools”*, A. Aho, R. Sethi, J. Ullman





P1: "*JOSTLE: Partitioning of Unstructured Meshes for Massively Parallel Machines"*, C. Walshaw, M. Cross, M. G. Everett, **S. Johnson**

P2: "*Partitioning Mapping of Unstructured Meshes to Parallel Machine Topologies"*, C. Walshaw, M. Cross, M. G. Everett, **S. Johnson**, K. McManus

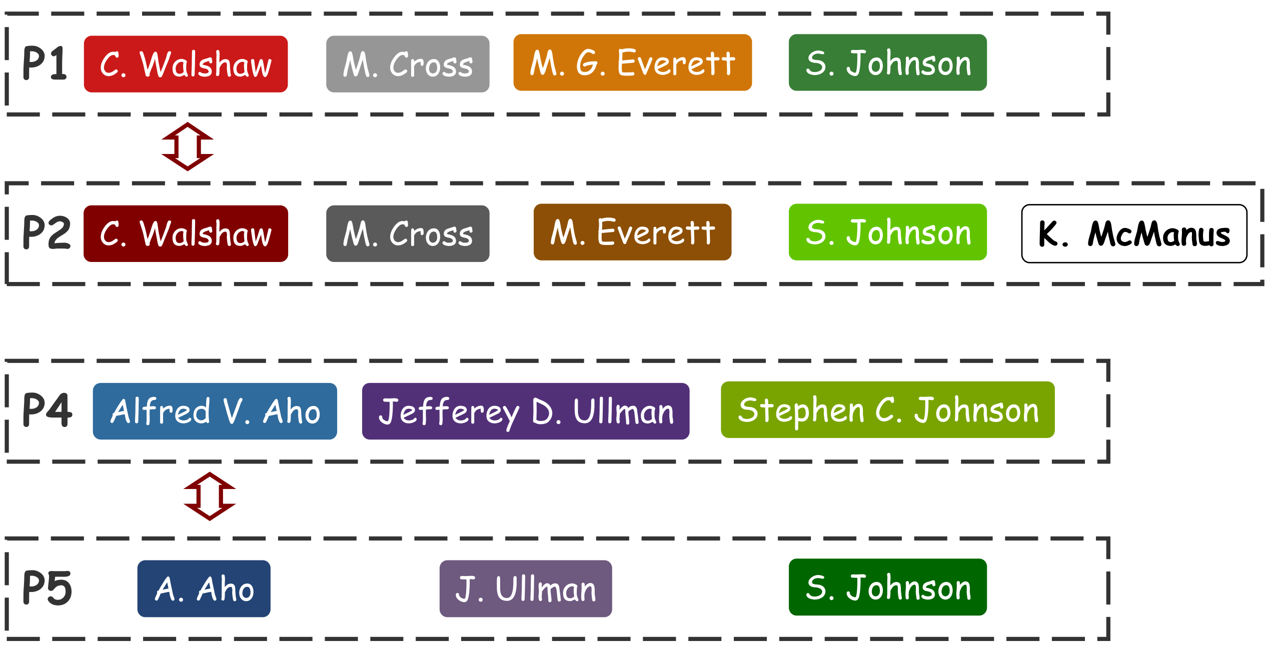
P3: "*Dynamic Mesh Partitioning: A Unied Optimisation and Load-Balancing Algorithm"*, C. Walshaw, M. Cross, M. G. Everett

P4: "*Code Generation for Machines with Multiregister Operations"*, Alfred V. Aho, **Stephen C. Johnson**, Jefferey D. Ullman

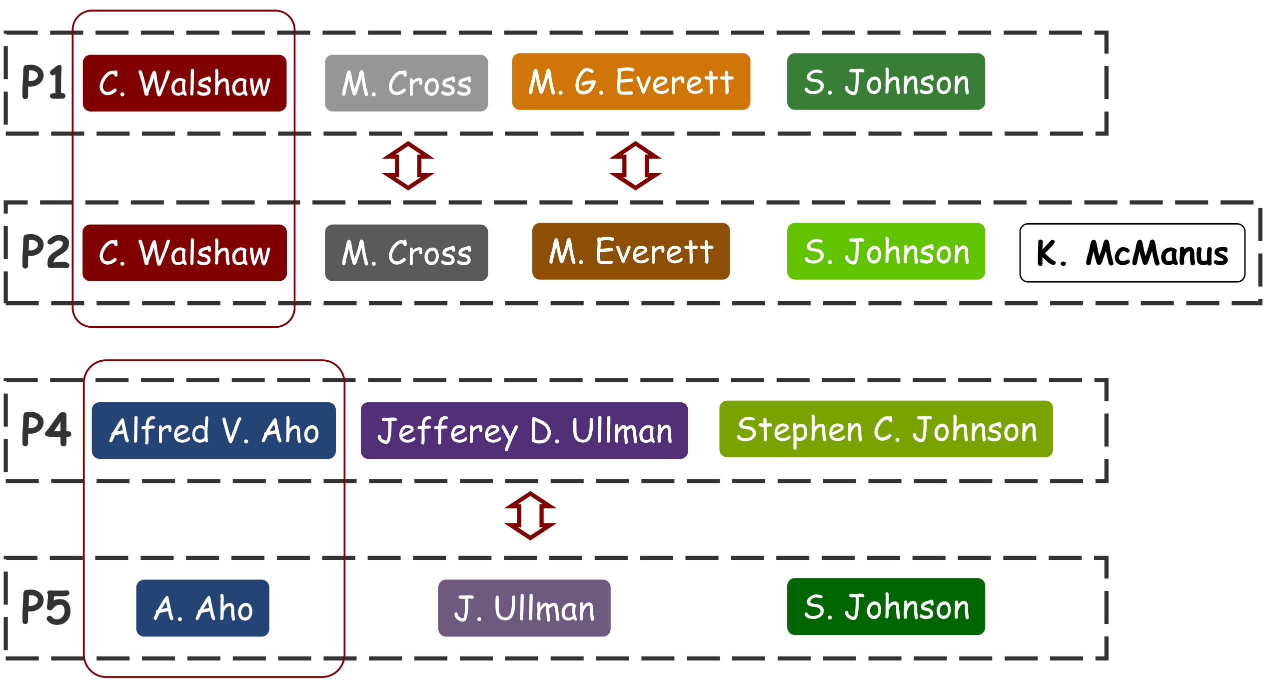
P5: "*Deterministic Parsing of Ambiguous Grammars"*, A. Aho, **S. Johnson**, J. Ullman

P6: "*Compilers: Principles, Techniques, and Tools"*, A. Aho, R. Sethi, J. Ullman

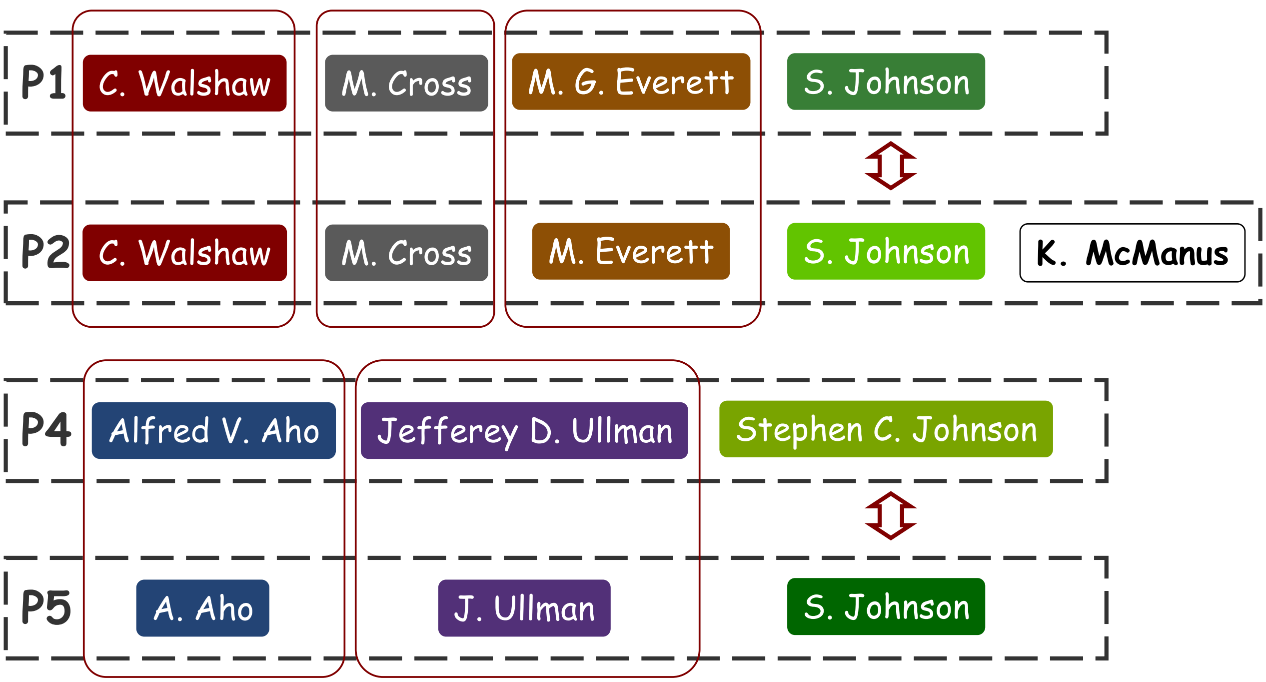
Relational Clustering (RC-ER)



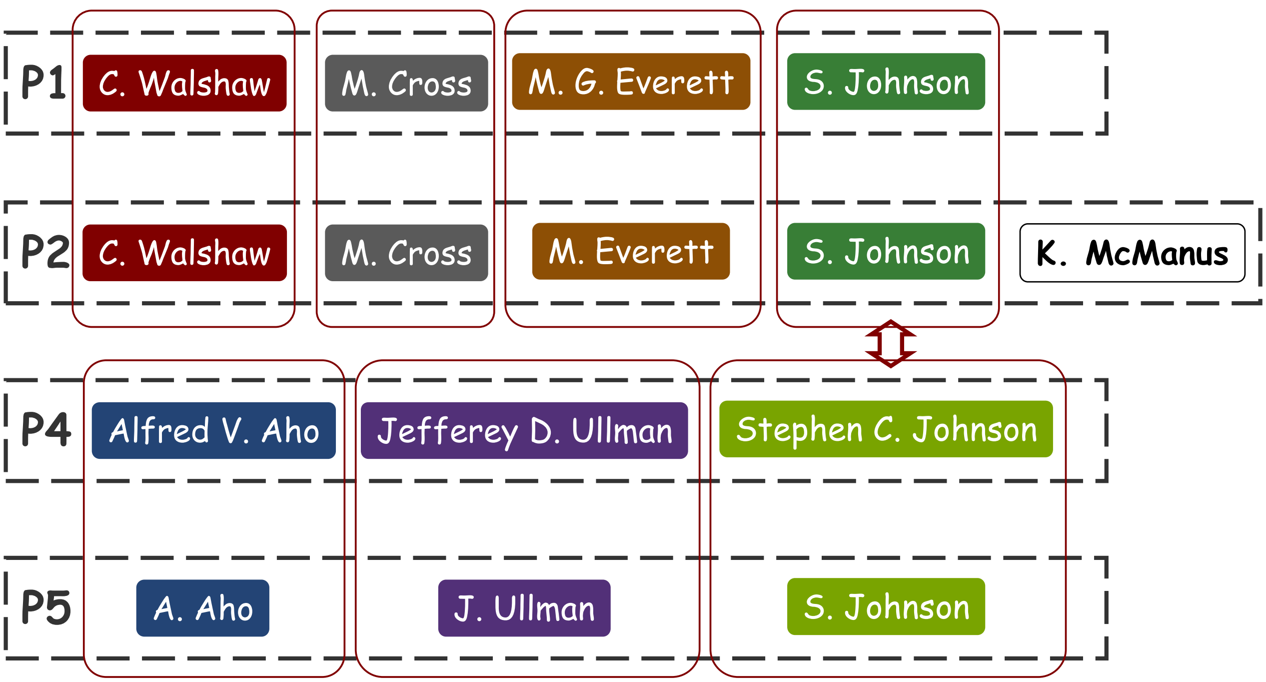
Relational Clustering (RC-ER)



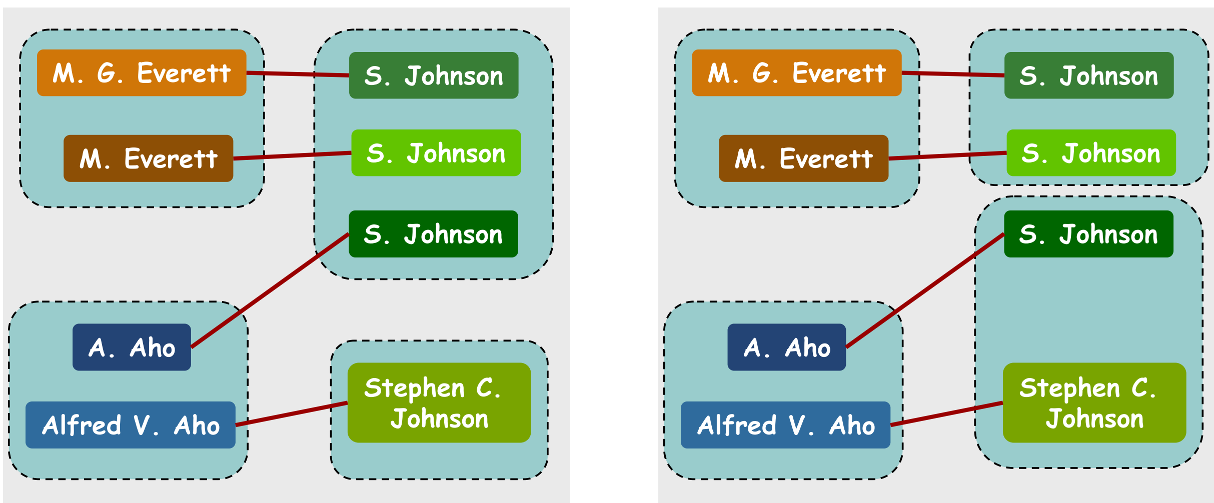
Relational Clustering (RC-ER)



Relational Clustering (RC-ER)



Cut-based Formulation of RC-ER



Good separation of attributes
Many cluster-cluster relationships

- Aho-Johnson1, Aho-Johnson2, Everett-Johnson1

Worse in terms of attributes
Fewer cluster-cluster relationships

- Aho-Johnson1, Everett-Johnson2

Objective Function

- **Minimize:**

$$\sum_i \sum_j w_A sim_A(c_i, c_j) + w_R sim_R(c_i, c_j)$$

weight for attributes | similarity of attributes | weight for relations | Similarity based on relational edges between c_i and c_j

- **Greedy clustering algorithm:** merge cluster pair with max reduction in objective function

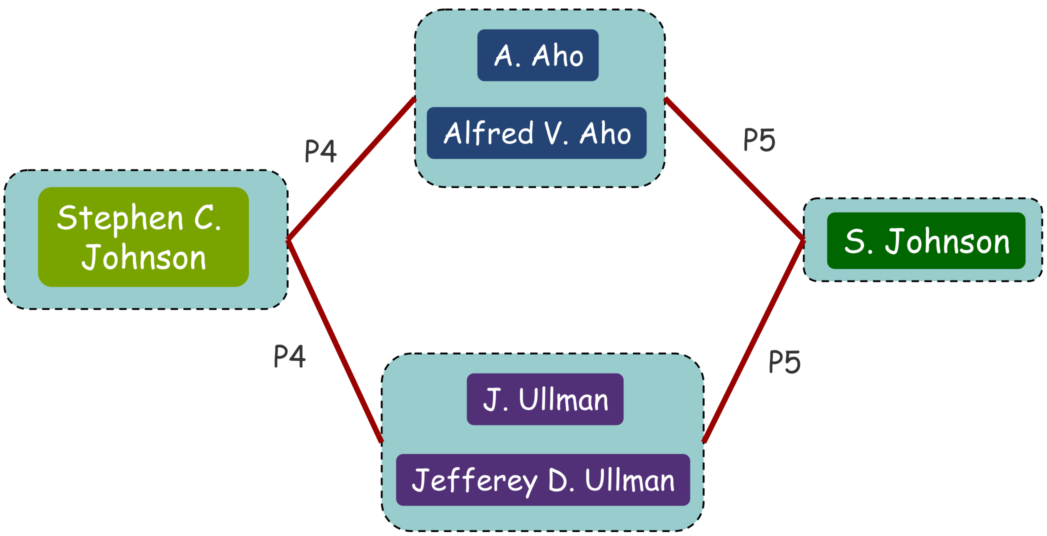
$$\Delta(c_i, c_j) = w_A sim_A(c_i, c_j) + w_R (|N(c_i)| \bigcap |N(c_j)|)$$

Similarity of attributes | Common cluster neighborhood

Measures for Attribute Similarity

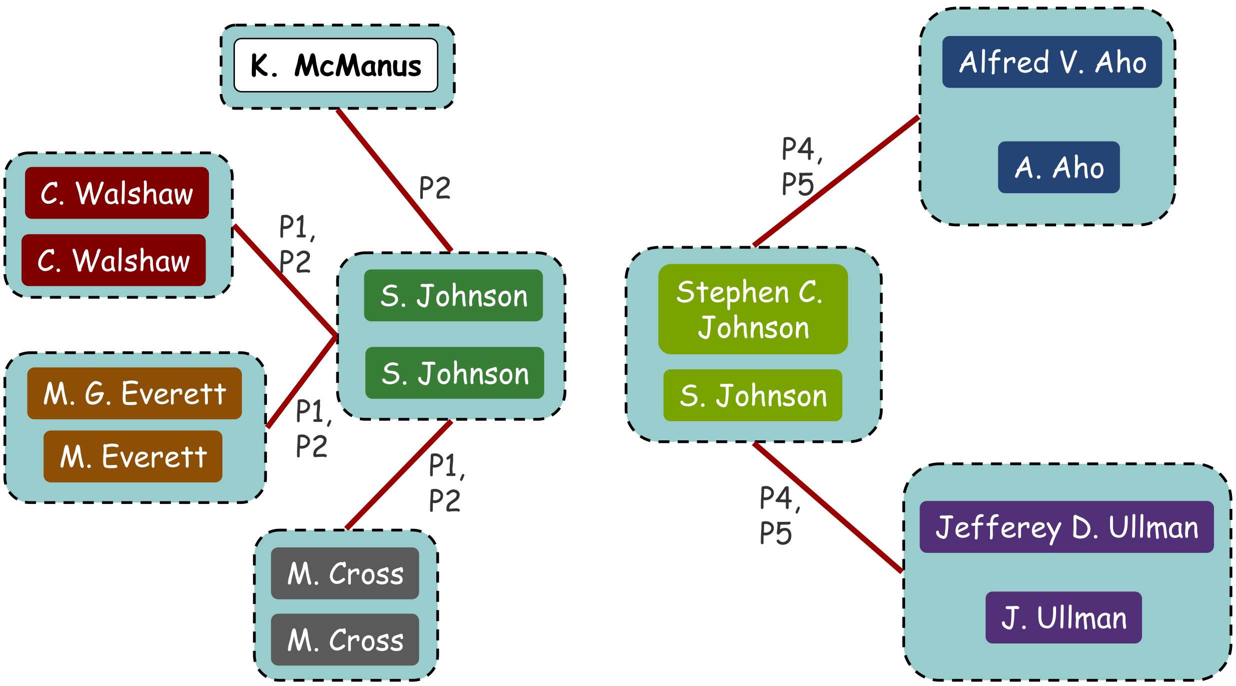
- Use best available measure for each attribute
  - Name Strings: *Soft TF-IDF, Levenstein, Jaro*
  - Textual Attributes: *TF-IDF*
- Aggregate to find similarity between clusters
  - Single link, Average link, Complete link
  - Cluster representative

Relational Similarity: Example 1



All neighborhood clusters are shared: high relational similarity

Relational Similarity: Example 2



No neighborhood cluster is shared: no relational similarity

Comparing Cluster Neighborhoods

- Consider neighborhood as multi-set

- Different measures of set similarity
  - Common Neighbors: Intersection size
  - Jaccard's Coefficient: Normalize by union size
  - Adar Coefficient: Weighted set similarity
  - Higher order similarity: Consider neighbors of neighbors

Relational Clustering Algorithm

1. Find similar references using ‘blocking’
2. Bootstrap clusters using attributes and relations
3. Compute similarities for cluster pairs and insert into priority queue
4. Repeat until priority queue is empty
5. Find ‘closest’ cluster pair
6. Stop if similarity below threshold
7. Merge to create new cluster
8. Update similarity for ‘related’ clusters

- O(n k log n) algorithm w/ efficient implementation

Entity Resolution

- The Problem
- Relational Entity Resolution
- **Algorithms**
  - Relational Clustering (RC-ER)
  - **Probabilistic Model (LDA-ER)**
    - *SIAM SDM'06, Best Paper Award*
  - Experimental Evaluation

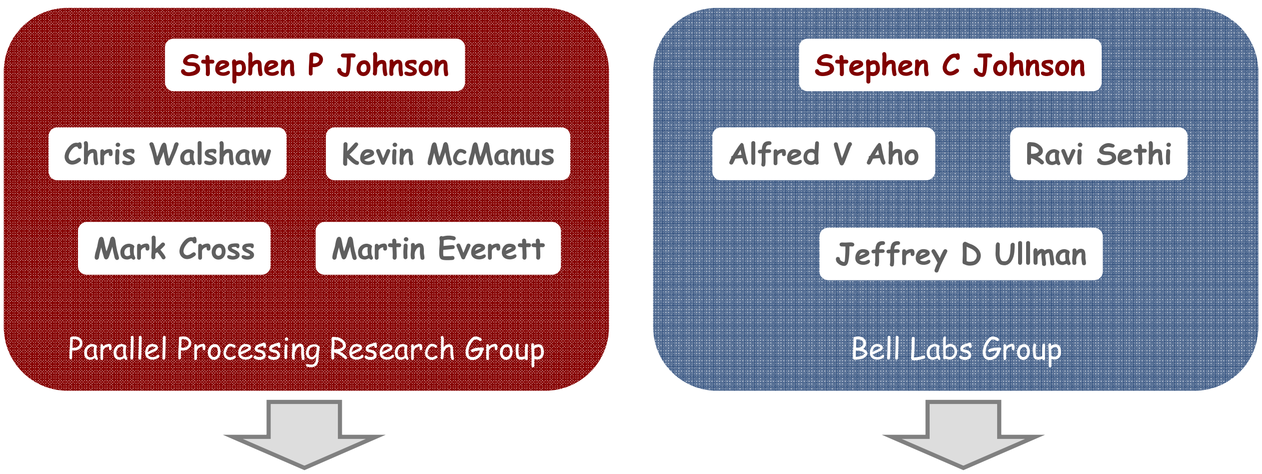
Probabilistic Generative Model for Collective Entity Resolution

- Model how references co-occur in data

  1. Generation of references from entities

  2. Relationships between underlying entities
     - Groups of entities instead of pair-wise relations

Discovering Groups from Relations



P1: C. Walshaw, M. Cross, M. G. Everett, **S. Johnson**

P2: C. Walshaw, M. Cross, M. G. Everett, **S. Johnson**, K. McManus

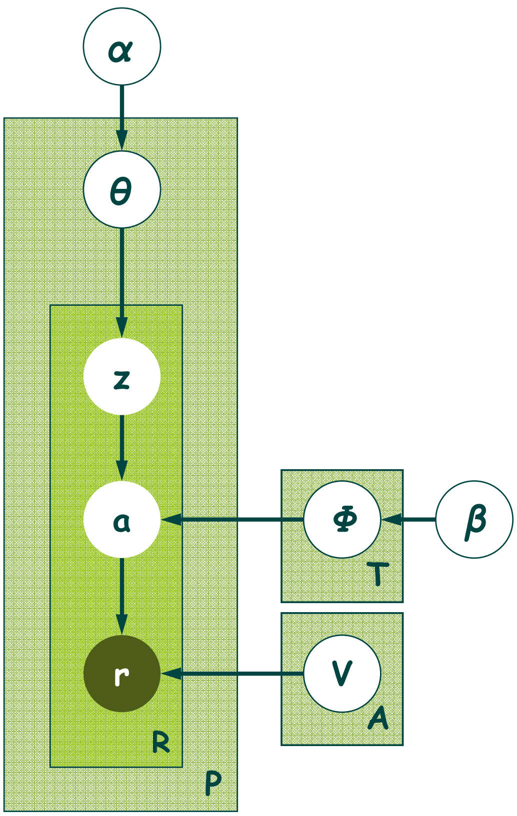
P3: C. Walshaw, M. Cross, M. G. Everett

P4: Alfred V. Aho, **Stephen C. Johnson**, Jefferey D. Ullman

P5: A. Aho, **S. Johnson**, J. Ullman

P6: A. Aho, R. Sethi, J. Ullman

LDA-ER Model

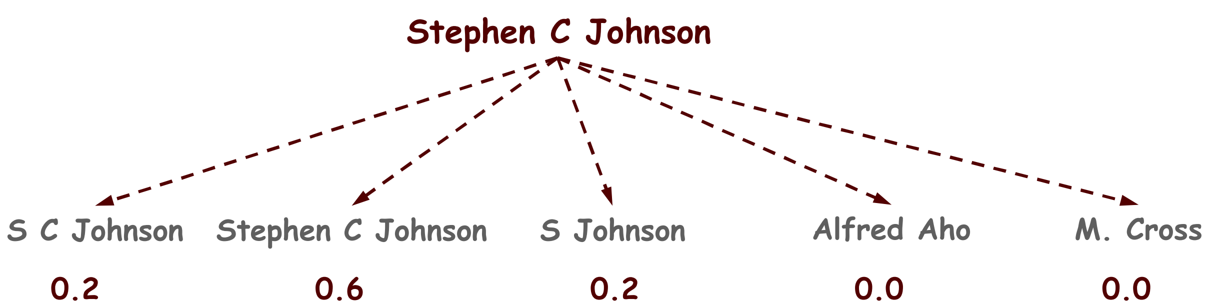


- Entity label ***a*** and group label ***z*** for each reference ***r***
- ***Θ*: '**mixture' of groups for each co-occurrence
- $\boldsymbol{\Phi_z}$**:** multinomial for choosing entity ***a*** for each group ***z***
- $\boldsymbol{V_a}$**:** multinomial for choosing reference ***r*** from entity ***a***
- Dirichlet priors with ***α*** and ***β***

Generating References from Entities

- Entities are not directly observed
  1. Hidden attribute for each entity
  2. Similarity measure for pairs of attributes

- A distribution over attributes for each entity



Approx. Inference Using Gibbs Sampling

- Conditional distribution over labels for each ref.
- Sample next labels from conditional distribution
- Repeat over all references until convergence

$$P(z_i = t \mid \mathbf{z}_{-i}, \mathbf{a}, \mathbf{r}) \propto \frac{n^{DT}_{d_i t} + \alpha/T}{n^{DT}_{d_i *} + \alpha} \times \frac{n^{AT}_{a_i t} + \beta/A}{n^{AT}_{* t} + \beta}$$

$$P(a_i = a \mid \mathbf{z}, \mathbf{a}_{-i}, \mathbf{r}) \propto \frac{n^{AT}_{a_i t} + \beta/A}{n^{AT}_{* t} + \beta} \times Sim(r_i, v_a)$$

- Converges to most likely number of entities

Faster Inference: Split-Merge Sampling

- Naïve strategy reassigns references individually
- Alternative: allow entities to merge or split
- For entity a_i, find conditional distribution for
  1. Merging with existing entity a_j
  2. Splitting back to last merged entities
  3. Remaining unchanged
- Sample next state for a_i from distribution
- O(n g + e) time per iteration compared to O(n g + n e)

Entity Resolution

- The Problem
- Relational Entity Resolution
- **Algorithms**
  - Relational Clustering (RC-ER)
  - Probabilistic Model (LDA-ER)
  - **Experimental Evaluation**

Evaluation Datasets

- CiteSeer
  - 1,504 citations to machine learning papers (Lawrence et al.)
  - 2,892 references to 1,165 author entities
- arXiv
  - 29,555 publications from High Energy Physics (KDD Cup'03)
  - 58,515 refs to 9,200 authors
- Elsevier BioBase
  - 156,156 Biology papers (IBM KDD Challenge '05)
  - 831,991 author refs
  - Keywords, topic classifications, language, country and affiliation of corresponding author, etc

Baselines

- **A:** Pair-wise duplicate decisions w/ attributes only
  - **Names:** *Soft-TFIDF* with *Levenstein, Jaro, Jaro-Winkler*
  - **Other textual attributes:** *TF-IDF*
- **A*:** Transitive closure over **A**

- **A+N**: Add attribute similarity of co-occurring refs
- **A+N***: Transitive closure over **A+N**

- Evaluate pair-wise decisions over references
- F1-measure (harmonic mean of precision and recall)

ER over Entire Dataset

	CiteSeer	arXiv	BioBase
A	0.980	0.976	0.568
A*	0.990	0.971	0.559
A+N	0.973	0.938	0.710
A+N*	0.984	0.934	0.753
RC-ER	**0.995**	**0.985**	**0.818**
LDA-ER	0.993	0.981	0.645

- RC-ER & LDA-ER outperform baselines in all datasets
- Collective resolution better than naïve relational resolution
- RC-ER and baselines require threshold as parameter
  - Best achievable performance over all thresholds
- Best RC-ER performance better than LDA-ER
- LDA-ER does not require similarity threshold

Bhattacharya and Getoor, TKDD 07

ER over Entire Dataset

	CiteSeer	arXiv	BioBase
A	0.980	0.976	0.568
A*	0.990	0.971	0.559
A+N	0.973	0.938	0.710
A+N*	0.984	0.934	0.753
RC-ER	**0.995**	**0.985**	**0.818**
LDA-ER	0.993	0.981	0.645

- CiteSeer: Near perfect resolution; 22% error reduction
- arXiv: 6,500 additional correct resolutions; 20% error reduction
- BioBase: Biggest improvement over baselines

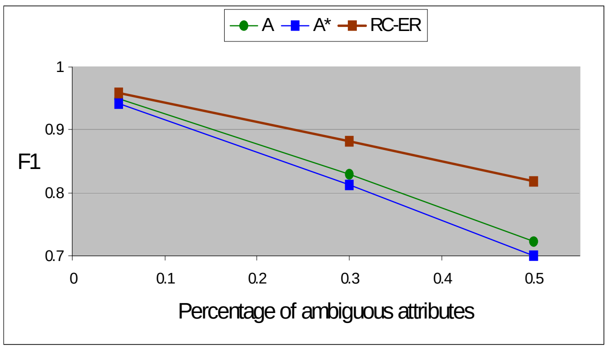
Performance for Specific Names

Name	Best F1 for ATTR/ATTR*	F1 for LDA-ER
cho_h	0.80	1.00
davis_a	0.67	0.89
kim_s	0.93	0.99
kim_y	0.93	0.99
lee_h	0.88	0.99
lee_j	0.98	1.00
liu_j	0.95	0.97
sarkar_s	0.67	1.00
sato_h	0.82	0.97
sato_t	0.85	1.00
shin_h	0.69	1.00
veselov_a	0.78	1.00
yamamoto_k	0.29	1.00
yang_z	0.77	0.97
zhang_r	0.83	1.00
zhu_z	0.57	1.00

arXiv

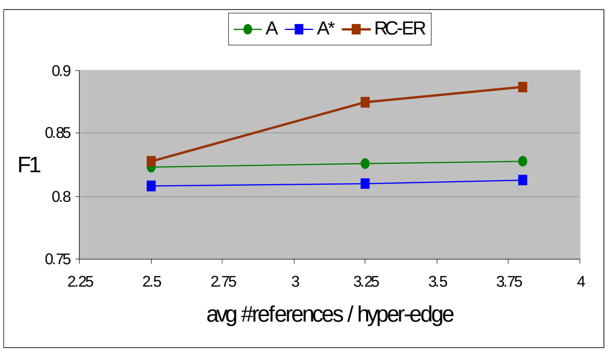
Significantly larger improvements for 'ambiguous names'

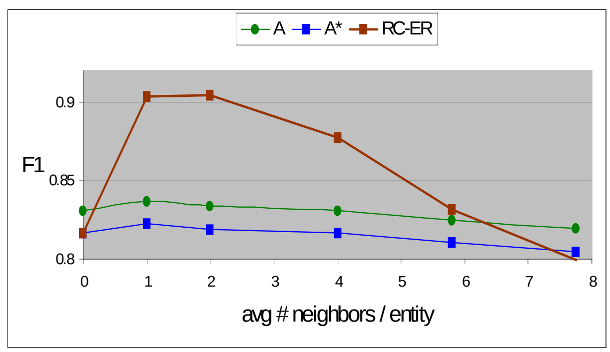
Trends in Synthetic Data



Bigger improvement with

- bigger % of ambiguous refs
- more refs per co-occurrence
- more neighbors per entity





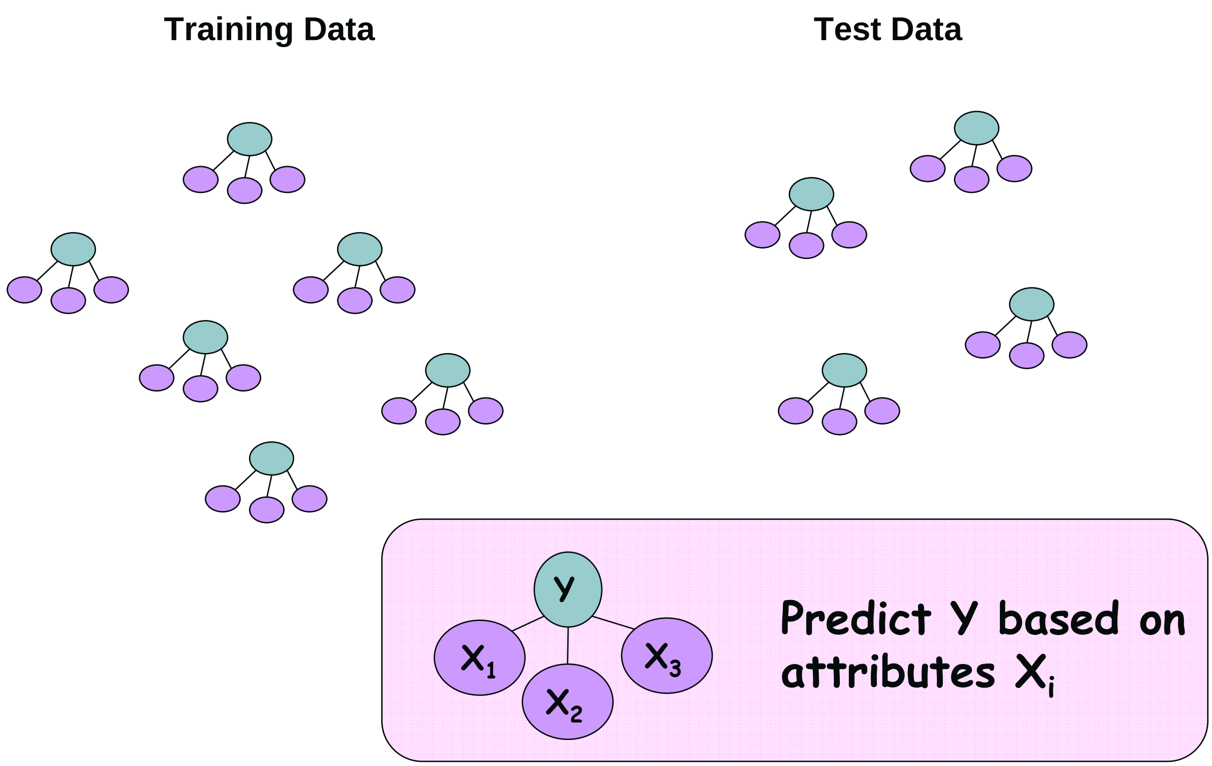
Roadmap

- The Problem
- **The Components**
  - Entity Resolution
  - **Collective Classification**
  - Link Prediction
  - Putting It All Together
- Open Questions

Collective Classification

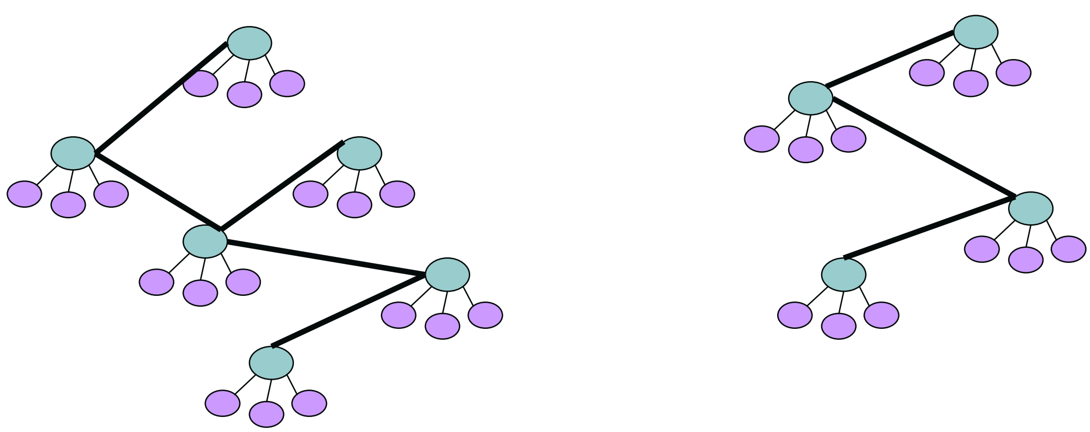
- **The Problem**
- Collective Relational Classification
- Algorithms

Traditional Classification



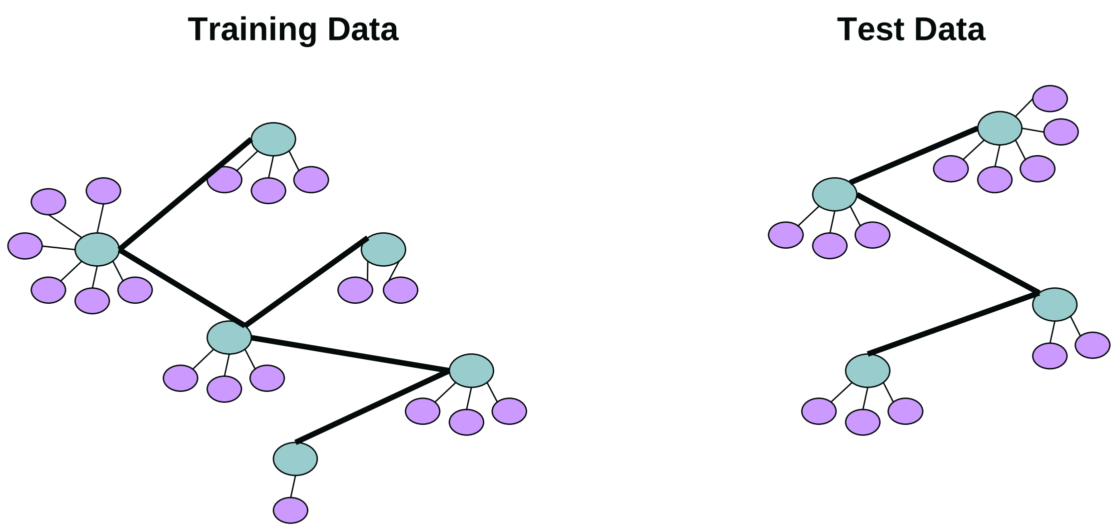
Relational Classification (1)

Training Data **Test Data**



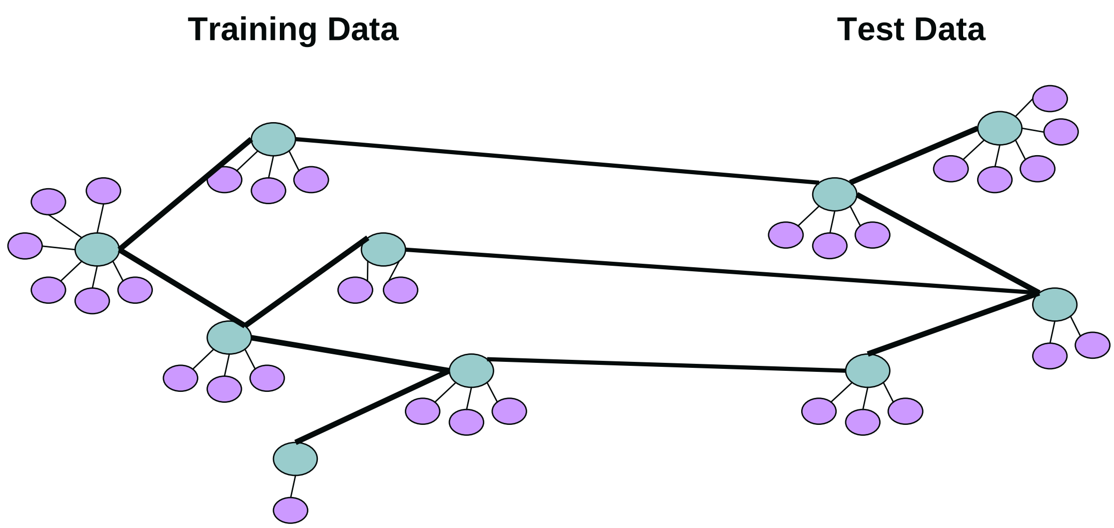
Correlations among linked instances
autocorrelation: labels are likely to be the same
homophily: similar nodes are more likely to be linked

Relational Classification (2)



Irregular graph structure

Relational Classification (3)



Links between training set & test set
learning with partial labels or within network classification

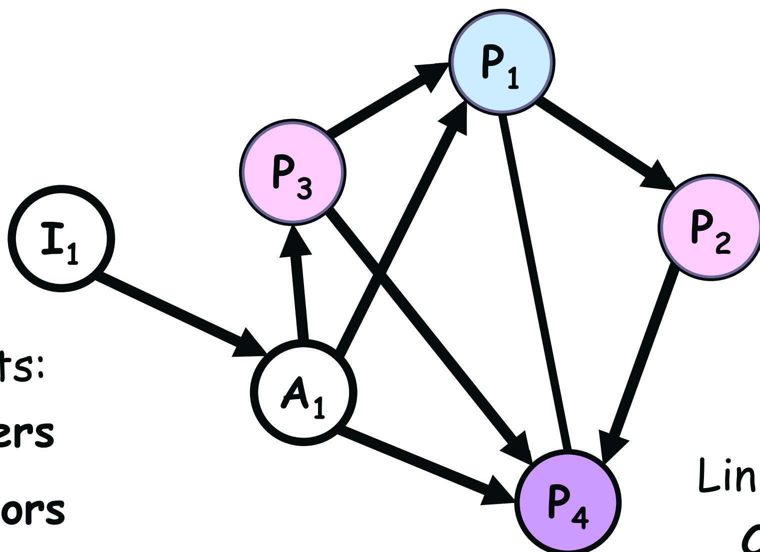
The Problem

- Relational Classification: predicting the category of an object based on its attributes *and* its links *and* attributes of linked objects

- Collective Classification: jointly predicting the categories for a collection of connected, unlabelled objects

Neville & Jensen 00, Taskar , Abbeel & Koller 02, Lu & Getoor 03, Neville, Jensen & Galliger 04, Sen & Getoor TR07, Macskassy & Provost 07, Gupta, Diwam & Sarawagi 07, Macskassy AAAI07, McDowell, Gupta & Aha AAAI07

Example: Linked Bibliographic Data



Objects:
- Papers
- Authors
- Institutions

Links:
- Citation
- Co-Citation
- Author-of
- Author-affiliation

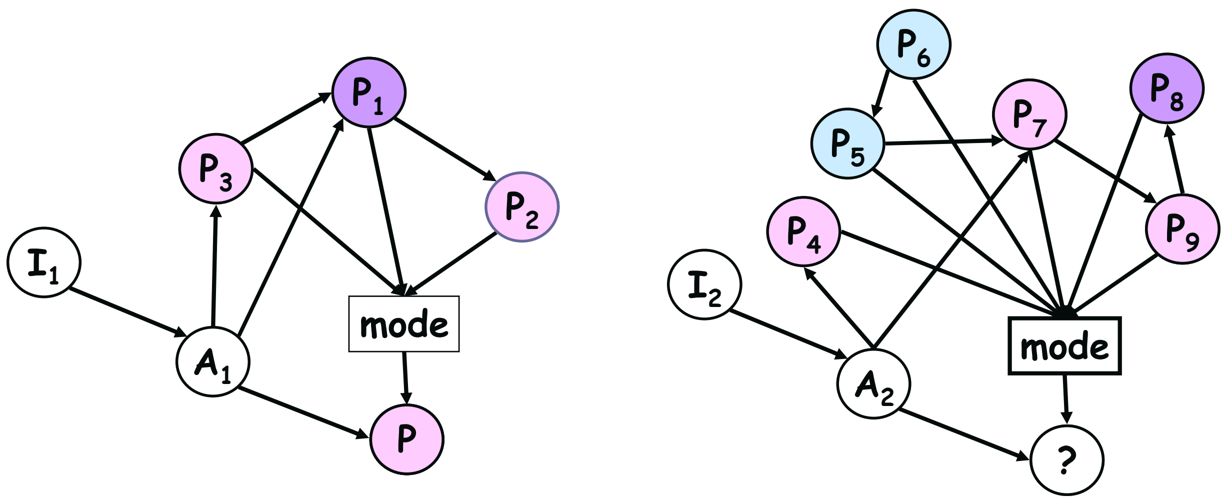
Labels:

Feature Construction

- Objects are linked to a **set** of objects. To construct features from this set of objects, we need feature aggregation methods

Perlich & Provost 03, 04, 05, Popescul & Ungar 03, Lu & Getoor 03, Gupta, Diwam & Sarawagi 07

Simple Aggregation



Other aggregates: count, min, max, prop, exists, selection

Feature Construction

- In many cases, objects are linked to a **set** of objects. To construct features from this set of objects, we need feature aggregation methods

- Instances vs. generics
  - Features may refer
    - explicitly to individuals
    - classes or generic categories of individuals
  - On one hand, want to model that a particular individual may be highly predictive
  - On the other hand, want models to generalize to new situations, with different individuals

Aggregate Features Used

	Mode	Prop	Count	Exists	SQL	FOL
PRMs, Friedman et al.	X				X	
RMNs, Taskar et al.					X	
MLNs, Domingos et al.						X
RDNs, Neville et al.						X
Lu & Getoor ICML03	X		X	X		
Sen & Getoor, TR07	X		X	X		
Maskassy & Provost JMLR07		X				
Gupta et al. ICML07	X		X			
McDowell et al. AAAI07		X				

Formulation

- Directed Model
  - Collection of Local Conditional Models
  - Inference Algorithms:
    - Iterative Classification Algorithm (ICA)
    - Gibbs Sampling (Gibbs)

- Undirected Model
  - (Pairwise) Markov Random Fields
  - Inference Algorithms:
    - Loopy Belief Propagation (LBP)
    - Gibbs Sampling
    - Mean Field Relaxation Labeling (MF)

CC Inference Algorithms

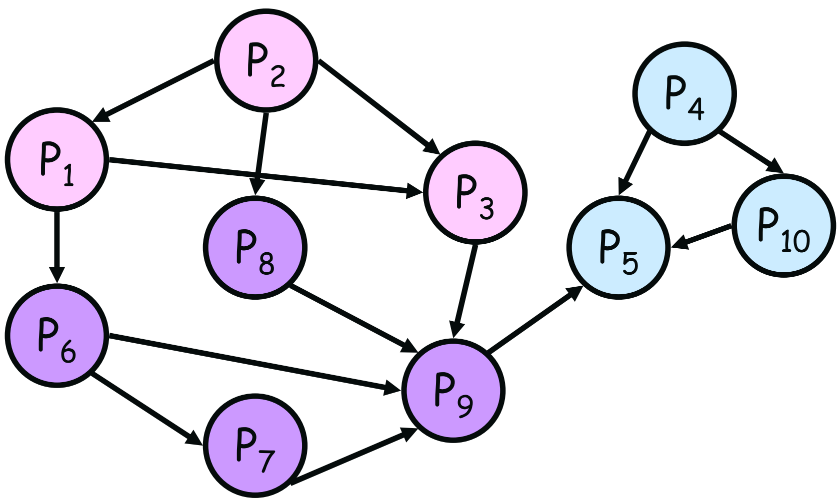
	MF	LBP	Gibbs	ICA
Chakrabarti et al SIGMOD98	X			
Jensen & Neville SRL00				X
Getoor et al. IJCAI WS		X		
Taskar et al. UAI02		X		
Lu & Getoor ICML03				X
Neville & Jensen KDD04			X	
Sen & Getoor, TR07	X	X		X
Maskassy & Provost JMLR07	X		X	X
Gupta et al. ICML07		X		X
McDowell et al. AAAI07			X	X

Local Classifiers Used in ICA

	NB	LR	DT	kNN	wvRN
Chakrabarti et al. 1998	X				
Jensen & Neville 2000	X				
Lu & Getoor ICML03	X	X			
Neville et al. KDD04	X		X		
Macskassy & Provost JMLR07					X
McDowell et al. AAAI07	X			X	

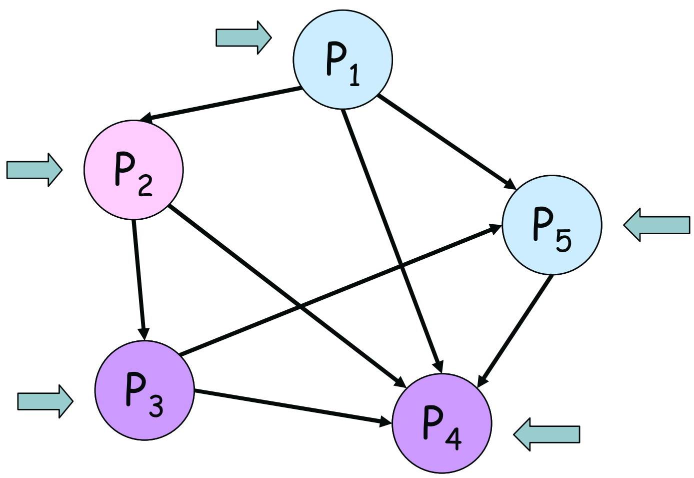
ICA: Learning

- label set:



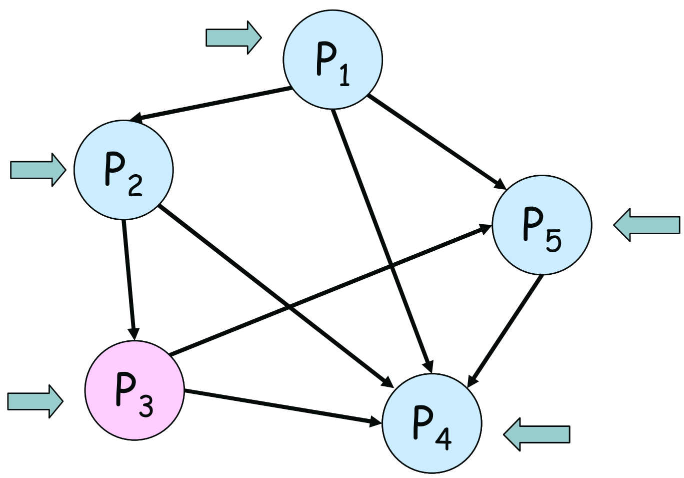
Learn model from fully labeled training set

ICA: Inference (1)



Step 1: Bootstrap using object attributes only

ICA: Inference (2)



Step 2: Iteratively update the category of each object, based on linked object's categories

Experimental Evaluation

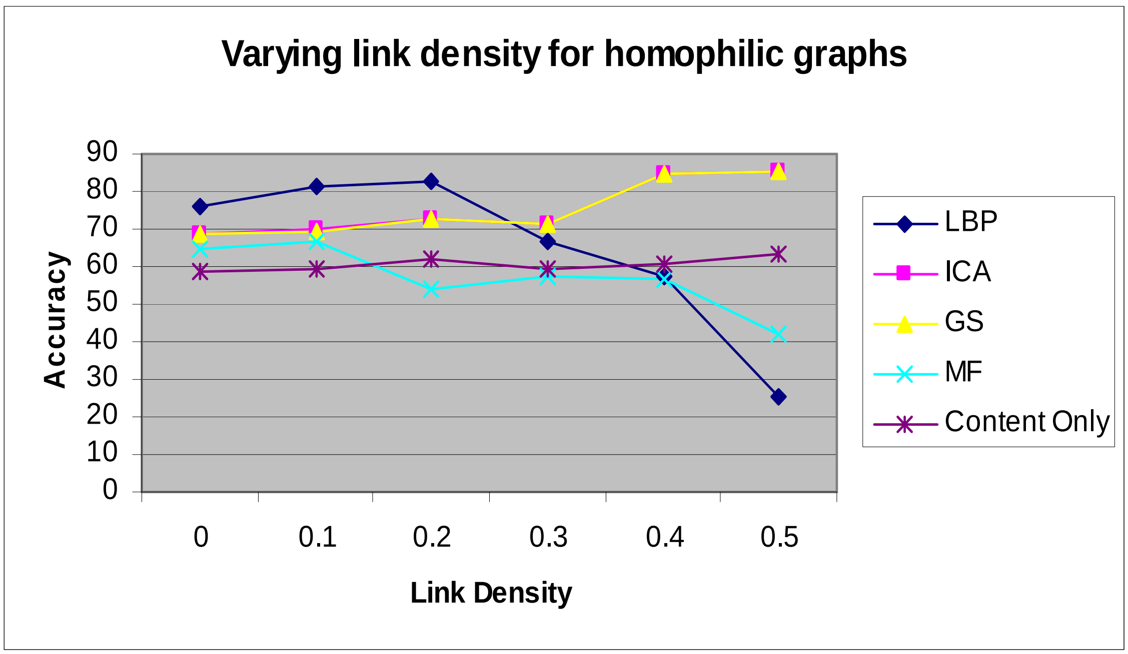
- Comparison of Collective Classification Algorithms
  - Mean Field Relaxation Labeling (MF)
  - Iterative Classification Algorithm (ICA)
  - Loopy Belief Propagation (LBP)
  - Baseline: Content Only
- Datasets
  - Real Data
    - Bibliographic Data (Cora & Citeseer), WebKB, etc.
  - Synthetic Data
    - Data generator which can vary the class label correlations (homophily), attribute noise, and link density

Results on Real Data

Algorithm	Cora	CiteSeer	WebKB
Content Only	66.51	59.77	62.49
ICA	74.99	62.46	**65.99**
Gibbs	74.64	62.52	65.64
MF	79.70	**62.91**	65.65
LBP	**82.48**	62.64	65.13

Sen and Getoor, TR 07

Effect of Structure



Results clearly indicate that algorithms' performance depends (in non-trivial ways) on structure

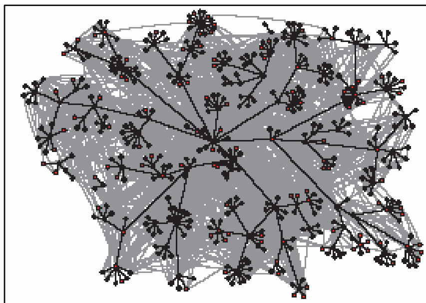
Roadmap

- The Problem
- **The Components**
  - Entity Resolution
  - Collective Classification
  - **Link Prediction**
  - Putting It All Together
- Open Questions

Link Prediction

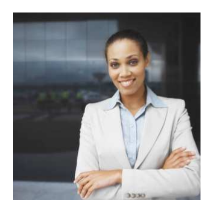
- **The Problem**
- Predicting Relations
- Algorithms
  - Link Labeling
  - Link Ranking
  - Link Existence

Links in Data Graph

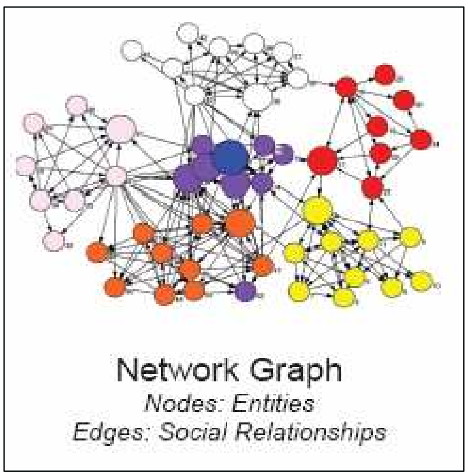


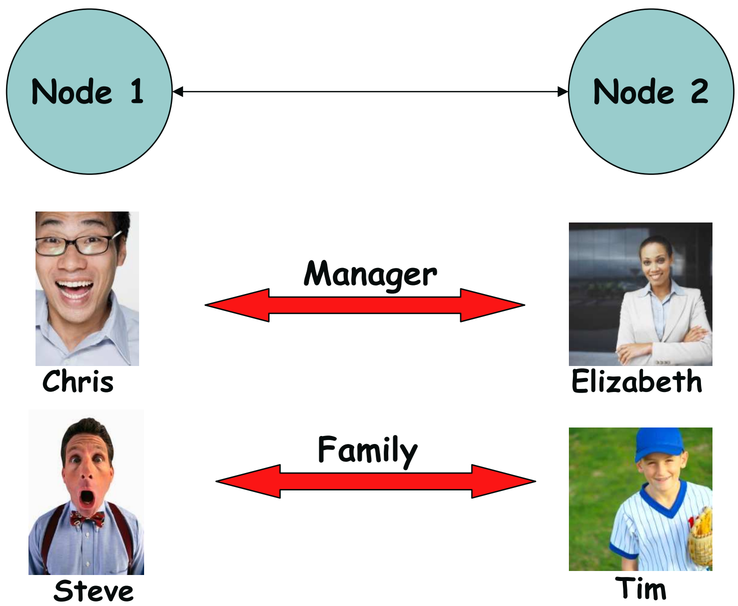
Communications Graph
Nodes: Network References
Edges: Communications Events

chris@enron.com	Email ⟷	liz@enron.com
chris37	IM ⟷	lizs22
555-450-0981	TXT ⟷	555-901-8812



⇒ Links in Information Graph





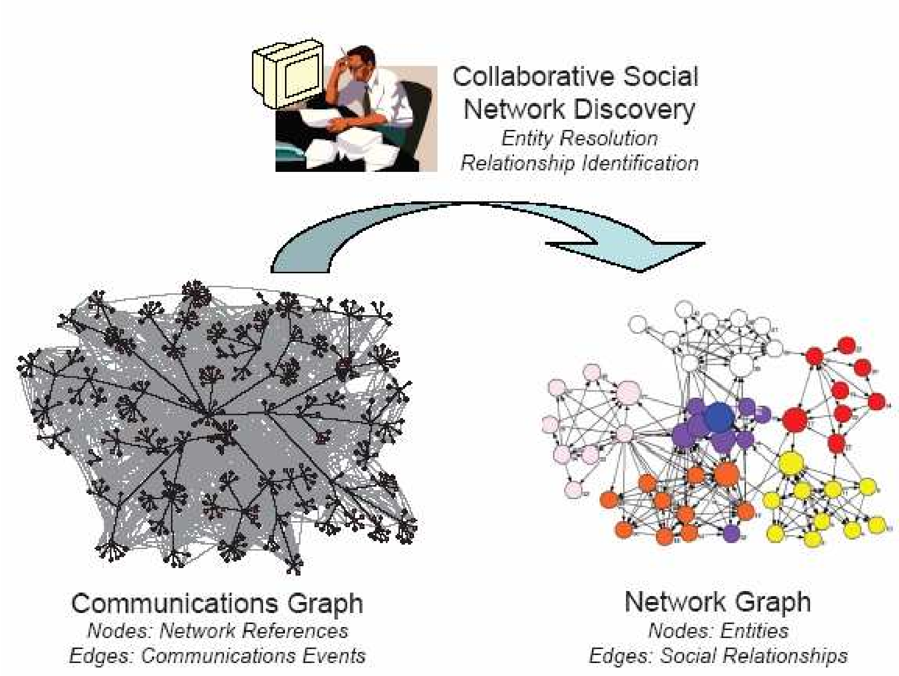
Predicting Relations

- Link Labeling
  - Can use similar approaches to collective classification
- Link Ranking
  - Many variations
    - Diehl, Namata, Getoor, *Relationship Identification for Social Network Discovery*, AAAI07
  - 'Leak detection'
    - Carvalho & Cohen, SDM07
- Link Existence
  - HARD!
  - Huge class skew problem
  - Variations: Link completion, find missing link

Roadmap

- The Problem
- The Components
- **Putting It All Together**
- Open Questions

Putting Everything together….



PART II: METADATA ALIGNMENT

Ontology Alignment

- **Motivation and goals**
- Short overview of OWL Lite
- The ILIADS method
- Experimental evaluation

Motivation and goals

- No silver bullet on how to represent a domain
  - To use knowledge effectively, we need to integrate multiple ontologies
- Our goals:
  - Improve the quality of computed alignments
    - In a way flexible enough to adapt to a wide variety of inputs
  - Find correlations between the features of the input and the criteria for good quality alignments

The method at a glance

- Produce better quality alignments by
  - using data (instances) effectively and
  - using logical inference (e.g., in OWL) to estimate how good an alignment is

- 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

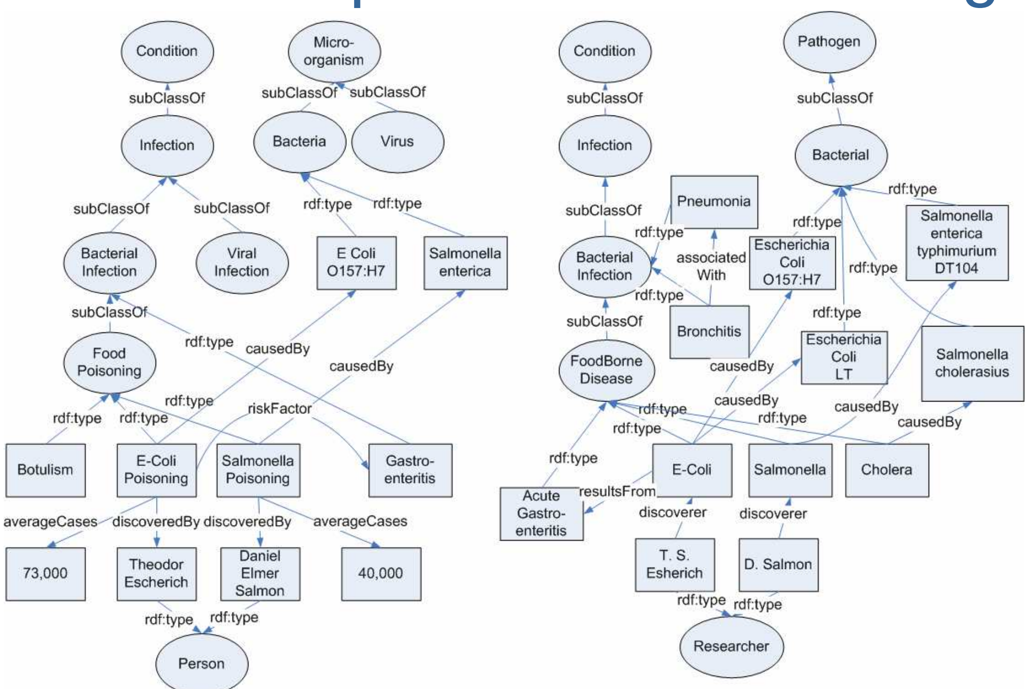
Defining the terms

- **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

Ontology Alignment

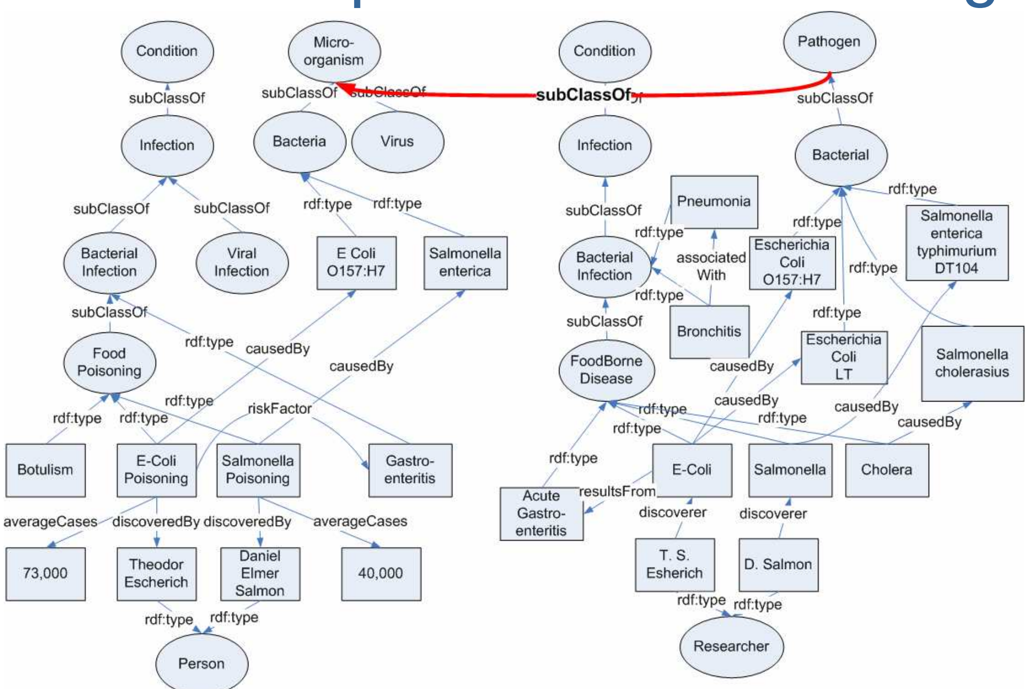
- Motivation and goals
- **Short overview of OWL Lite**
- The ILIADS method
- Experimental evaluation

Example OWL Lite ontologies



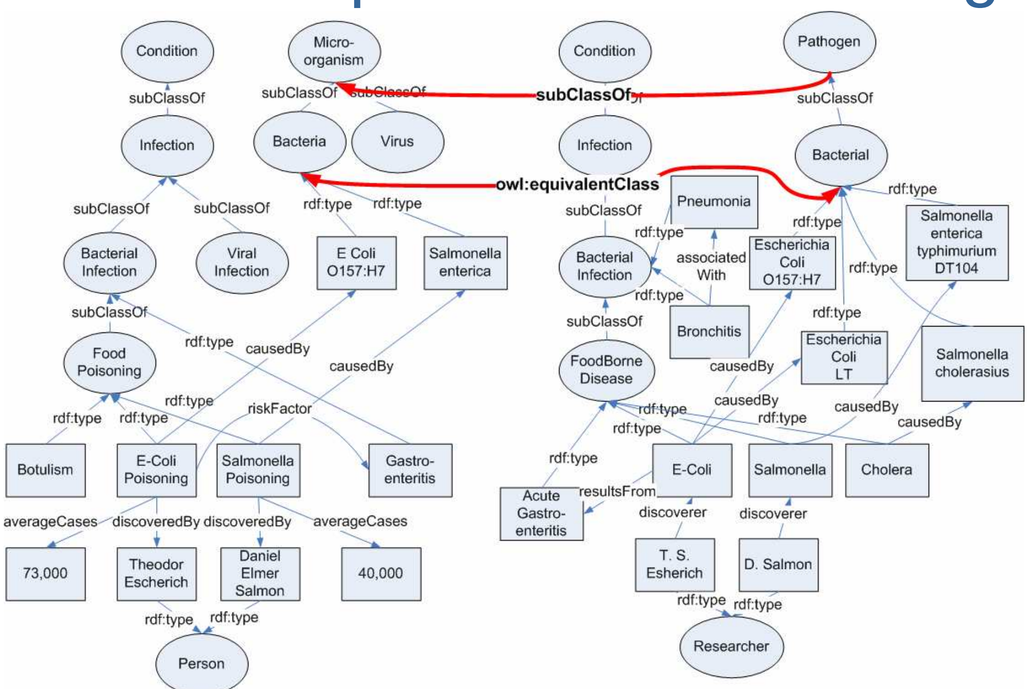
(discoveredBy, owl:inverseOf, discoverer); (discoveredBy, owl:type, owl:FunctionalProperty)
(discoveredBy, owl:inverseOf, discoverer); (associatedWith, owl:type, owl:TransitiveProperty)
(resultsF rom, rdfs:subPropertyOf, associatedWith)

Example OWL Lite ontologies



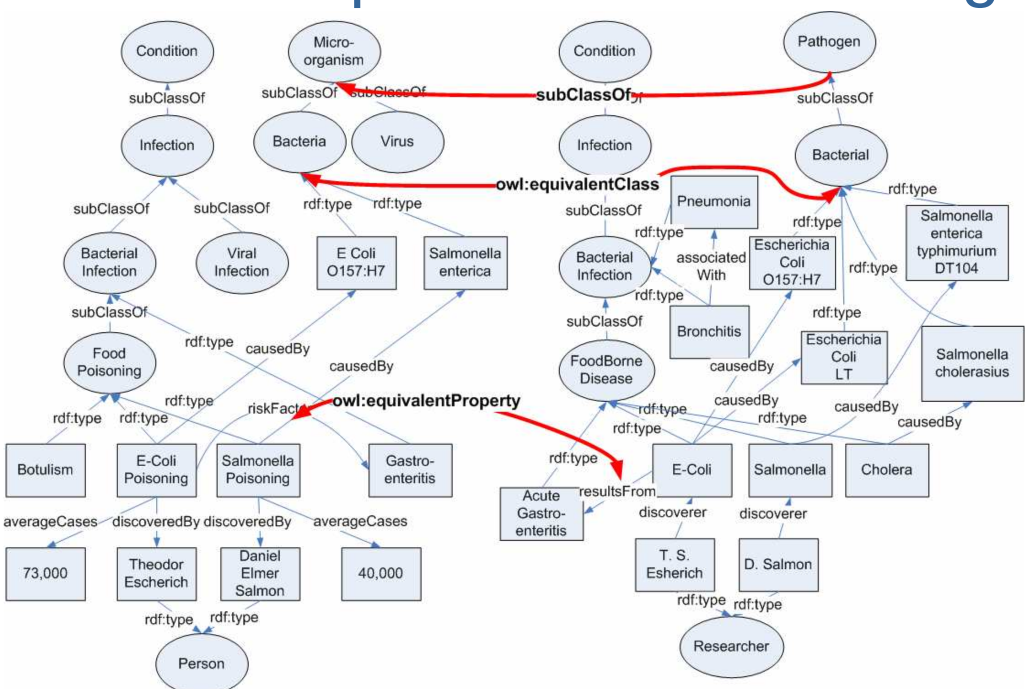
(discoveredBy, owl:inverseOf, discoverer); (discoveredBy, owl:type, owl:FunctionalProperty)
(discoveredBy, owl:inverseOf, discoverer); (associatedWith, owl:type, owl:TransitiveProperty)
(resultsF rom, rdfs:subPropertyOf, associatedWith)

Example OWL Lite ontologies



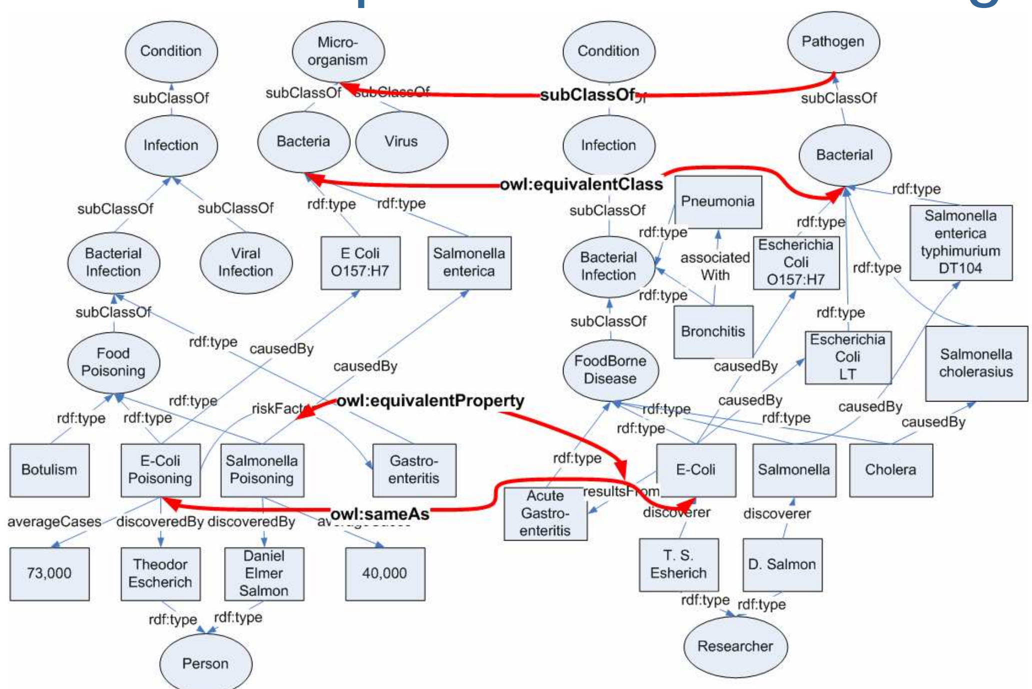
(discoveredBy, owl:inverseOf, discoverer); (discoveredBy, owl:type, owl:FunctionalProperty)
(discoveredBy, owl:inverseOf, discoverer); (associatedWith, owl:type, owl:TransitiveProperty)
(resultsF rom, rdfs:subPropertyOf, associatedWith)

Example 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 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)

Inference in OWL (Lite)

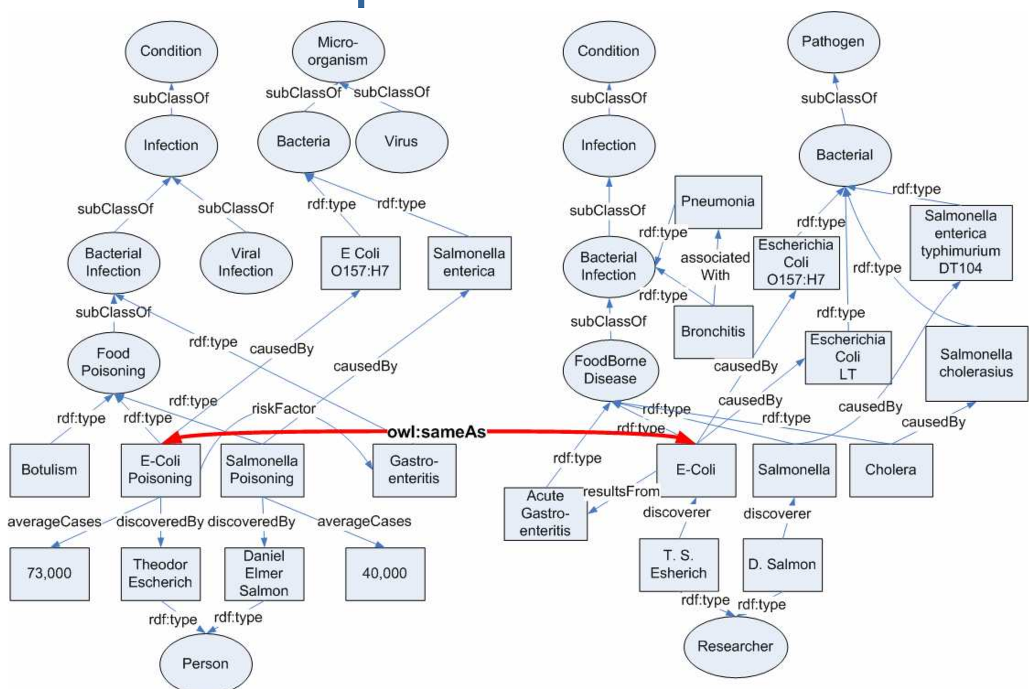
- A tableau-based method
- Example tableau rule:

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

- Example inconsistency:

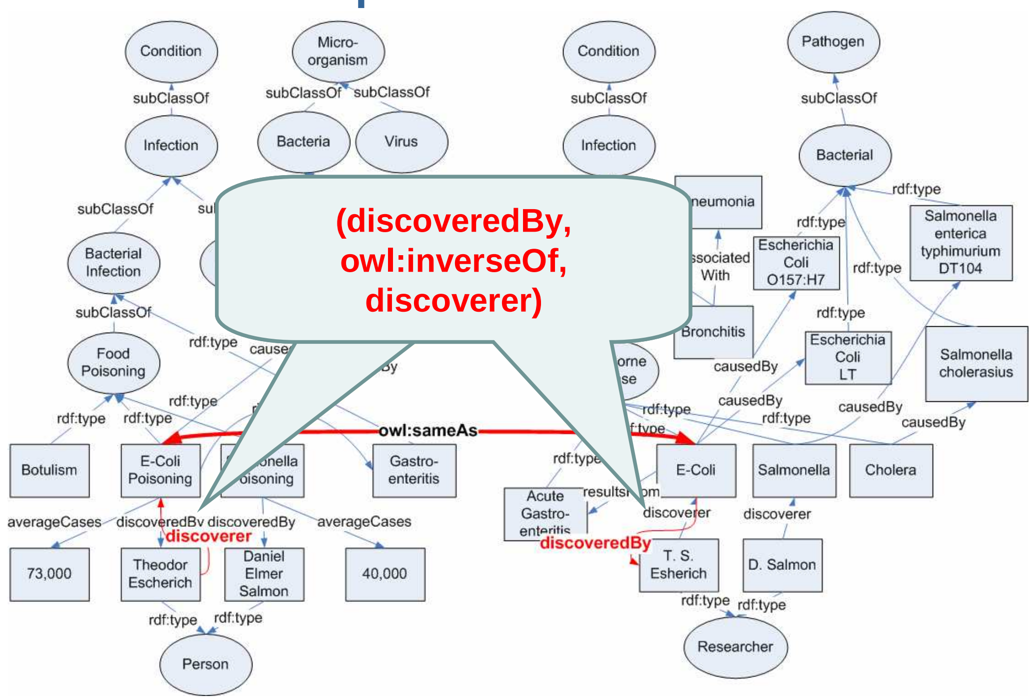
$$\frac{(o_1 \text{ owl:sameAs } o_2) \ (o_2 \text{ owl:differentFrom } o_1)}{\perp}$$

Example inference

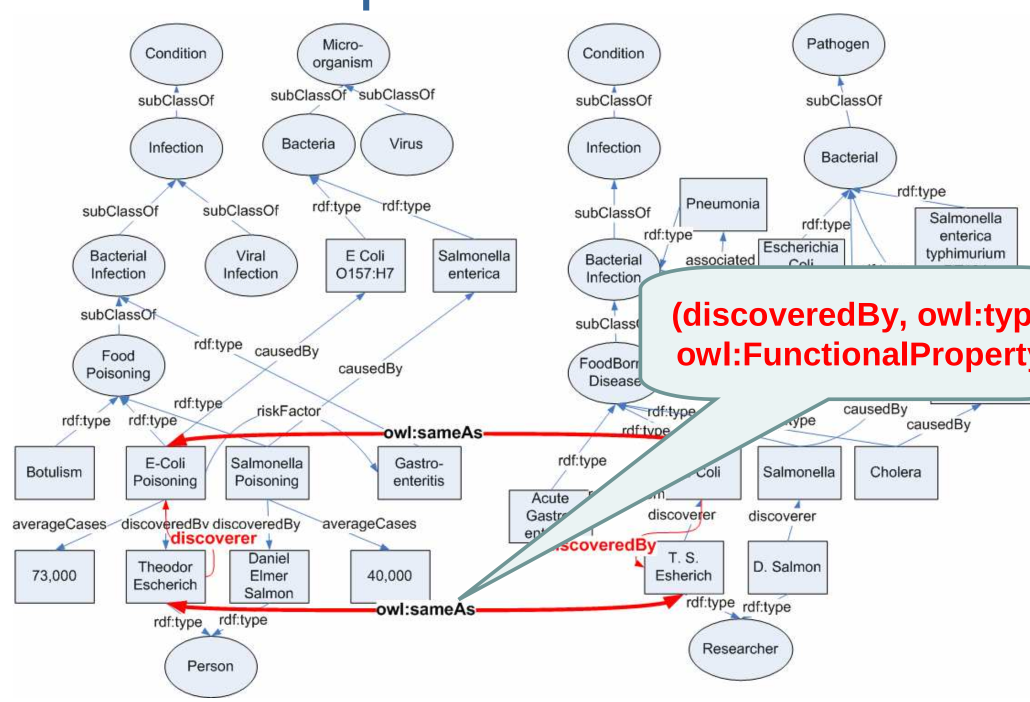


(discoveredBy, owl:inverseOf, discoverer); (discoveredBy, owl:type, owl:FunctionalProperty)
(discoveredBy, owl:inverseOf, discoverer); (associatedWith, owl:type, owl:TransitiveProperty)
(resultsF rom, rdfs:subPropertyOf, associatedWith)

Example inference



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**.

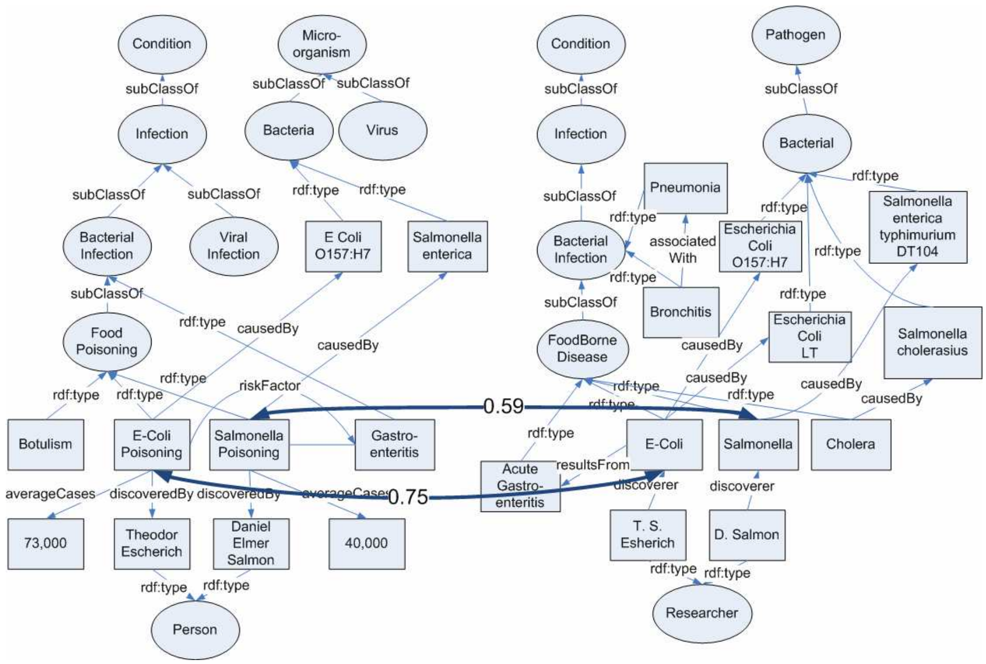
Ontology Alignment

- Motivation and goals
- Short overview of OWL Lite
- **The ILIADS method** Udrea, Getoor, Miller, SIGMOD07
- Experimental evaluation

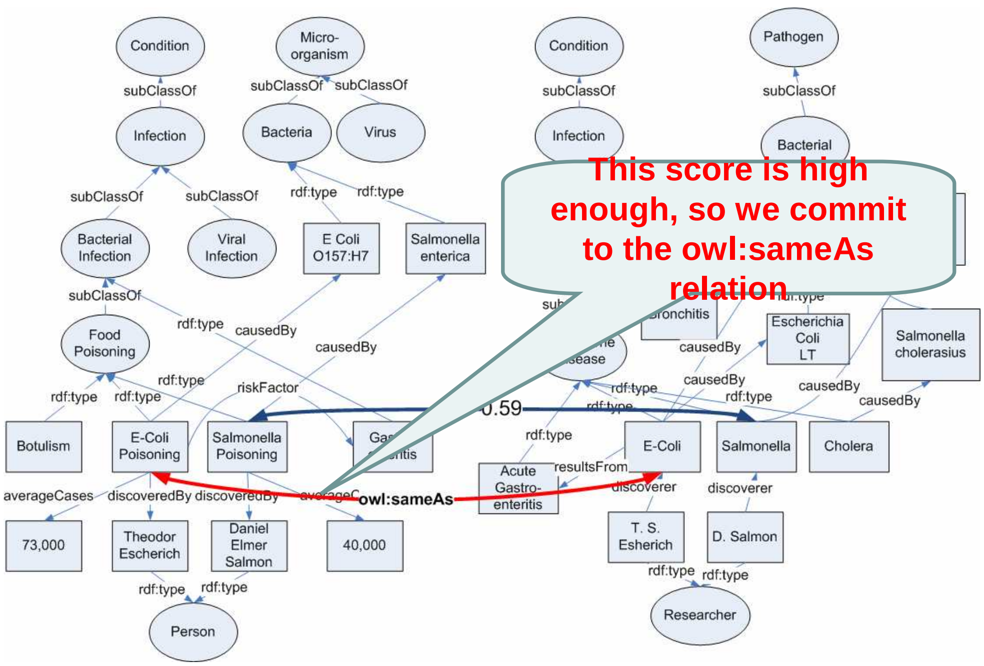
State of the art

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

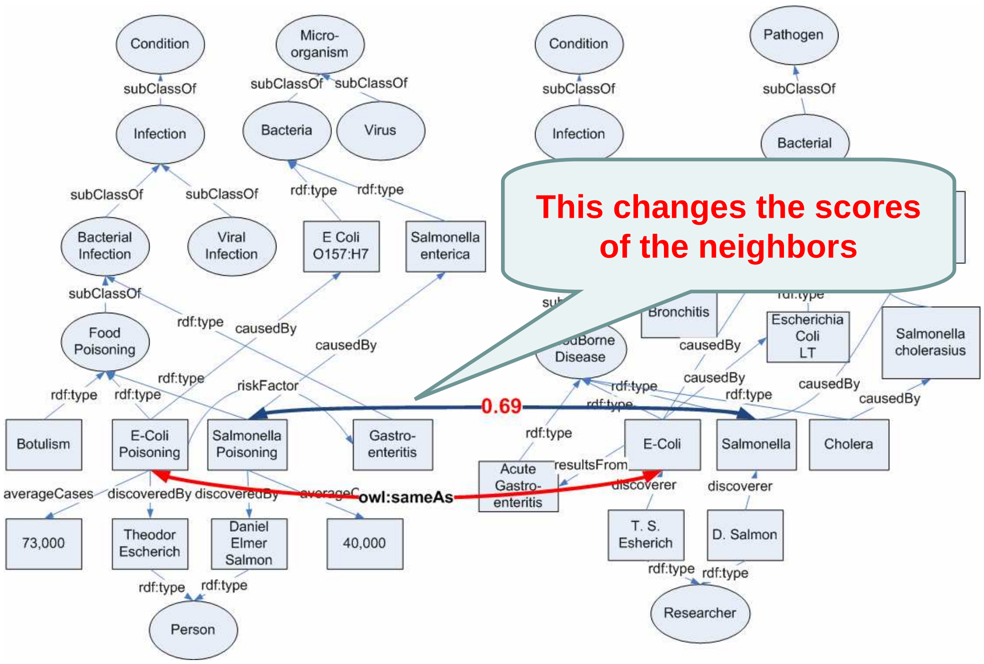
Incremental methods



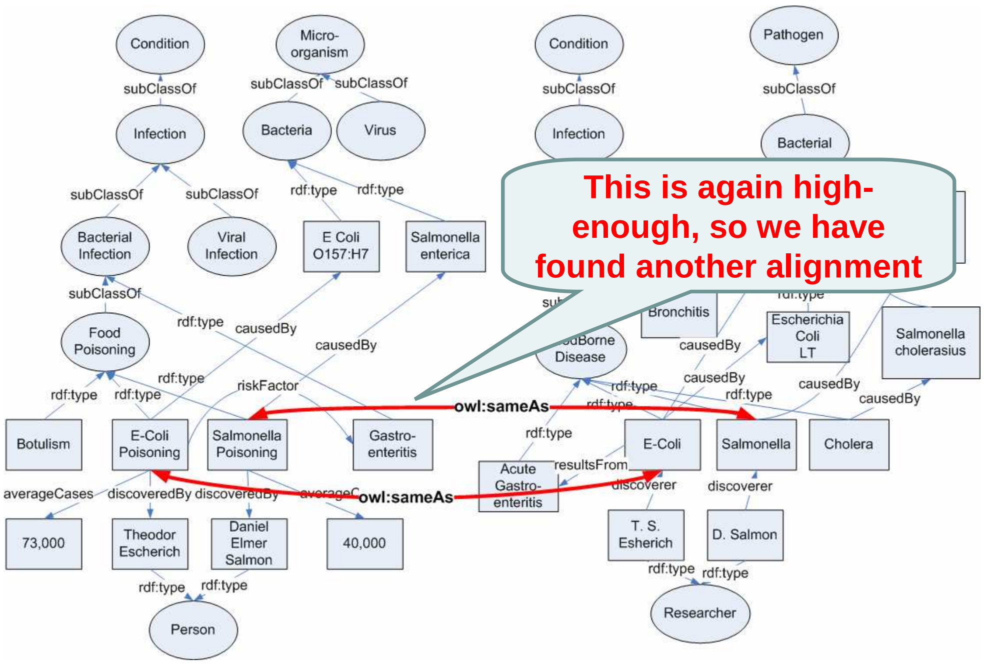
Incremental methods



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

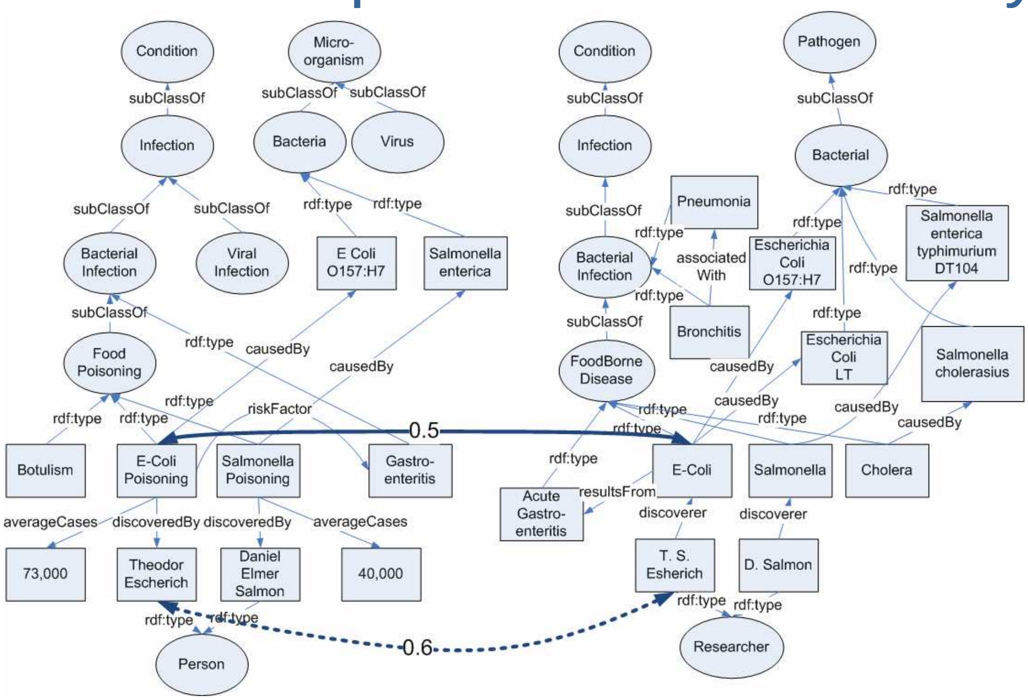
1. Compute local similarities
2. Select *promising* candidates
3. **For each** candidate
   a. Perform N inference steps
   b. **Update score with the inference similarity**
4. Select the candidate with the best score

end

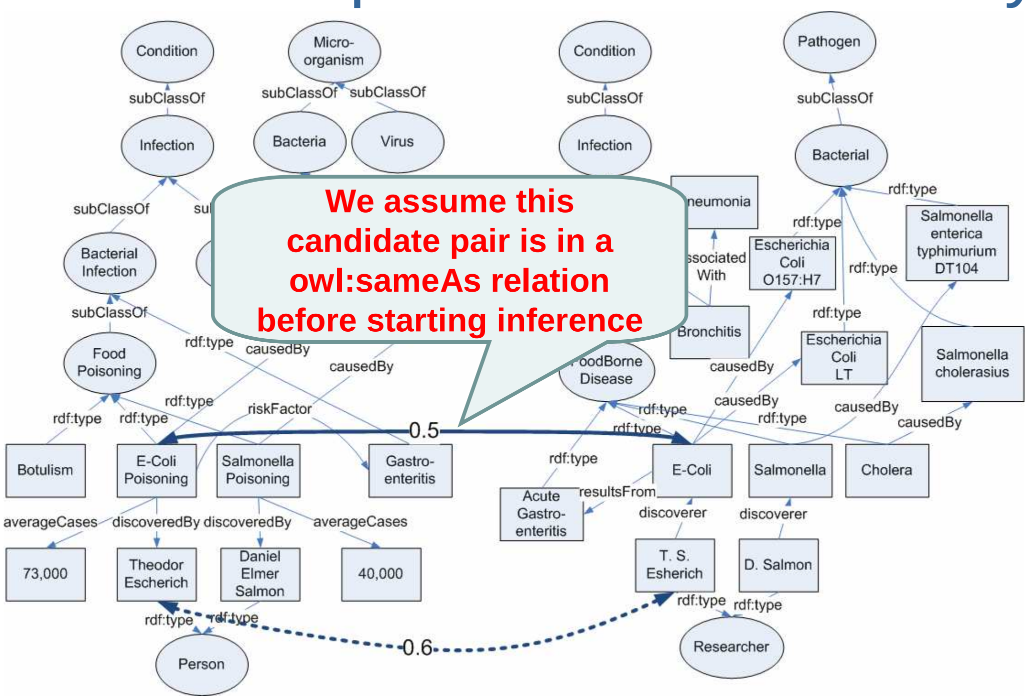
For the candidate pair (e,e'):

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

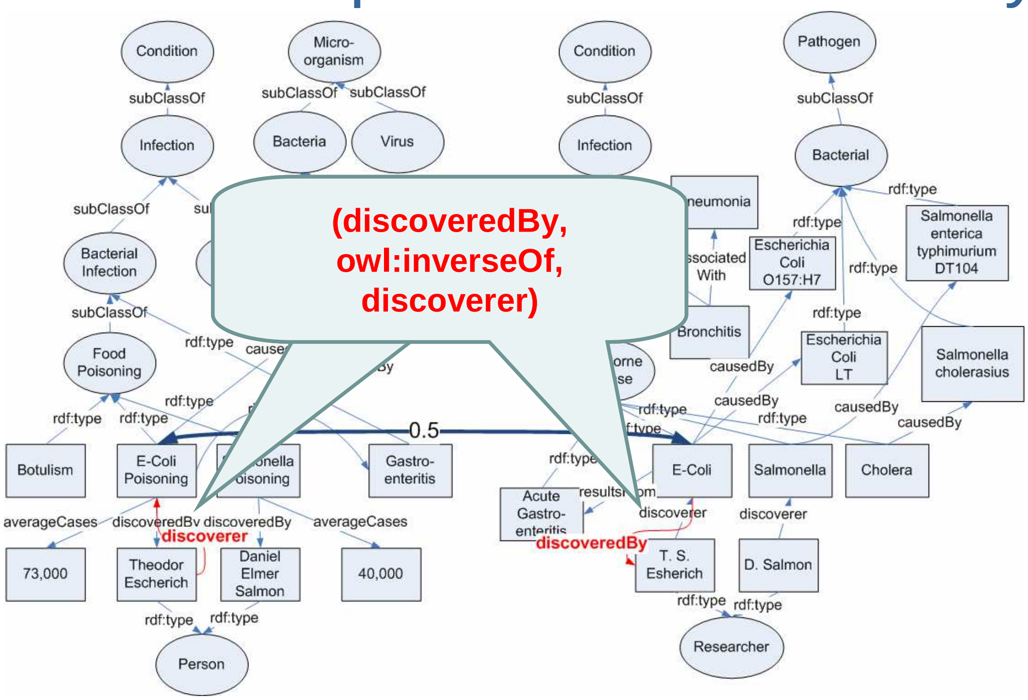
Example inference similarity



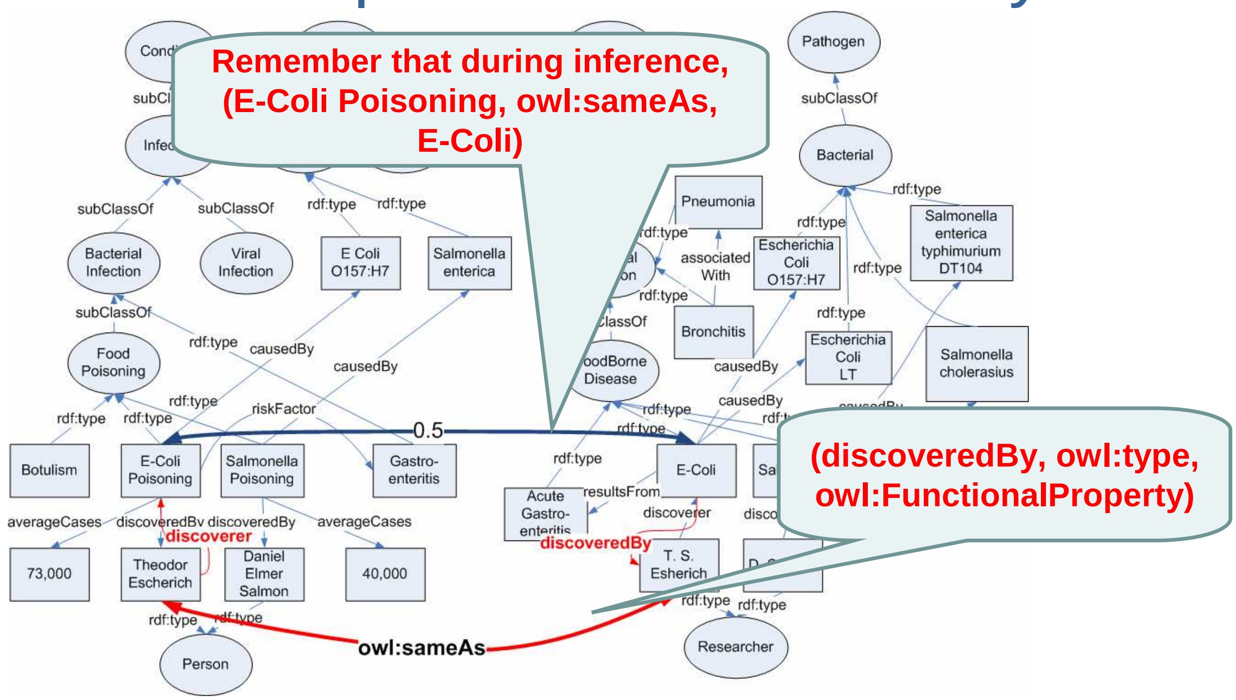
Example inference similarity



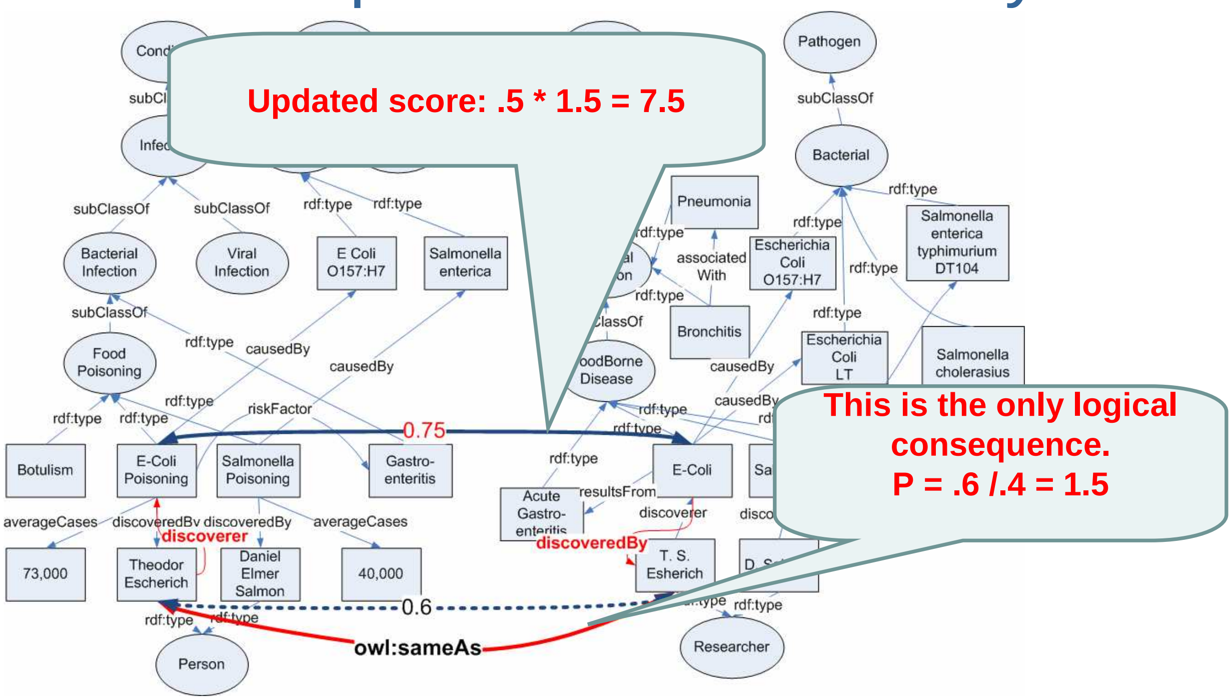
Example inference similarity



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

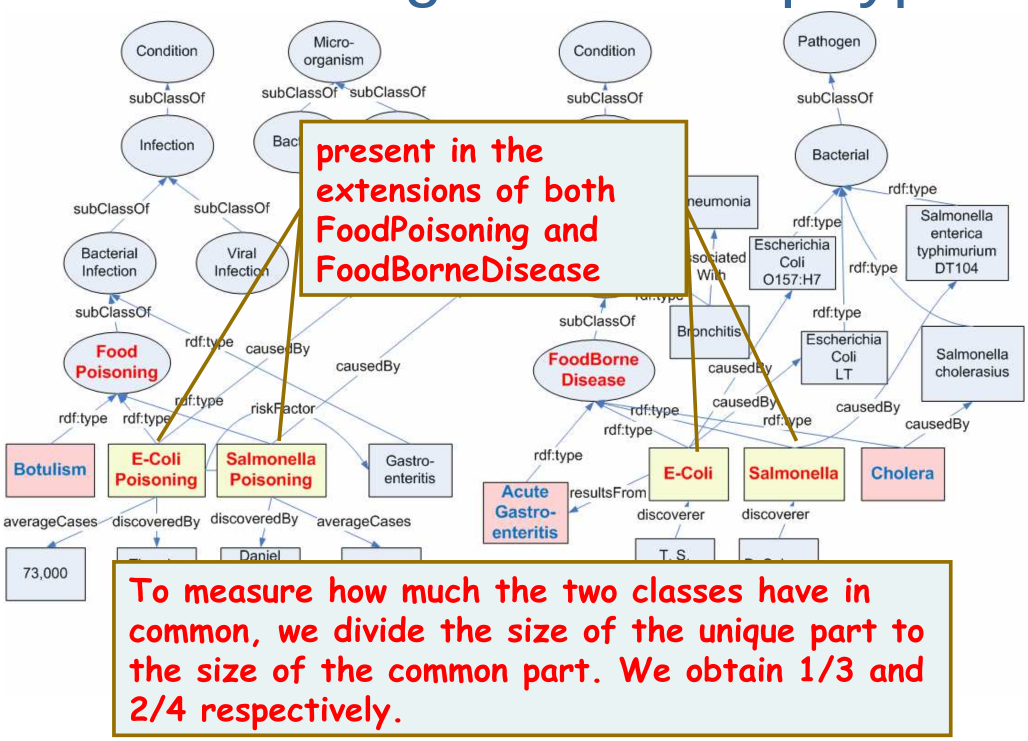
Other issues

- 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

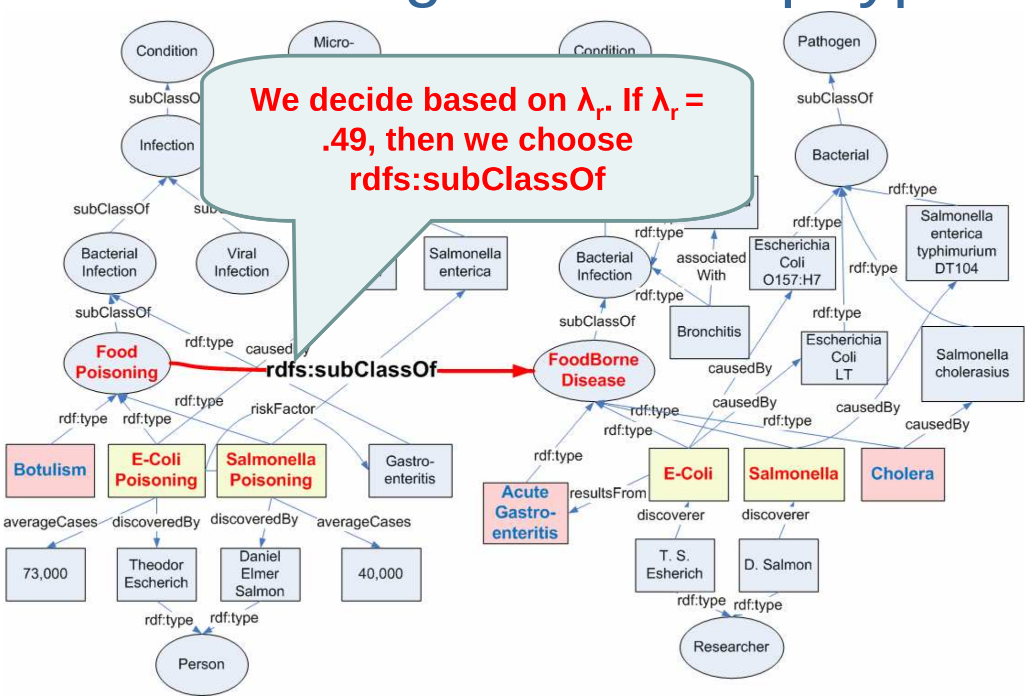
Deciding relationship type



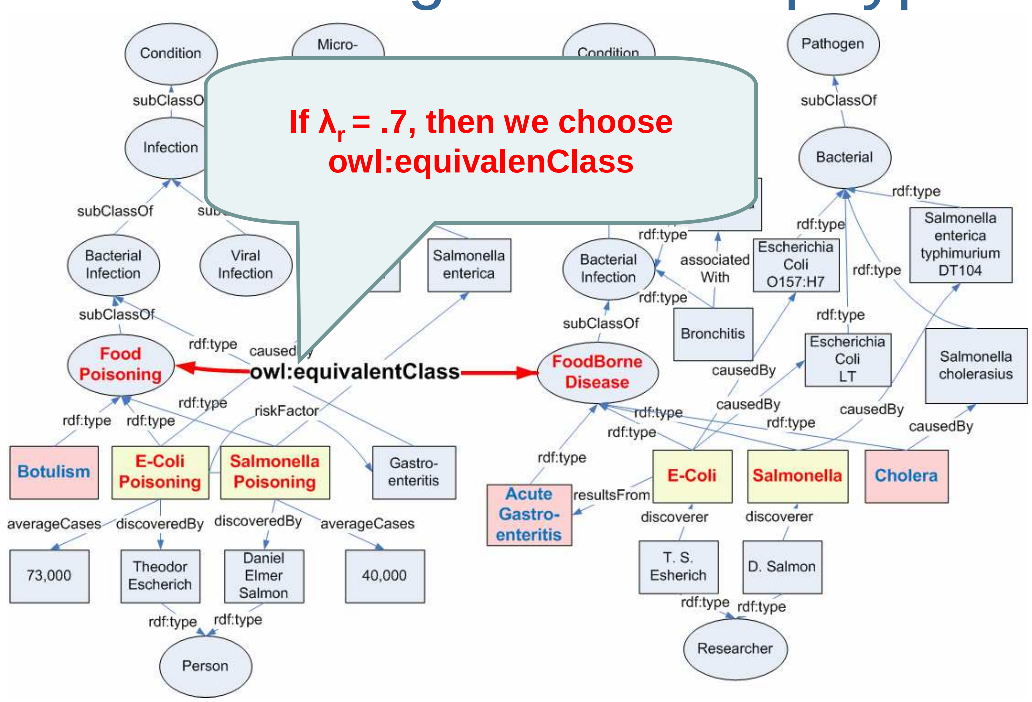
present in the extensions of both FoodPoisoning and FoodBorneDisease

To measure how much the two classes have in common, we divide the size of the unique part to the size of the common part. We obtain 1/3 and 2/4 respectively.

Deciding relationship type



Deciding relationship type



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)

Ontology Alignment

- Motivation and goals
- Short overview of OWL Lite
- The ILIADS method
- **Experimental evaluation**

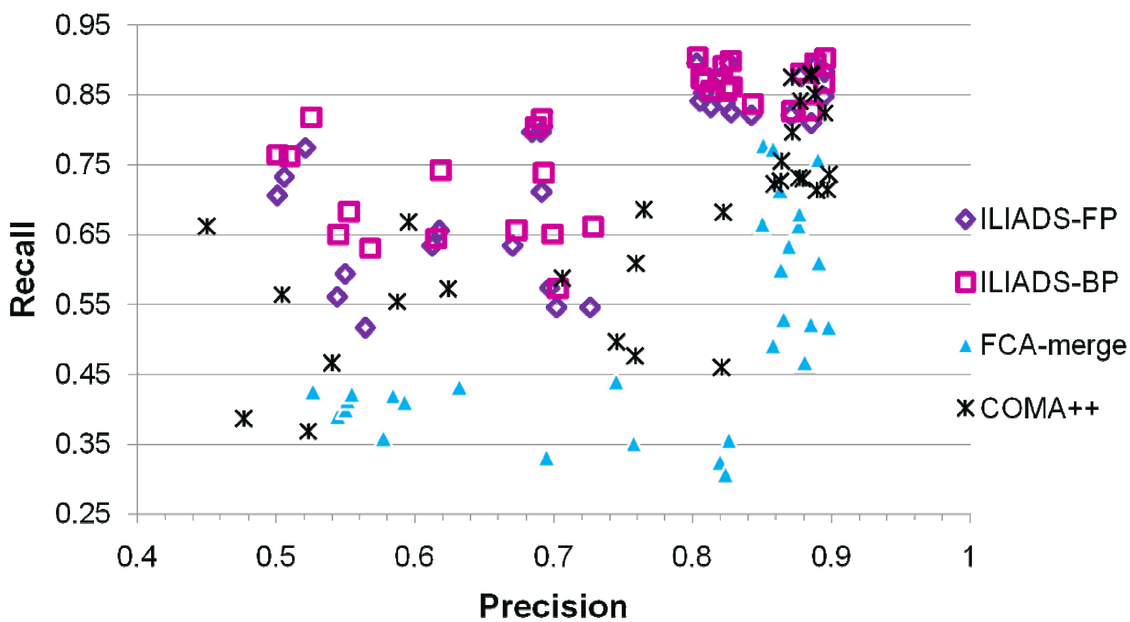
Experimental framework

- 30 pairs of ontologies
  - Ontologies from 194 to over 20000 triples
- Ground truth provided by human reviewers
- Comparison in terms of recall and precision with FCA-merge and COMA++
- Two versions of the algorithm
  - Best overall average quality ILIADS – FP
  - Best parameters for each pair ILIADS – BP

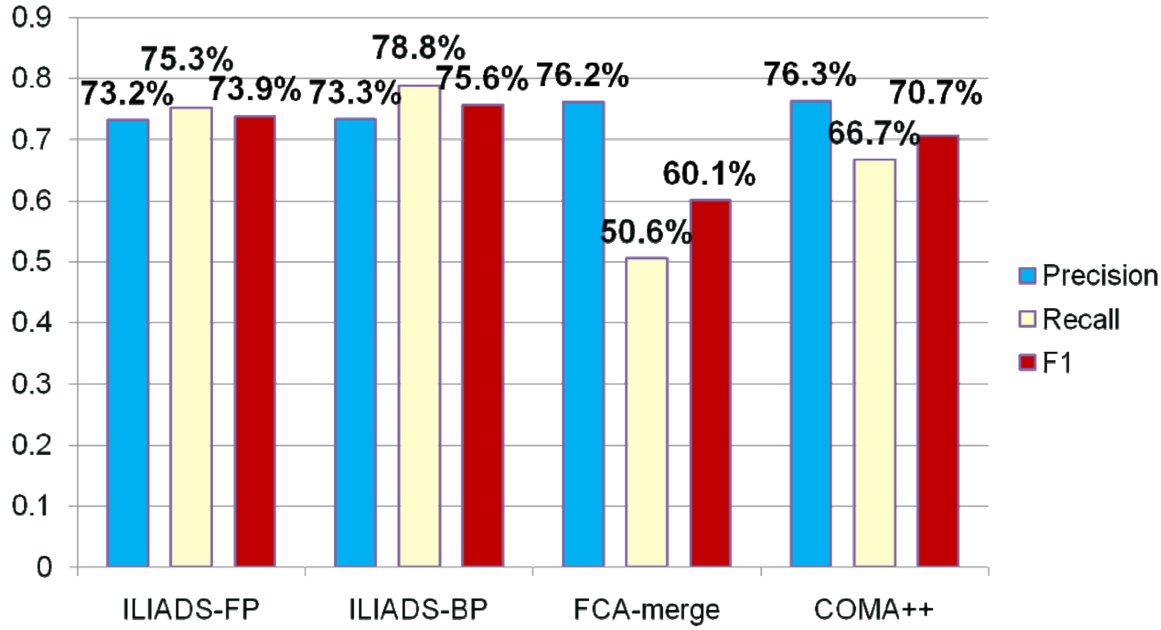
ILIADS-BP parameter setting

	λ_x^c	λ_x^i	λ_x^p	λ_s^c	λ_s^i	λ_s^p	λ_e^c	λ_e^p	λ_t	λ_r
Avg. F1	.2	.4	.1	.5	.6	.4	.3	.5	.7	.2
Min	.15	.4	0	.3	.45	.35	.2	.35	.65	.2
Max	.25	.45	.1	.65	.7	.5	.35	.65	.7	.2

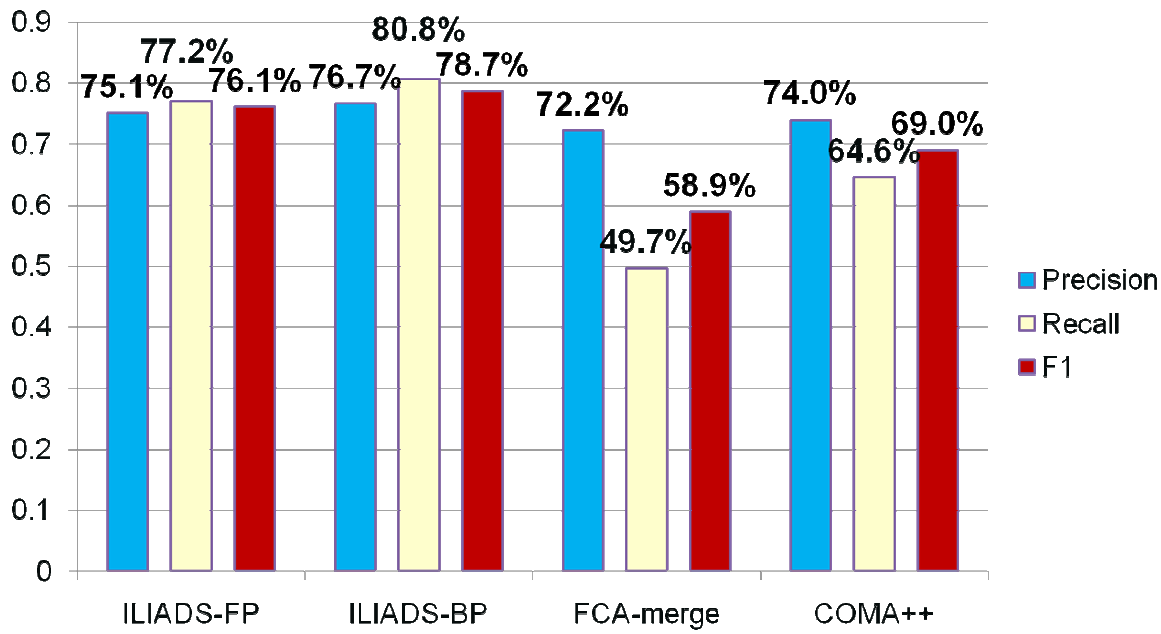
Precision/recall



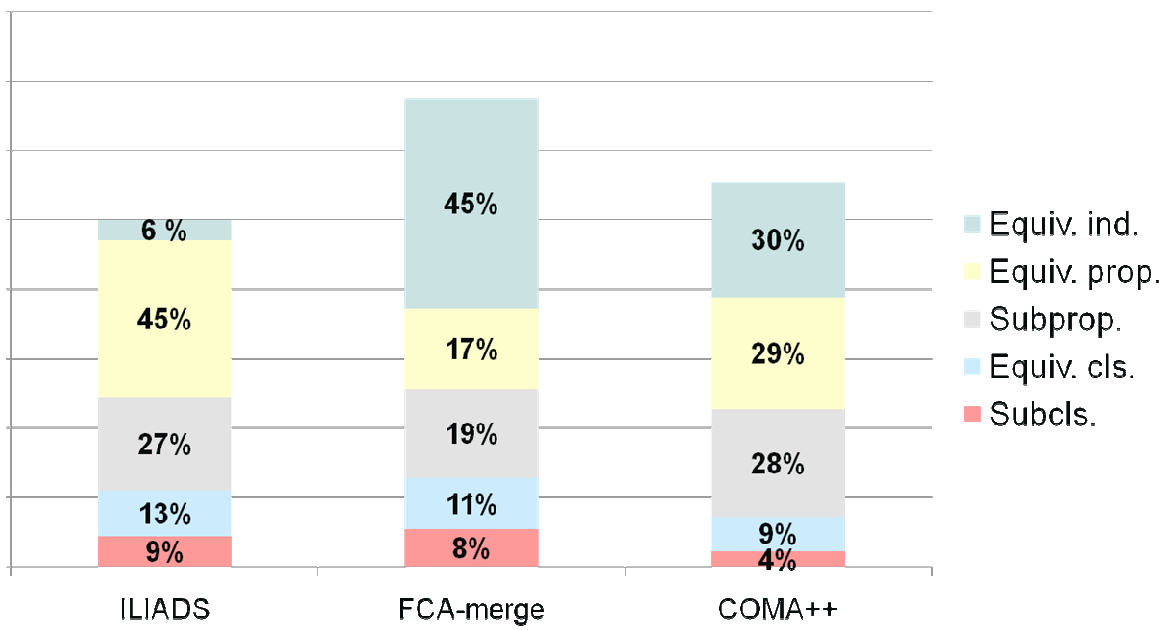
Precision/recall comparison



Precision/recall for ontologies with substantial instance data

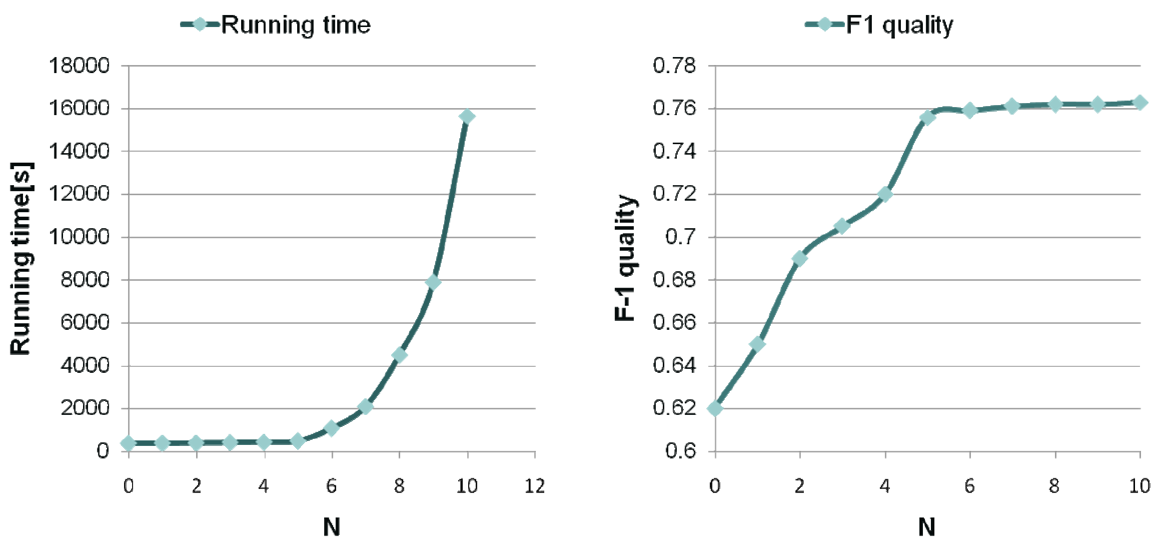


False negative analysis

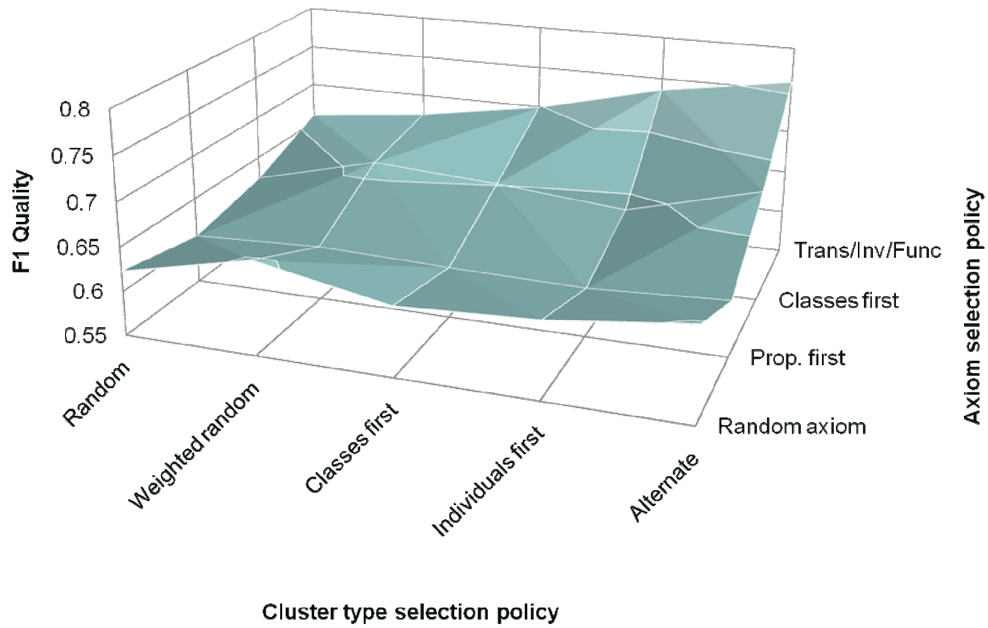


Number of inference steps

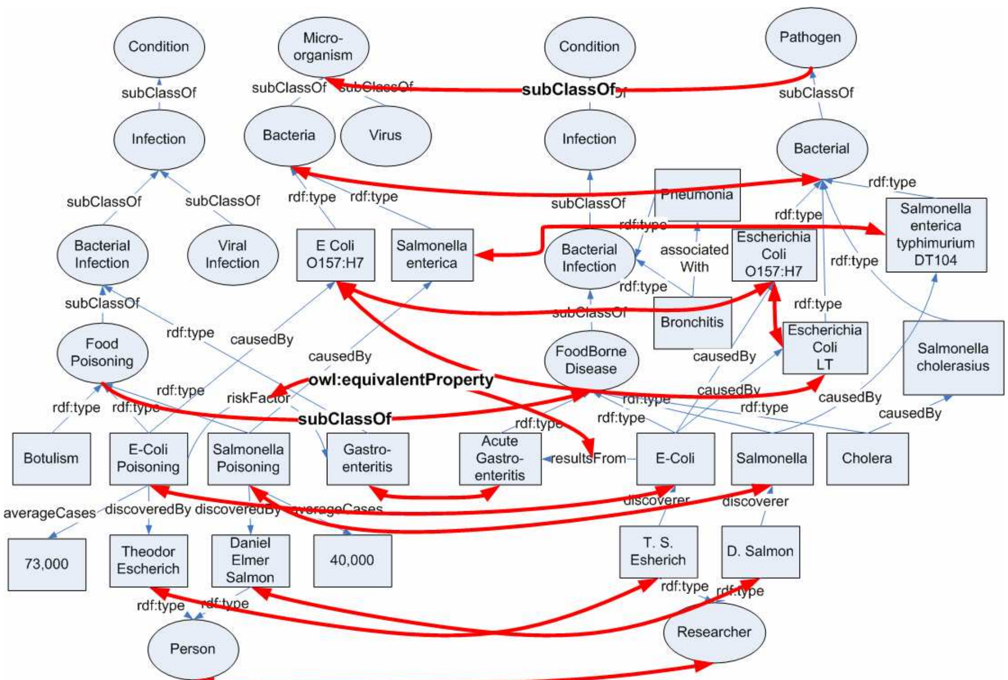
The number of 5 inference steps was chosen as the best compromise between:



Cluster type/axiom selection policies



And the result is...



(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

Learning and Inference **Hard**

- Full Joint Probabilistic Representations
  - Directed vs. Undirected
  - Require sophisticated approximate inference algorithms
  - Tradeoff: hard inference vs. hard learning
- Combinations of Local Classifiers
  - Local classifiers choices
  - Require sophisticated updating and truth maintenance or global optimization via LP
  - Tradeoff: granularity vs. complexity

Many interesting and challenging research problems!!

Roadmap

- The Problem
- The Components
- Putting It All Together
- **Open Questions**

1. Query-time GIA

- Instead of viewing as an off-line knowledge reformulation process

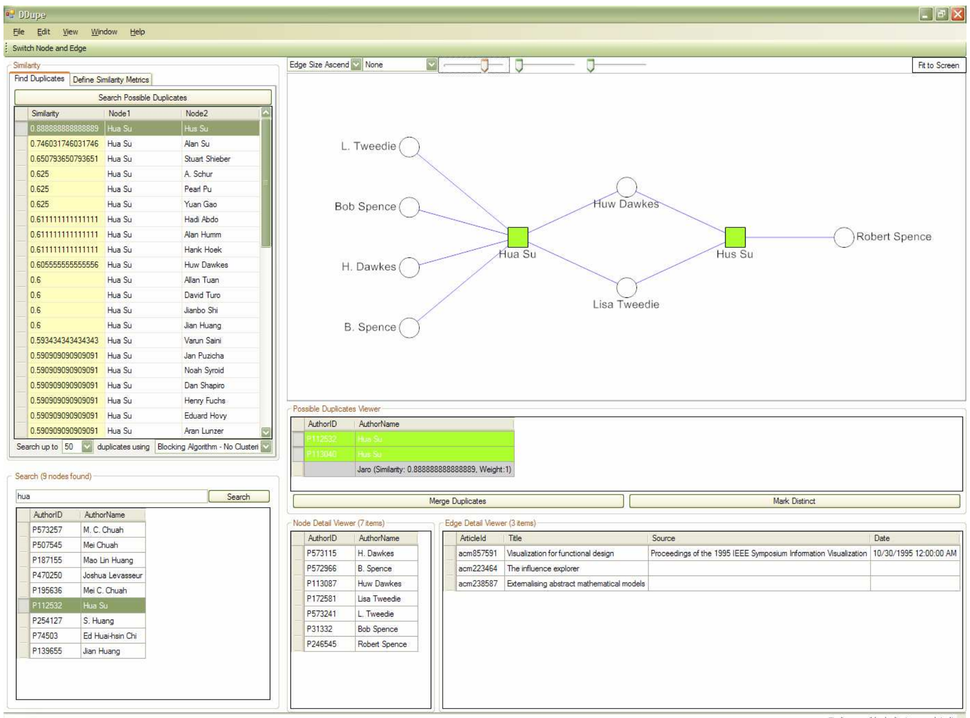
- consider as real-time data gathering with
  - varying resource constraints
  - ability to reason about value of information
  - e.g., what attributes are most useful to acquire? Which relationships? Which will lead to the greatest reduction in ambiguity?

- Bhattacharya & Getoor, *Query-time Entity Resolution,* JAIR 2007.

2. Visual Analytics for GIA

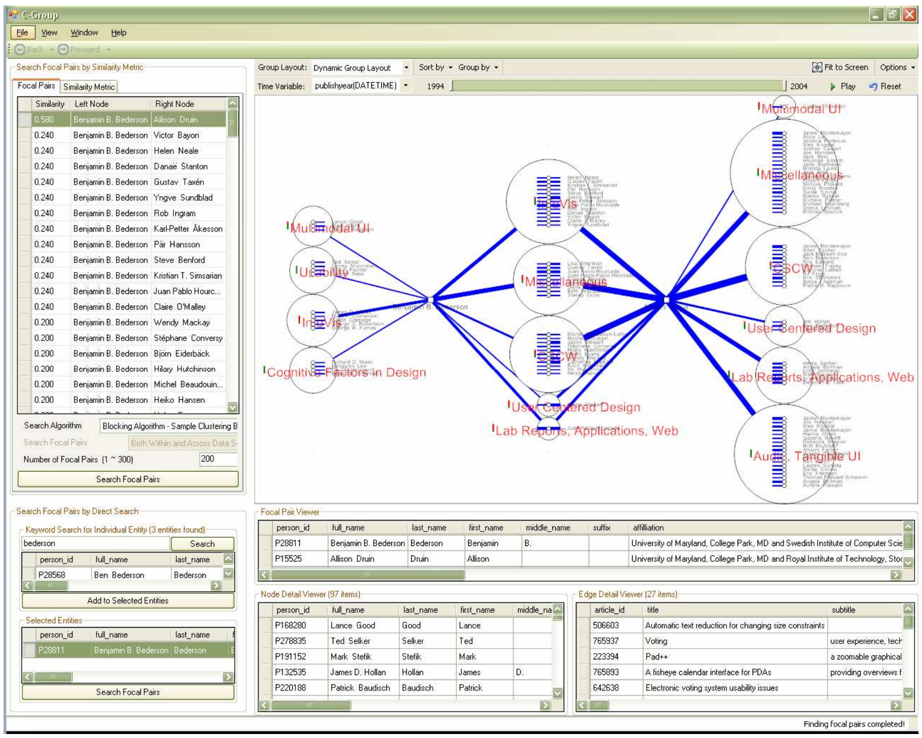
- Combining rich statistical inference models with visual interfaces that support knowledge discovery and understanding
- Because the statistical confidence we may have in any of our inferences may be low, it is important to be able to have a human in the loop, to understand and validate results, and to provide feedback.
- Especially for graph and network data, a well-chosen visual representation, suited to the inference task at hand, can improve the accuracy and confidence of user input

D-Dupe: An Interactive Tool for Entity Resolution



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

C-Group: A Visual Analytic Tool for Pairwise Analysis of Dynamic Group Membership



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

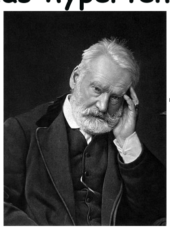
3. GI & Privacy

- Obvious privacy concerns that need to be taken into account!!!
- A better theoretical understanding of when graph identification is feasible will also help us understand what must be done to maintain privacy of graph data
- … Graph Re-Identification: study of anonymization strategies such that the information graph **cannot** be inferred from released data graph

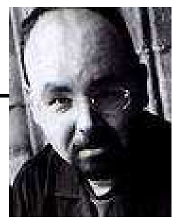
Link Re-Identification

Disease data

has hypertension



father-of



Communication data

?

call

Robert Lady

Search data

Query 1:

"how to tell if your wife is cheating on you"

same-user

Query 2:

"myrtle beach golf course job listings"

Social network data

friends

Zheleva and Getoor, Preserving the Privacy of Sensitive Relationshops in Graph Data, PINKDD 2007

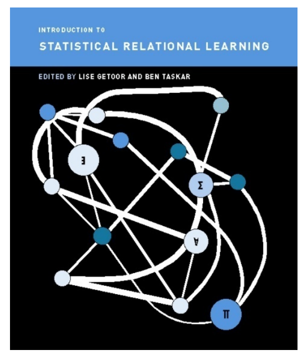
Summary: GIA & D/MD Alignment

- Graph Identification and alignment can be seen as a process of **knowledge reformulation**
- In the context where we have some statistical information to help us **learn** which reformulations are more promising than others
- **Inference** is the process of transferring the learned knowledge to new situations

Statistical Relational Learning (SRL)

- Methods that combine expressive knowledge representation formalisms such as relational and first-order logic with principled probabilistic and statistical approaches to inference and learning

Dagstuhl April 2007



- Hendrik Blockeel, Mark Craven, James Cussens, Bruce D'Ambrosio, Luc De Raedt, Tom Dietterich, Pedro Domingos, Saso Dzeroski, Peter Flach, Rob Holte, Manfred Jaeger, David Jensen, Kristian Kersting, Heikki Mannila, Andrew McCallum, Tom Mitchell, Ray Mooney, Stephen Muggleton, Kevin Murphy, Jen Neville, David Page, Avi Pfeffer, Claudia Perlich, David Poole, Foster Provost, Dan Roth, Stuart Russell, Taisuke Sato, Jude Shavlik, Ben Taskar, Lyle Ungar and many others

Conclusion

- Relationships matter!
- Structure matters!

- Killer Apps:
  - Computer Vision: Human Activity Recognition
  - Information Extraction: Entity Extraction & Role labeling
  - Data Integration: Ontology Alignment
  - Personal Information Management: Intelligent Desktop

- While there are important pitfalls to take into account (confidence and privacy), there are **many potential benefits and payoffs!**

Thanks!

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

Work sponsored by the National Science Foundation,
KDD program, National Geospatial Agency, Google and Microsoft