Link Mining & Entity Resolution

Lise Getoor

University of Maryland, College Park

Learning in Structured Domains

- Traditional machine learning and data mining approaches assume:
  - A random sample of homogeneous objects from single relation

- Real-world datasets:
  - Multi-relational, heterogeneous and semi-structured
    - represented as a graph or network
      - Nodes are objects
        - » May have different kinds of objects
        - » Objects have attributes
        - » Objects may have labels or classes
      - Edges are links
        - » May have different kinds of links
        - » Links may have attributes
        - » Links may be directed, are not required to be binary

- Statistical Relational Learning:
  - newly emerging research area at the intersection of research in social network and link analysis, hypertext and web mining, natural language processing, graph mining, relational learning and ILP.

- Sample Domains:
  - web data, bibliographic data, epidemiological data, communication data, customer networks, collaborative filtering, trust networks, biological data

Link Mining Tasks & Challenges

Object-Related Tasks

- Link-based Classification
- Link-based Ranking
- Group Detection
- Entity Resolution

Link-Related Tasks

- Link Type Prediction
- Predicting Link Existence
- Link Cardinality Estimation
- Predicate Invention

Graph-Related Tasks

- Subgraph Discovery
- Graph Classification
- Generative Models
- Meta-data Discovery

- **Challenges**
  - Modeling Logical vs. Statistical dependencies
  - Feature construction
  - Instances vs. Classes
  - Collective Classification
  - Collective Consolidation
  - Effective Use of Labeled & Unlabeled Data
  - Link Prediction
  - Closed vs. Open World

Reference: SIGKDD Explorations Special Issue on Link Mining, December 2005, edited with Chris Diehl from Johns Hopkins Applied Physics Lab

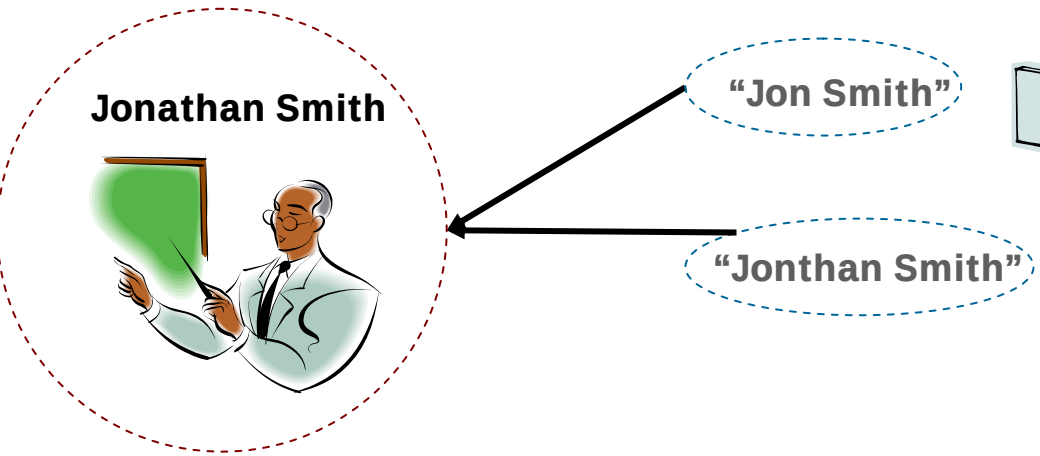
LINQs Group @ UMD

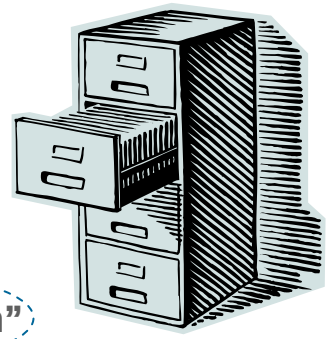
- Members
  - myself, Indrajit Bhattacharya, Mustafa Bilgic, Rezarta Islamaj, Louis Licamele, Galileo Namata, John Park, Prithivaraj Sen, Vivek Senghal
- Projects
  - Link-based Classification
  - Entity Resolution (ER)
    - Algorithms
    - Query-time ER
    - User Interface
  - Predictive Models for Social Network Analysis
    - Affiliation Networks
    - Social Capital in Friendship Event Networks
  - Temporal Analysis of Email Traffic Networks
  - Feature Generation for Sequences (biological data)

Entity Resolution

- The Problem
- Relational Entity Resolution
- Algorithms
  - Graph-based Clustering (GBC)
  - Probabilistic Model (LDA-ER)
- Query-time Entity Resolution
- ER User Interface

The Entity Resolution Problem





Issues:

1. Identification
2. Disambiguation

The Entity Resolution Problem

John Smith

"John Smith"

"J Smith"

James Smith

"Jim Smith"

"James Smith"

"J Smith"

Jonathan Smith

"Jon Smith"

"Jonthan Smith"

Unsupervised clustering approach

- Number of clusters/entities unknown apriori

Attribute-based Entity Resolution

Pair-wise classification

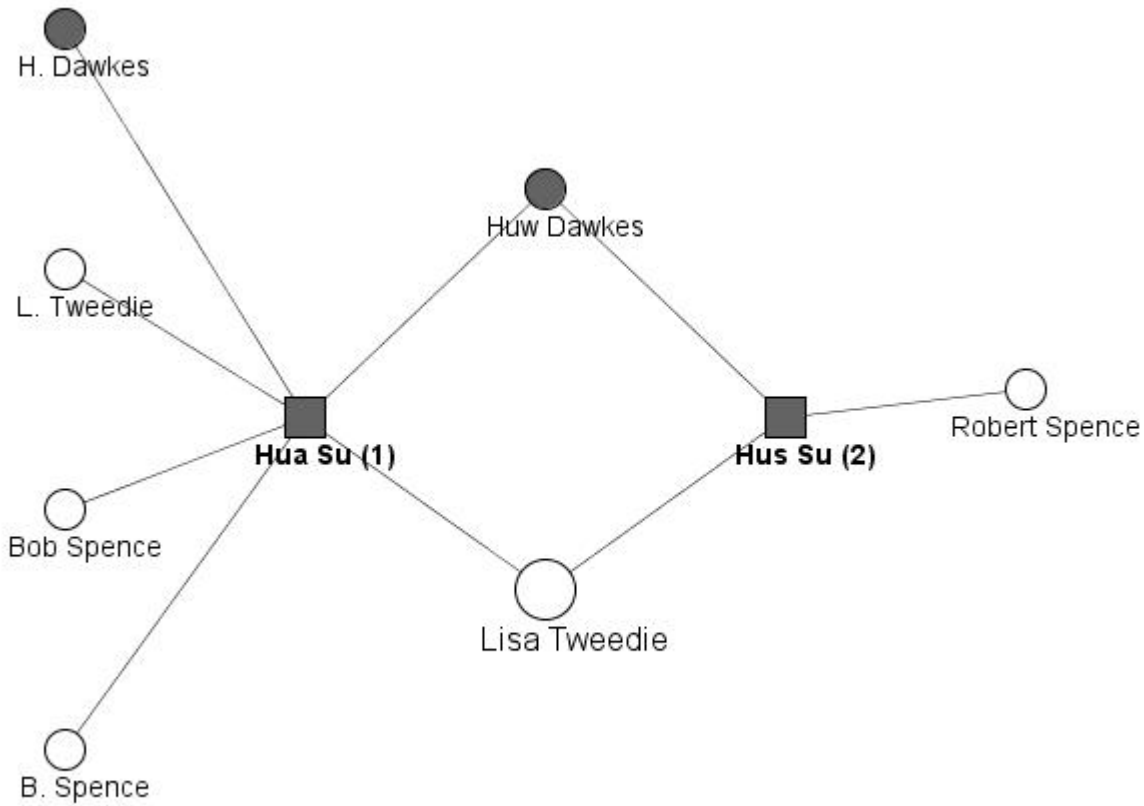
"J Smith"	"James Smith"	?
"Jim Smith"	"James Smith"	0.8
"J Smith"	"James Smith"	?
"John Smith"	"James Smith"	0.1
"Jon Smith"	"James Smith"	0.7
"Jonthan Smith"	"James Smith"	0.05

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

Relational Entity Resolution

- References not always observed independently
  - Links between references indicate relations between the entities
  - Co-author relations for bibliographic data
- Use relations to improve disambiguation and identification

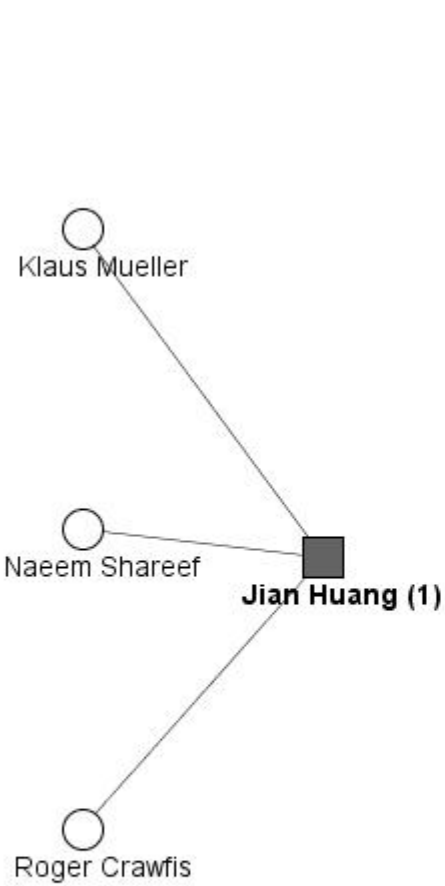
Relational Identification

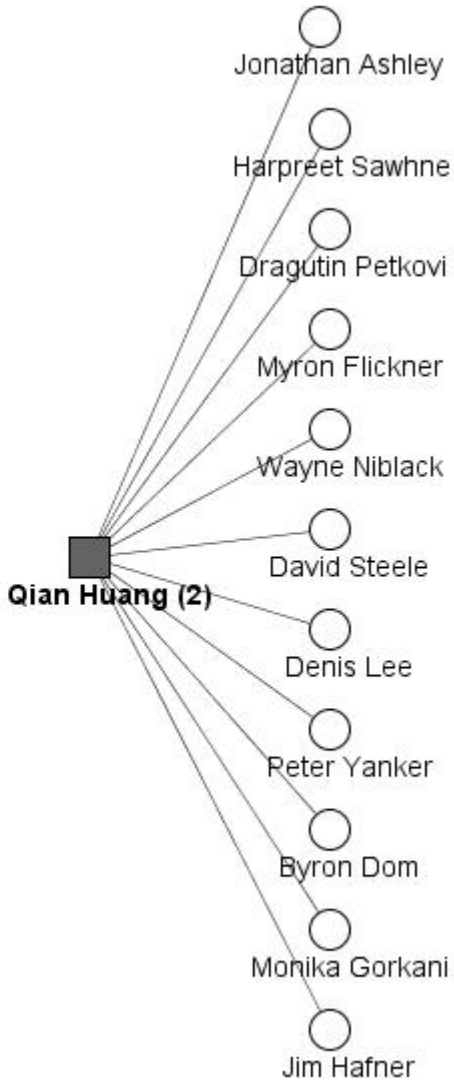


Very similar names.

Added evidence from shared co-authors

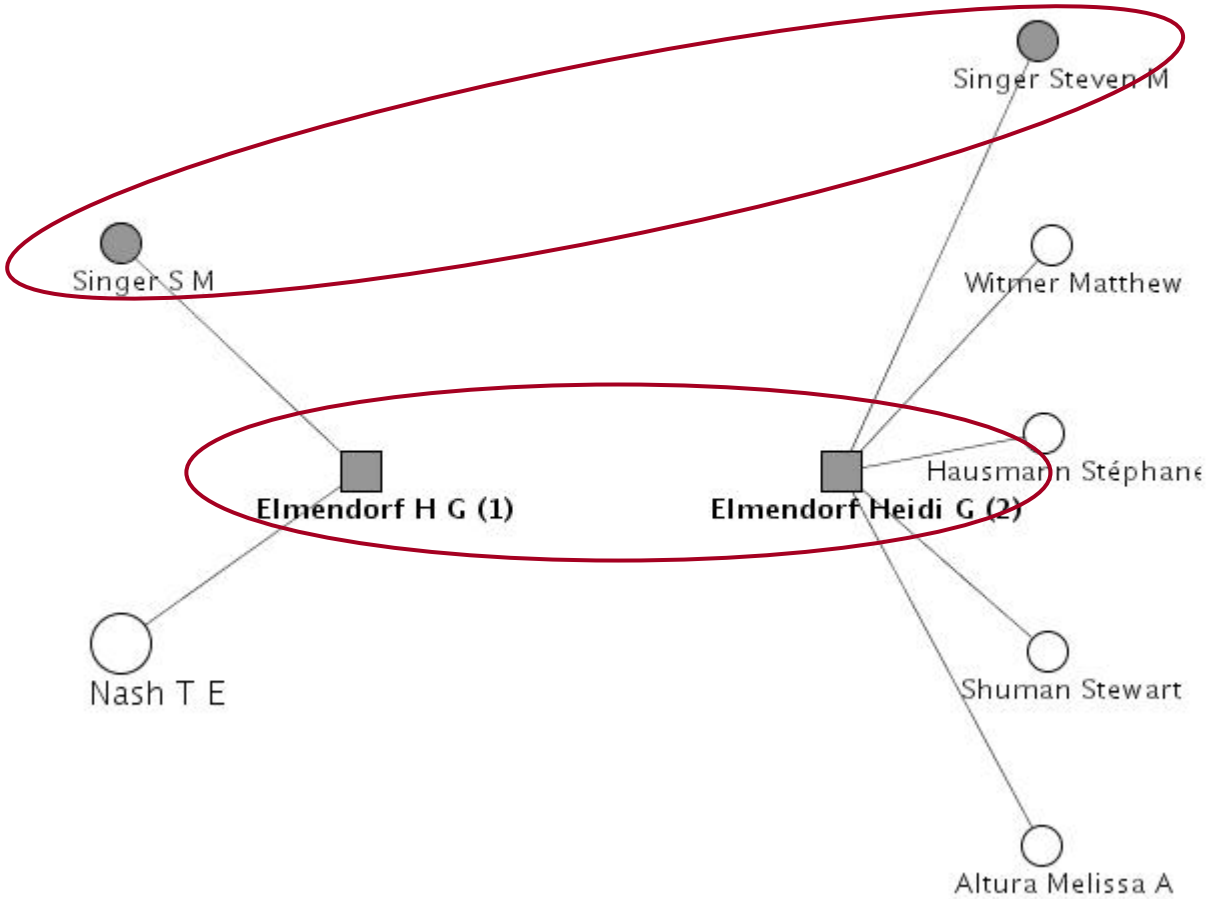
Relational Disambiguation





Very similar names but no shared collaborators

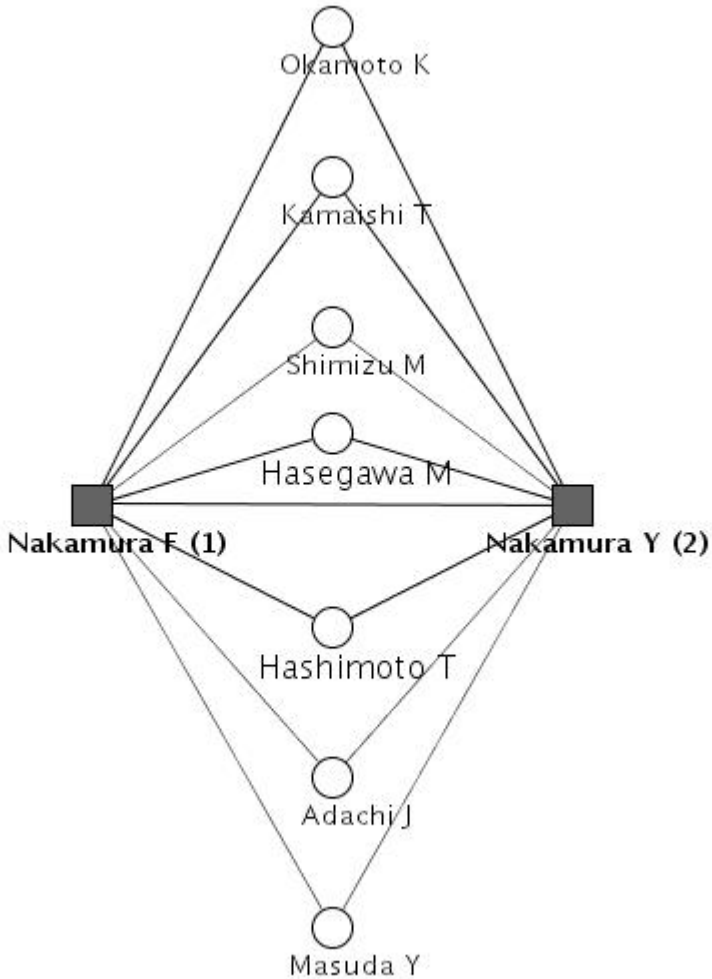
Collective Entity Resolution Using Relations



One resolutions provides evidence for another => joint resolution

Relational Constraints For Resolution

Co-authors are typically distinct



Entity Resolution

- The Problem
- Relational Entity Resolution
- Algorithms
  - Graph-based Clustering (GBC-ER)
  - Probabilistic Model (LDA-ER)
- Query-time Entity Resolution
- ER User Interface

Example: Bibliographic Entity Resolution

- Resolve author, paper, venue, publisher entities from citation strings

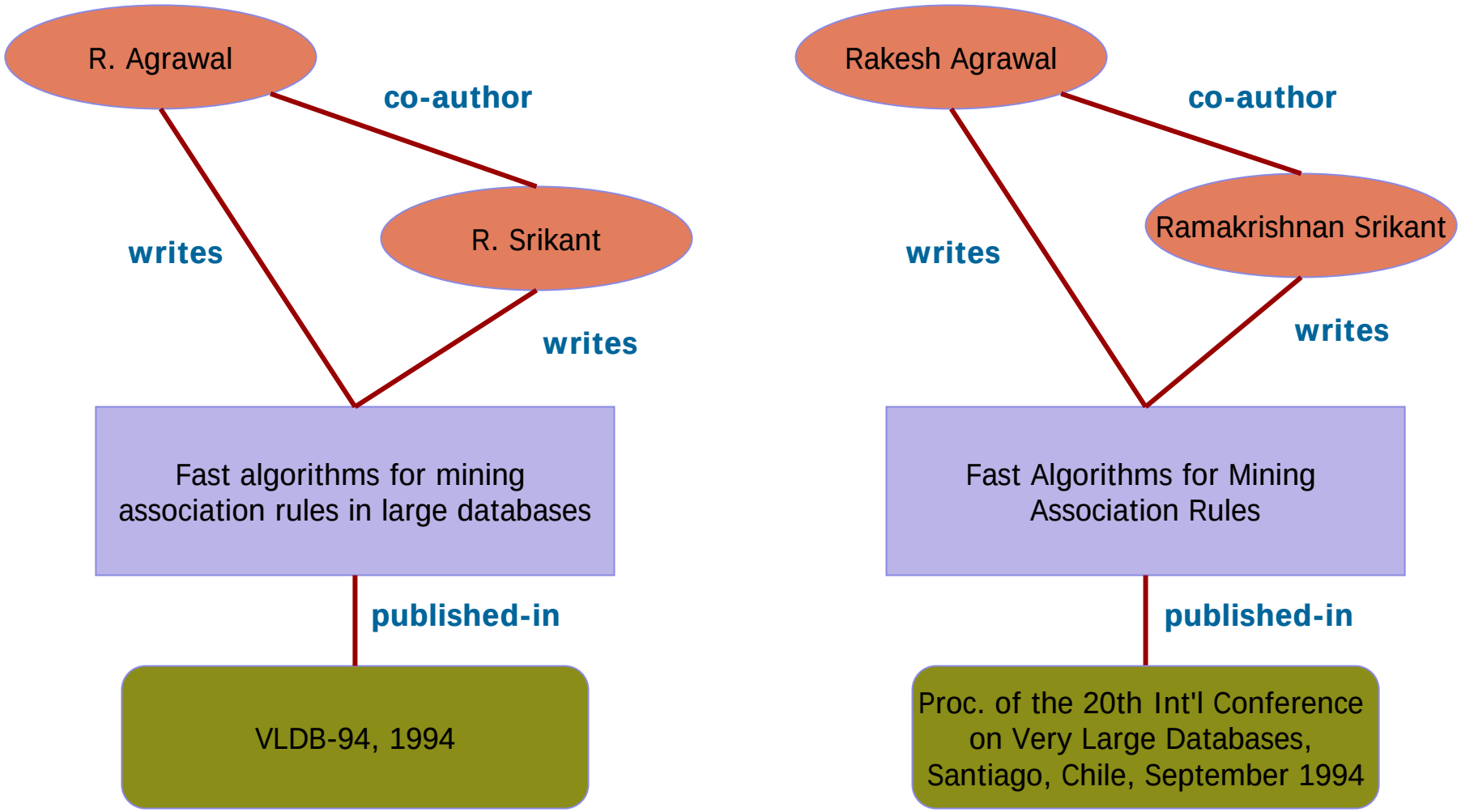
  - R. Agrawal, R. Srikant. Fast algorithms for mining association rules in large databases. In VLDB-94, 1994.
  - Rakesh Agrawal and Ramakrishnan Srikant. Fast Algorithms for Mining Association Rules. In Proc. of the 20th Int'l Conference on Very Large Databases, Santiago, Chile, September 1994.

Exploiting Bibliographic Links

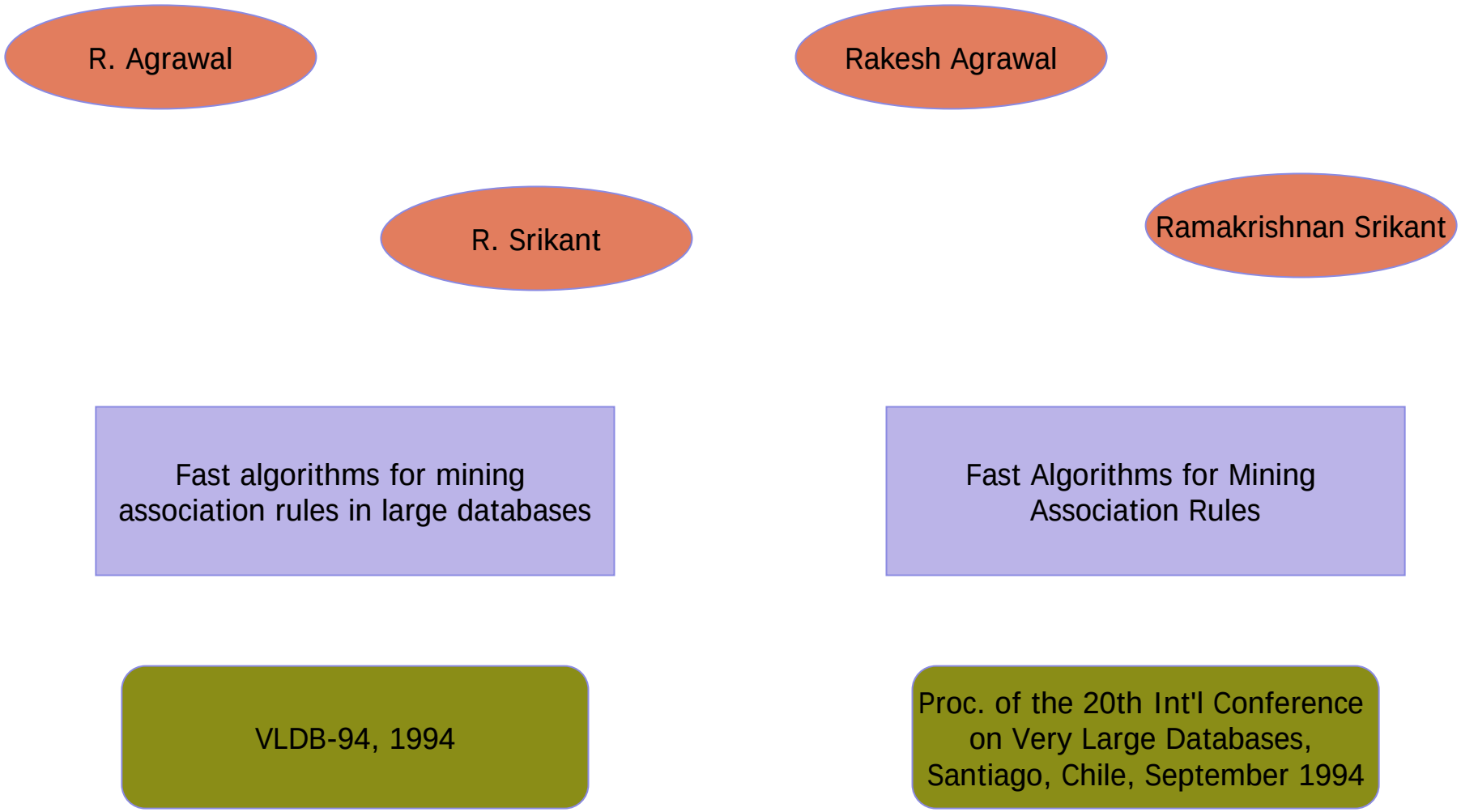
- Resolve author, paper, venue, publisher entities from citation strings

  - 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.

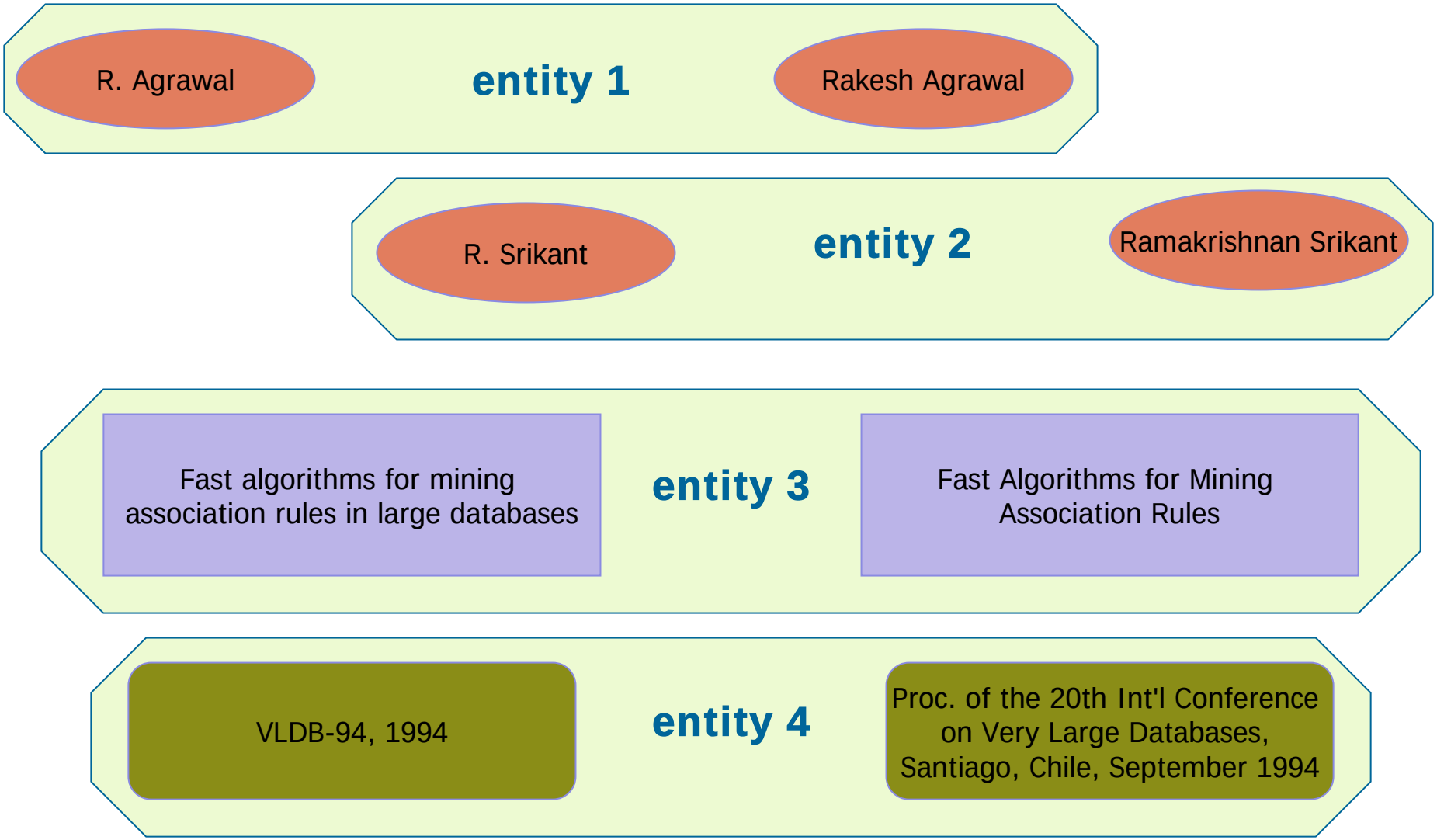
Exploiting Bibliographic Links



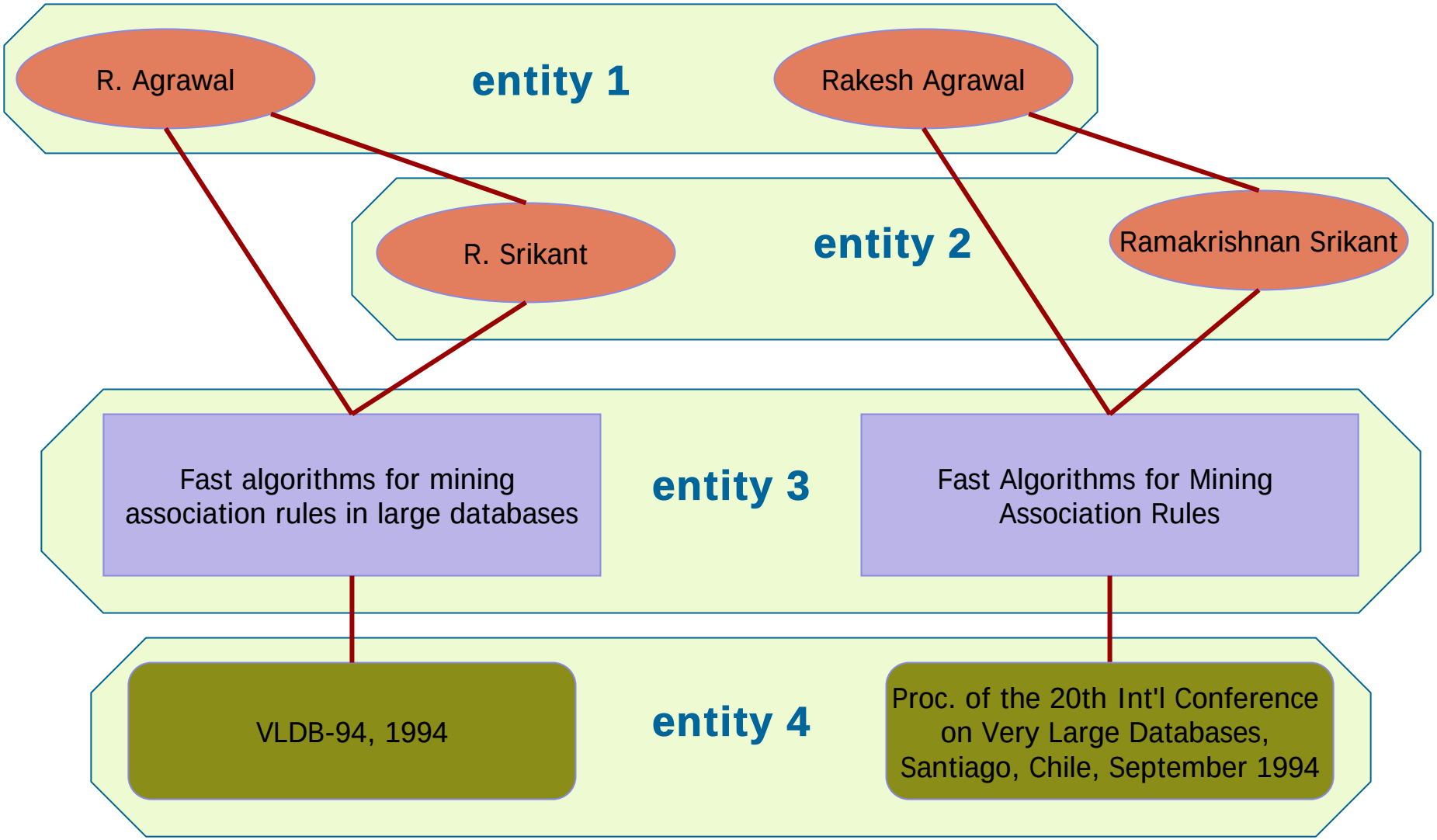
Exploiting Bibliographic Links



Exploiting Bibliographic Links

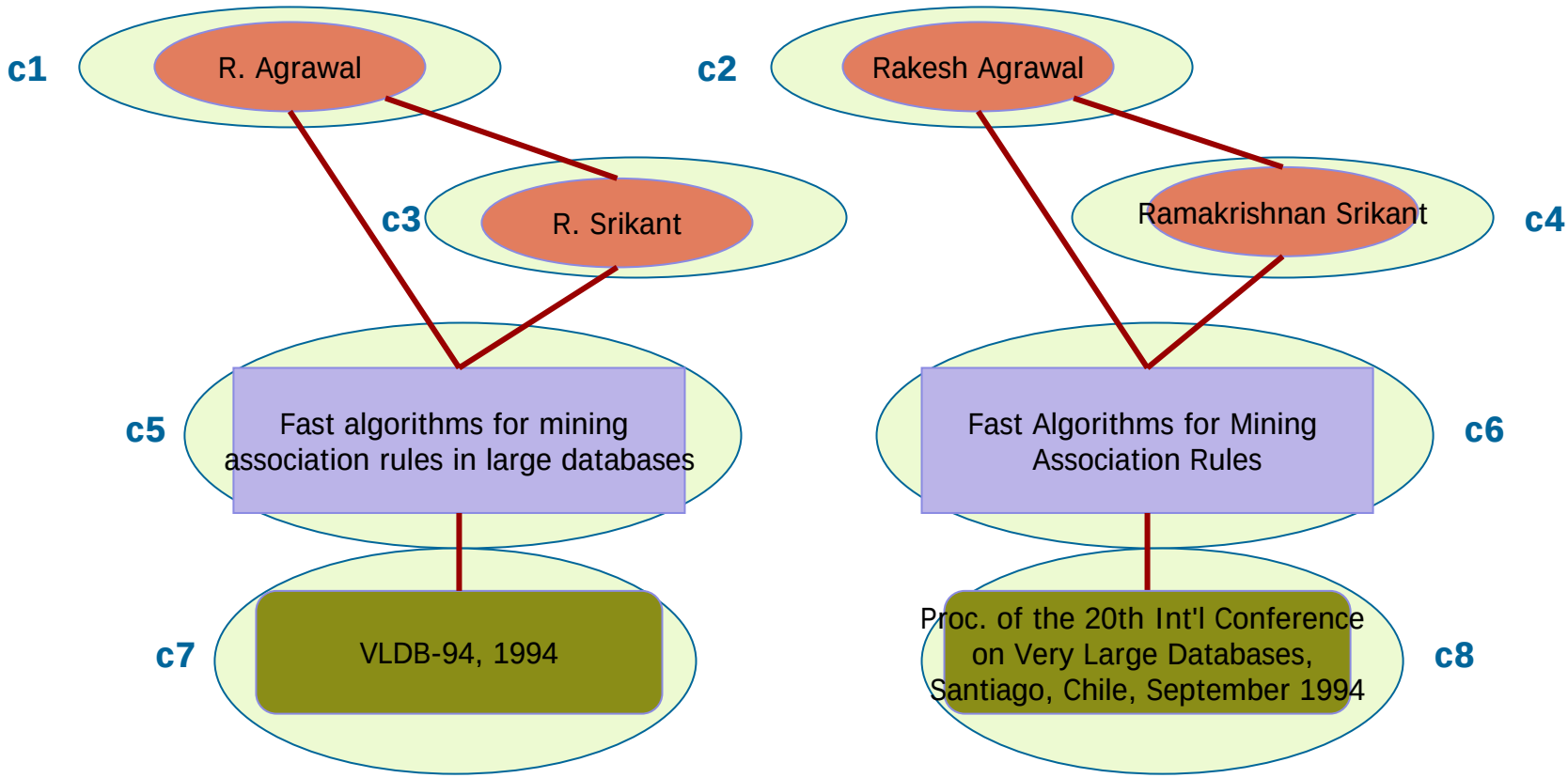


Exploiting Bibliographic Links



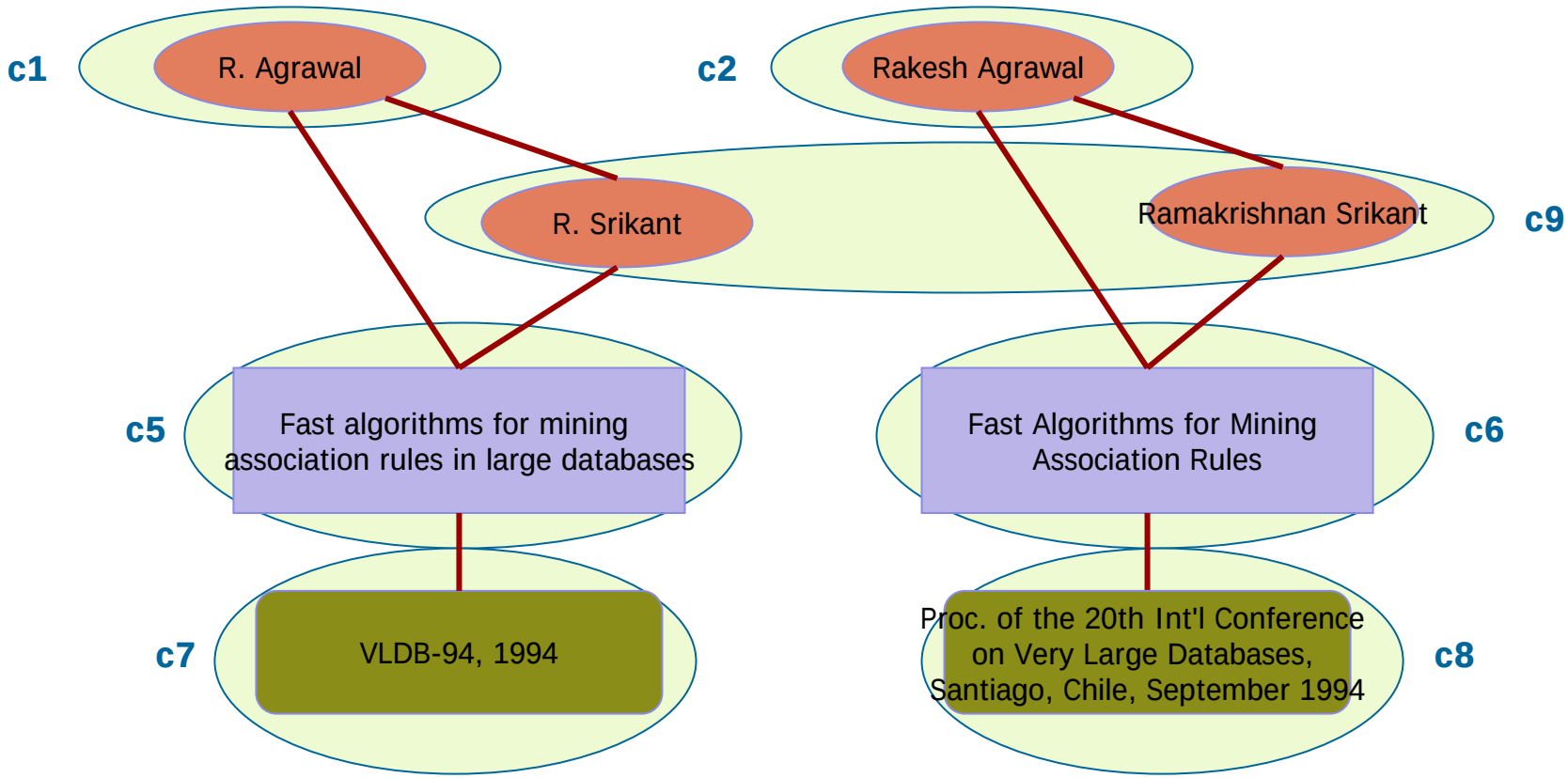
Approach 1: ER using Relational Clustering (RC-ER)

- Iteratively cluster 'similar' references into entities



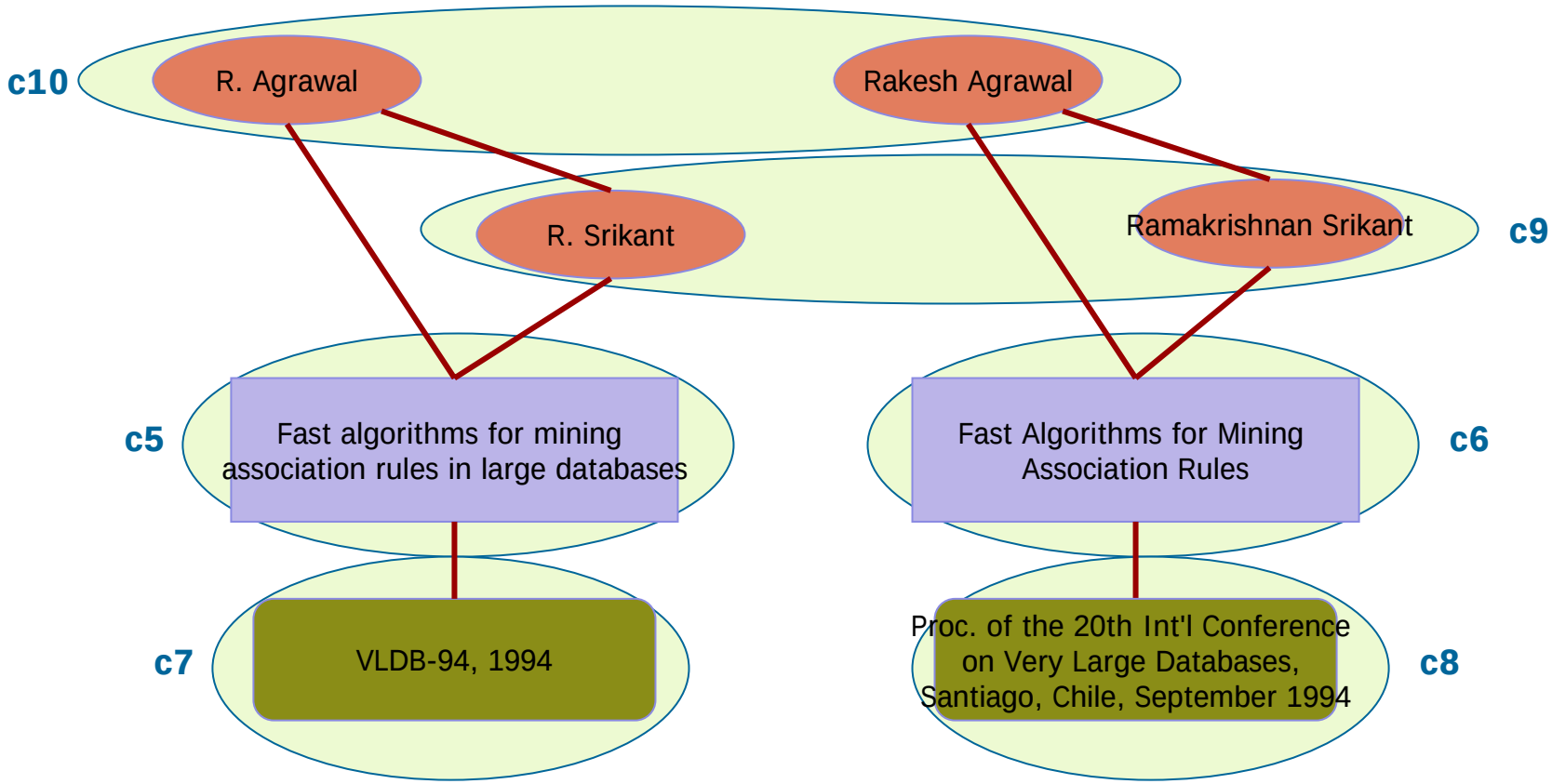
Approach 1: ER using Relational Clustering (RC-ER)

- Iteratively cluster 'similar' references into entities



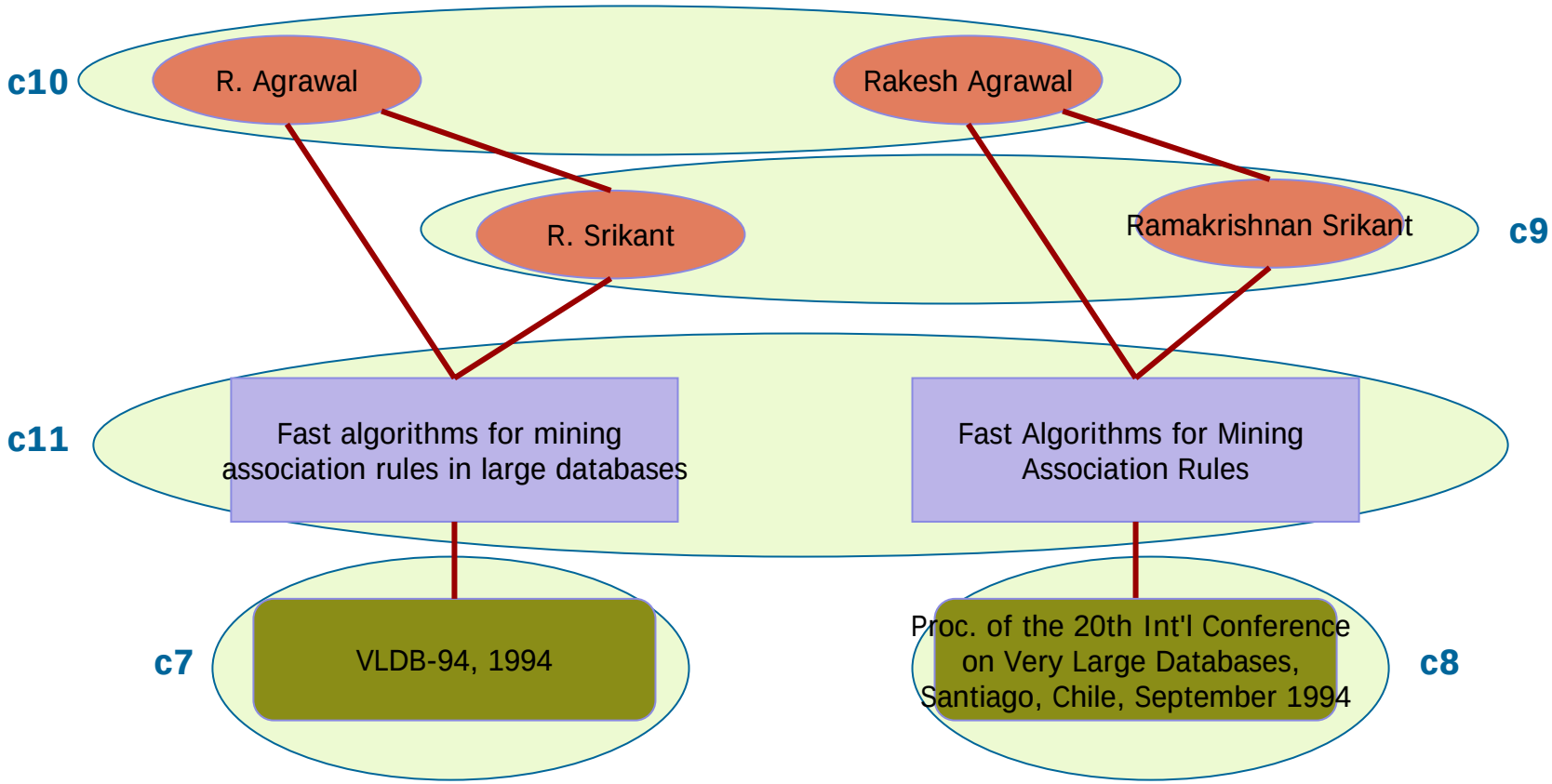
Approach 1: ER using Relational Clustering (RC-ER)

- Iteratively cluster 'similar' references into entities



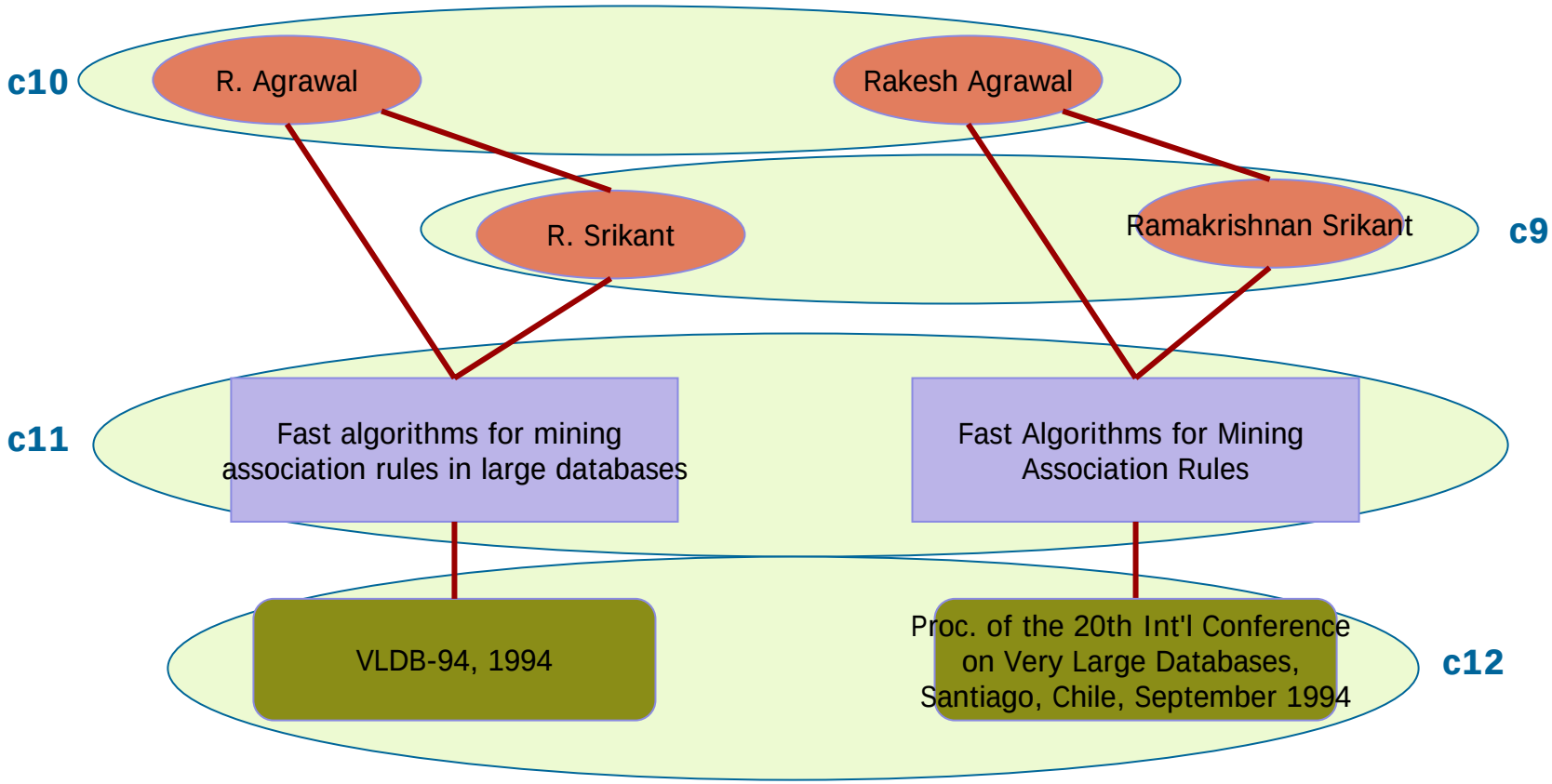
Approach 1: ER using Relational Clustering (RC-ER)

- Iteratively cluster 'similar' references into entities



Approach 1: ER using Relational Clustering (RC-ER)

- Iteratively cluster 'similar' references into entities



Similarity Measure For Clustering

$$sim(c_i, c_j) = (1- \alpha)*sim_{attr}(c_i, c_j) + \alpha *sim_{rel}(c_i, c_j)$$

Attribute similarity between clusters

Relational similarity between clusters

- **Attribute Similarity**: Compare attributes of individual references in the two clusters

1. Name: Single Valued Attribute
   - Cluster Similarity Metric / Representative Attribute
   - Jaro / Jaro-Winkler / Levenstein similarity with TF-IDF weights
2. Multi Valued Attributes
   - Countries, Addresses, Keywords, Classifications
   - Vector with TF-IDF weights; Cosine Similarity

Similarity Measure For Clustering

$$sim(c_i, c_j) = (1- \alpha)*sim_{attr}(c_i, c_j) + \alpha *sim_{rel}(c_i, c_j)$$

Attribute similarity between clusters

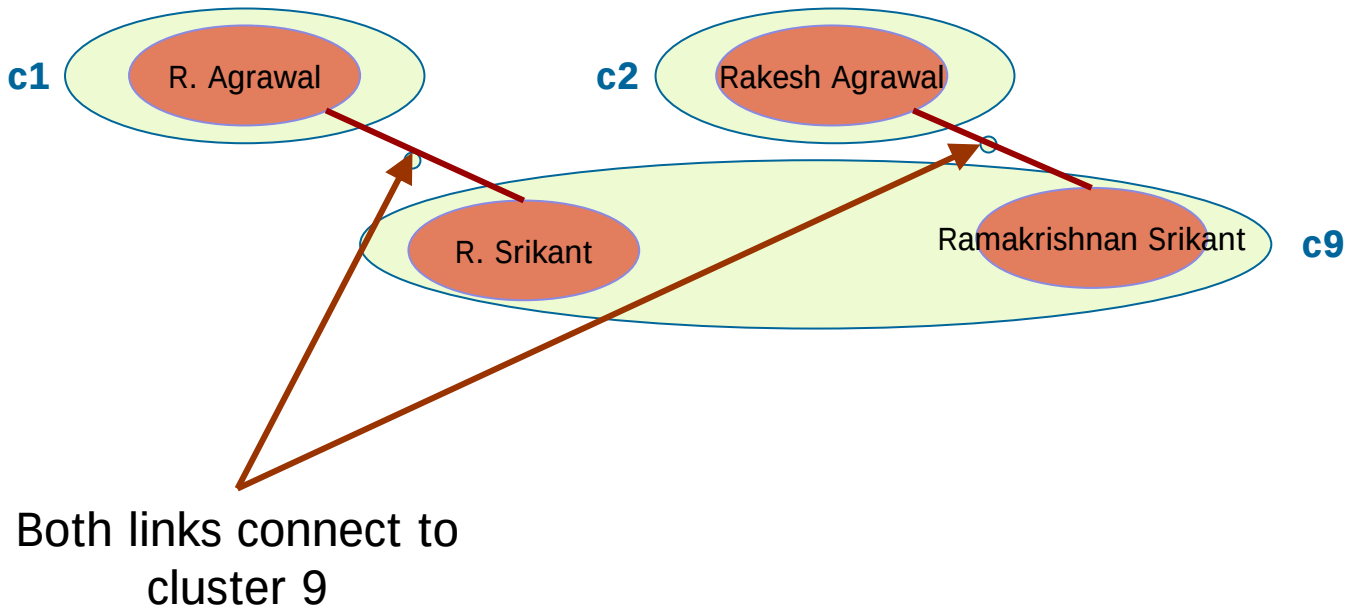
Relational similarity between clusters

- **Relational Similarity**: Use set similarity (eg Jaccard) to find shared clusters (resolutions) between links

1. Edge Detail Similarity
   - Compare individual links of two clusters
   - `Set of sets' similarity
   - Expensive
2. Neighborhood Similarity
   - Compare neighborhoods of two clusters
   - Reduce set of sets to multiset
   - Cheaper approximation

Edge Detail Similarity

- Similarity of two links depends on their references
  - Consider resolution decisions on the references



Edge Detail Similarity

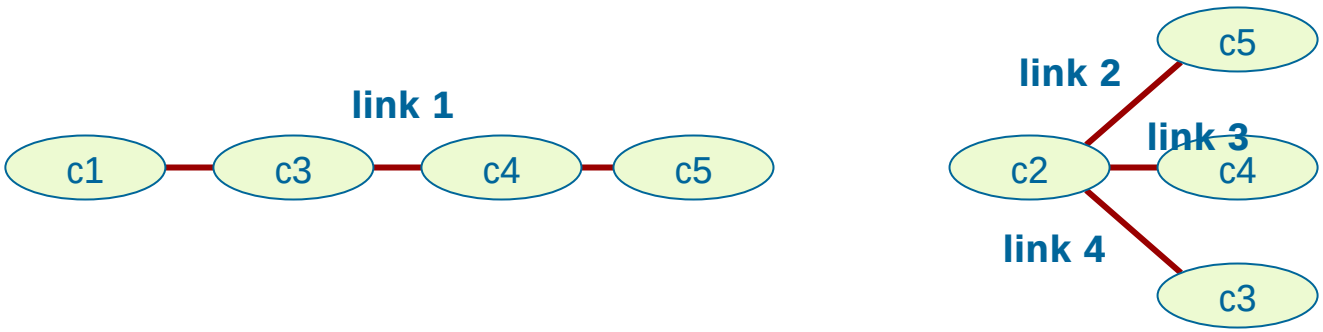
- Similarity of two links depends on their references
  - Consider resolution decisions on the references

- Label set $E_h(i)$ of i^{th} link
  - set of cluster labels of its reference
- $sim_h(i,j) = Jaccard(E_h(i), E_h(j))$

- Edge Detail Similarity of two clusters
  - $Sim_{rel}(c, c') = min(sim_h(i), sim_h(j)), i \in H(c), j \in H(c')$

Neighborhood Similarity

- Edge detail similarity is expensive
  - Ignore explicit link structure
  - Consider only set of neighborhood clusters



- Clusters c1, c2 still similar in terms of relationships

Neighborhood Similarity

- Edge detail similarity is expensive
  - Ignore explicit link structure
  - Consider only set of neighborhood clusters

- N(c) : multiset of cluster labels covered by links in H(c)
- Neighborhood similarity of two clusters
  - $Sim_{rel}(c,c') = Jaccard(N(c),N(c'))$

Approach #1: Algorithm (GBC-ER)

- Iteratively merge the most similar cluster pairs
- Similarities are dynamic: Update related similarities after each merge
- Indexed priority queue for fast update and extraction
- Relational bootstrapping for improvements in performance and efficiency

Baseline

- Pairwise duplicate decisions using Soft-TFIDF (ATTR)
  - Secondary string similarity: Scaled Levenstein(SL), Jaro(JA), Jaro-Winkler(JW)
- Transitive Closure over pairwise decisions (ATTR*)
- Precision, Recall and F1 over pairwise decisions
- Requires similarity threshold
  - Report **best** performance over all thresholds

Evaluation Datasets

- CiteSeer
  - Machine Learning Citations
  - Originally created by Lawrence et al.
  - 2,892 references to 1,165 true authors
  - 1,504 links

- arXiv HEP
  - Papers from High Energy Physics
  - Used for KDD-Cup '03 Data Cleaning Challenge
  - 58,515 references to 9,200 true authors
  - 29,555 links

- BioBase
  - Biology papers on immunology and infectious diseases
  - IBM KDD Challenge dataset constructed at Cornell
  - 156,156 publications, 831,991 author references
  - Ground truth for only ~1060 references

GBC Results: Best F1

	CiteSeer	HEP	BioBase
Attr	0.980	0.974	0.701
Attr*	0.990	0.967	0.687
GBC-Nbr	0.994	**0.985**	**0.819**
GBC-Edge	**0.995**	0.983	0.814

- Relational measures improve performance over attribute baseline in terms of precision, recall and F1
- Neighbor similarity performs almost as well as edge detail or better
- Neighborhood similarity **much** faster than edge detail

Structural Difference between Data Sets

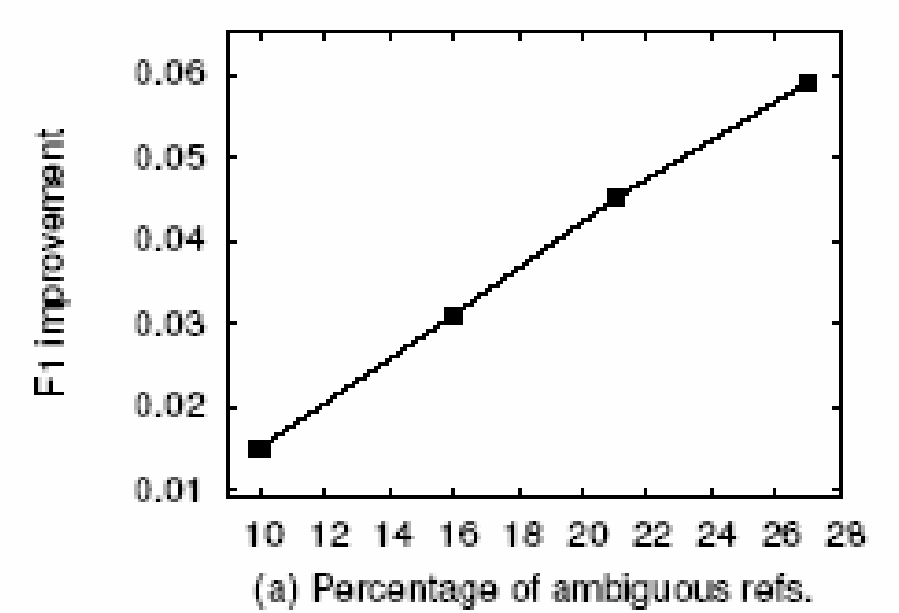
- Percentage of Ambiguous References
  - 0.5 % for Citeseer
  - 9% for HEP
  - 32% for BioBase
- Average number of collaborators per author
  - 2.15 for Citeseer
  - 4.5 for HEP
- Average number of references per author
  - 2.5 for Citeseer
  - 6.4 for HEP
  - 106 for BioBase

Synthetic Data Generator

- Data generator mimics real collaborations
- Create collaboration graph in Stage 1
- Create documents from this graph in Stage 2
- Can control
  - Number of entities and documents
  - Average number of collaborators per author
  - Average number of references per entity
  - Average number of references per document
  - Percentage of ambiguous references
  - …

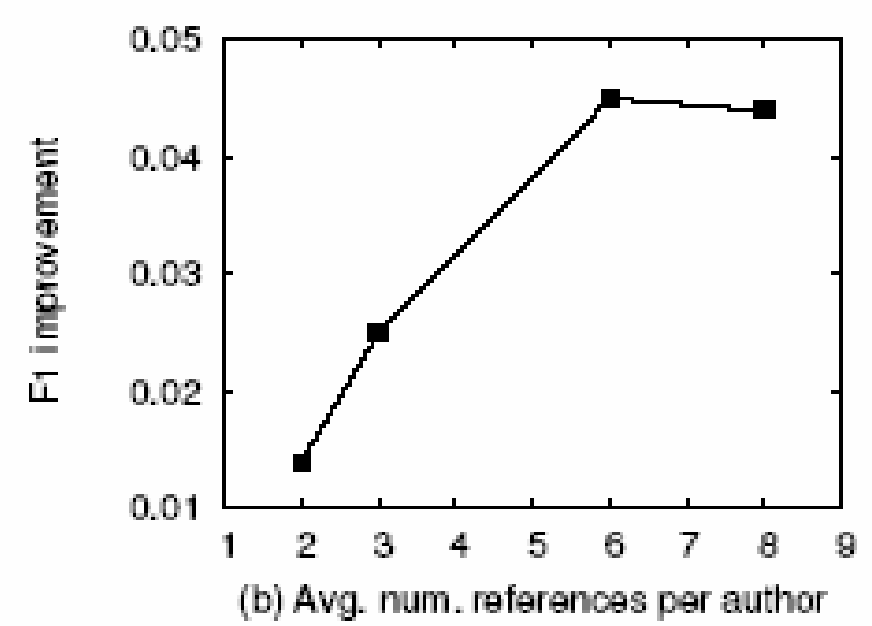
Trends in Synthetic Data

- Improvement increases sharply with higher ambiguity in references



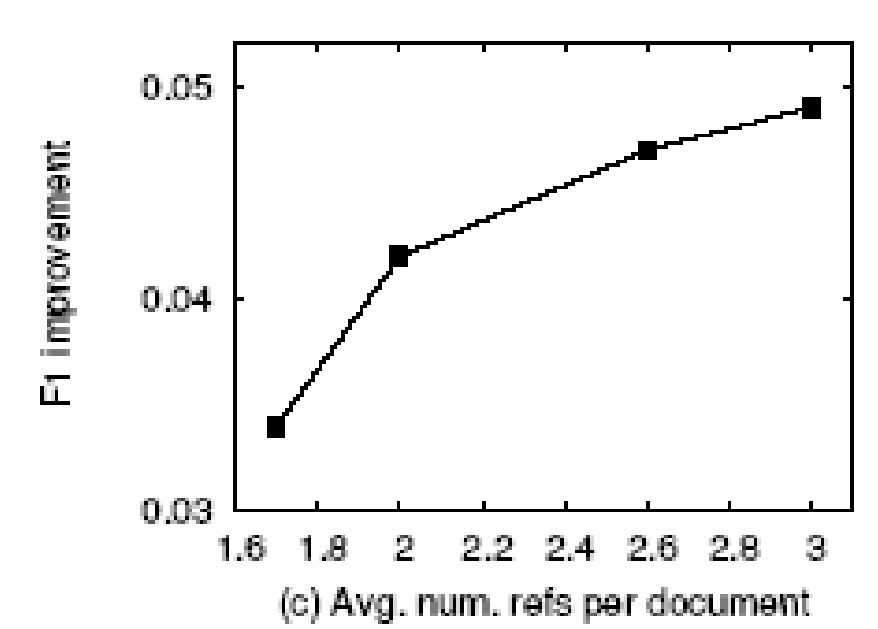
Trends in Synthetic Data

- Improvement increases with more references per author



Trends in Synthetic Data

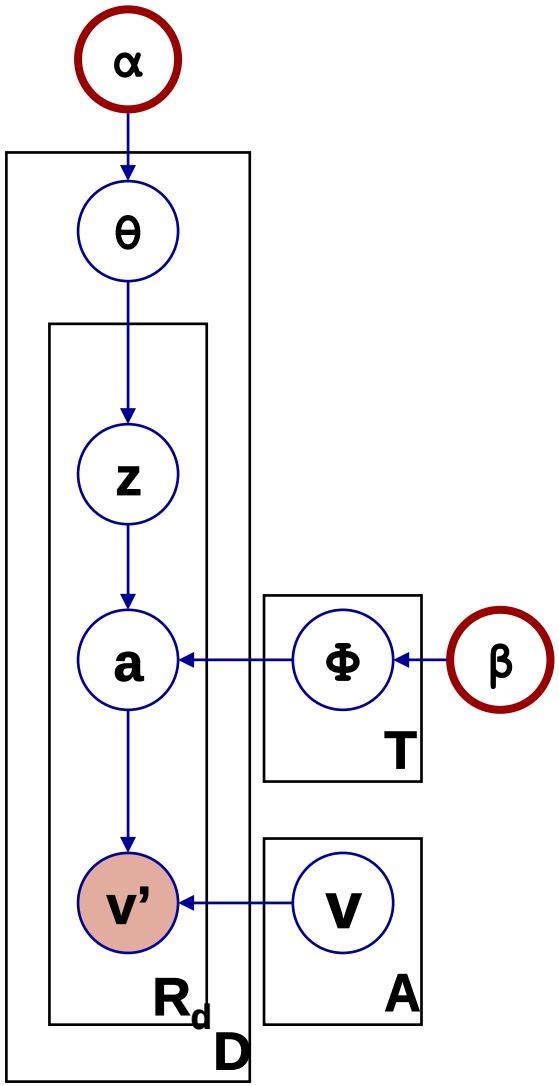
- Improvement increases with more references per document



Approach #2: Latent Dirichlet Model for ER

- Probabilistic model of entity collaboration groups
  - Entities (authors) belong to groups
  - Entities (authors) in a link (document) depend on the groups that are involved
- Latent group variable for each reference
- Group labels and entity labels unobserved

LDA for Entity Resolution (LDA-ER)



- Author entities not directly observed
- Generate entity a as before
- Entities have attributes **v**
- Generate attribute v_i' for i^{th} reference from entity attribute v_a using noise process

LDA-ER Contributions

- Group labels capture relationships among entities
- Group label and entity label for each reference rather than a variable for each pair
- Unsupervised learning of labels
- Number of entities not assumed to be known
  - Gibbs sampling to infer number of entities

LDA-ER Performance

- CiteSeer
  - Improves precision
  - 22% reduction in error
- arXiv
  - Improves recall as well as precision
  - 20% reduction in error

ER Algorithm Comparison

- Two approaches to relational entity resolution

1. Graph-Based Clustering
   - Efficient
   - Customizable attribute similarity measure
   - Performs slightly better than probabilistic model
   - Unsupervised -- needs threshold to determine duplicates

2. Probabilistic Generative Model
   - Notion of optimal solution
   - Group label for references
   - Can generalize for unseen data
   - Able to handle noise

Entity Resolution

- The Problem
- Relational Entity Resolution
- Algorithms
  - Graph-based Clustering (GBC-ER)
  - Probabilistic Model (LDA-ER)
- **Query-time Entity Resolution**
- ER User Interface

Query-time Entity Resolution

- Goal: Allow users to query an unresolved or partially resolved database
- Adaptive strategy which constructs set of relevant references and performs **collective** resolution
- Define canonical queries:
  - Disambiguation query
  - Entity Resolution query

Preliminary Results: F1

	arXiv	Biobase
Attr	0.72	0.71
Attr*	0.77	0.68
Naïve Rel	0.95	0.71
Naïve Rel*	0.95	0.75
Collective ER – depth 1	0.96	0.81
Collective ER – depth 3	0.97	0.82

Adaptive Strategy 200 times faster and just as accurate

IBM KDD Entity Resolution Challenge

- Recent bake-off among researchers in KDD program
- Our algorithms performed among the top; especially impressive since our algorithms are unsupervised
- Focused our efforts on scalability, query specific entity resolution, caching, etc.

D-Dupe: An Interactive Tool for ER

- Tool Integrates
  - entity resolution algorithms
  - simple visual interface optimized for ER
- Case studies on bibliographic datasets
  - on two clean datasets we quickly were able to find many duplicates
  - on one dataset w/o author keys, we were able to easily clean dataset to construct keys
- Currently
  - adapting tool for database integration
    - geospatial data
    - academic genealogy
    - email archives

ER References

- Bibliographic Data
  - Author resolution using co-author links
    1. Graph-based Clustering (**GBC-ER**)
       (DMKD '04, LinkKDD '04, Book Chapter, Tech Report)
    2. LDA based Group model (**LDA-ER**)
       (SDM '06,best paper awqard)
    3. Query-based Entity Resolution (**QB-ER**)
       Participants in IBM KDD Entity Resolution Challenge

- Email Archives
  - Name reference resolution using email traffic network
    1. Using a variety of temporal social network models
       (SDM '06)

- Natural Language
  - Sense resolution using translation links in parallel corpora **(ACL '04)**
    1. **Sense Model**: Senses in different languages depend directly on each other
    2. **Concept Model**: Semantic sense groups or Concepts relate senses from different languages

Thanks!!