Entity Resolution: Tutorial

Lise Getoor

University of Maryland
College Park, MD

Ashwin Machanavajjhala

Duke University
Durham, NC

http://www.cs.umd.edu/~getoor/Tutorials/ER_ASONAM2012.pdf

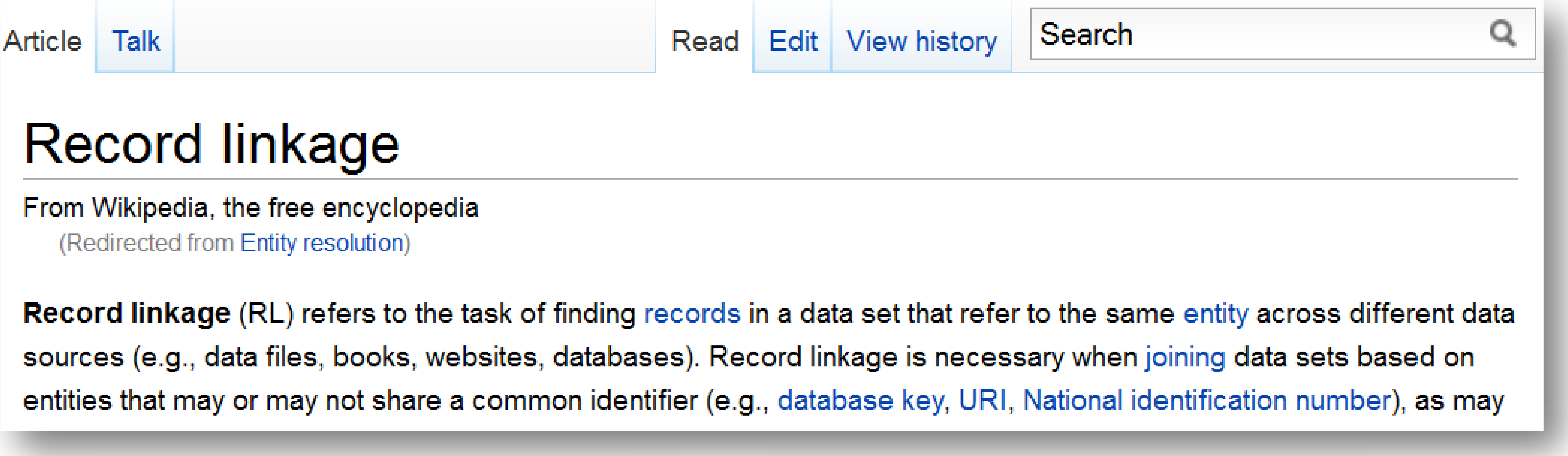
What is Entity Resolution?

Problem of identifying and linking/grouping different manifestations of the same real world object.

Examples of manifestations and objects:

- Different ways of addressing (names, email addresses, FaceBook accounts) the same person in text.
- Web pages with differing descriptions of the same business.
- Different photos of the same object.
- …

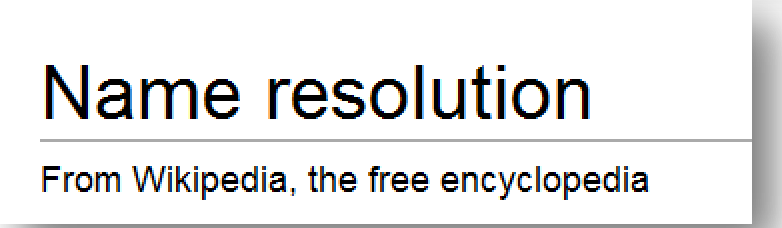
What is Entity Resolution?



Record linkage

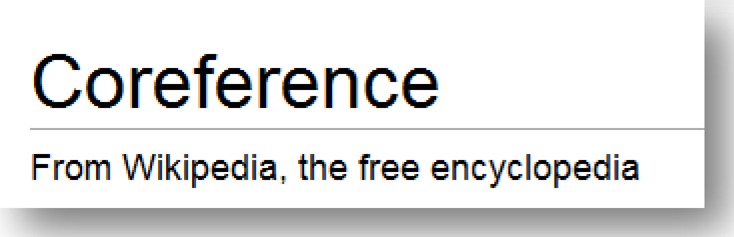
From Wikipedia, the free encyclopedia
(Redirected from Entity resolution)

Record linkage (RL) refers to the task of finding records in a data set that refer to the same entity across different data sources (e.g., data files, books, websites, databases). Record linkage is necessary when joining data sets based on entities that may or may not share a common identifier (e.g., database key, URI, National identification number), as may



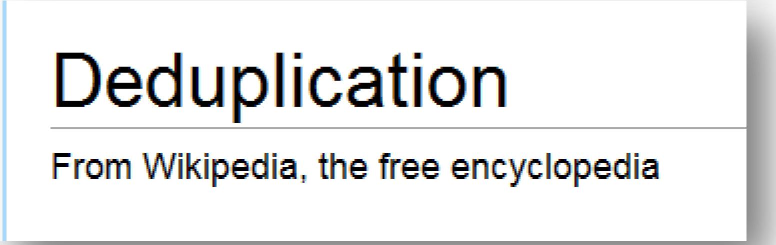
Name resolution

From Wikipedia, the free encyclopedia



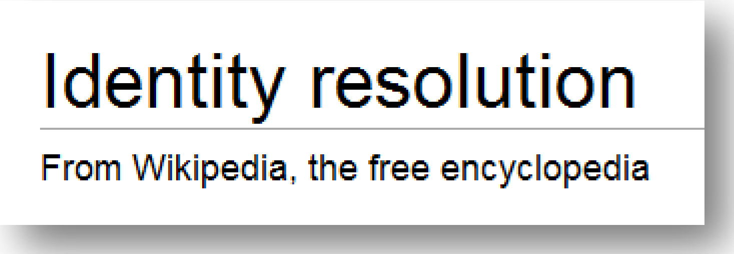
Coreference

From Wikipedia, the free encyclopedia



Deduplication

From Wikipedia, the free encyclopedia



Identity resolution

From Wikipedia, the free encyclopedia

Ironically, Entity Resolution has many duplicate names

Household matching

Doubles

Duplicate detection

Mixed and split citation problem

Record linkage

Object identification

Match

Deduplication

Fuzzy match

Object consolidation

Entity resolution

Entity clustering

Approximate match

Identity uncertainty

Reference reconciliation

Merge/purge

Hardening soft databases

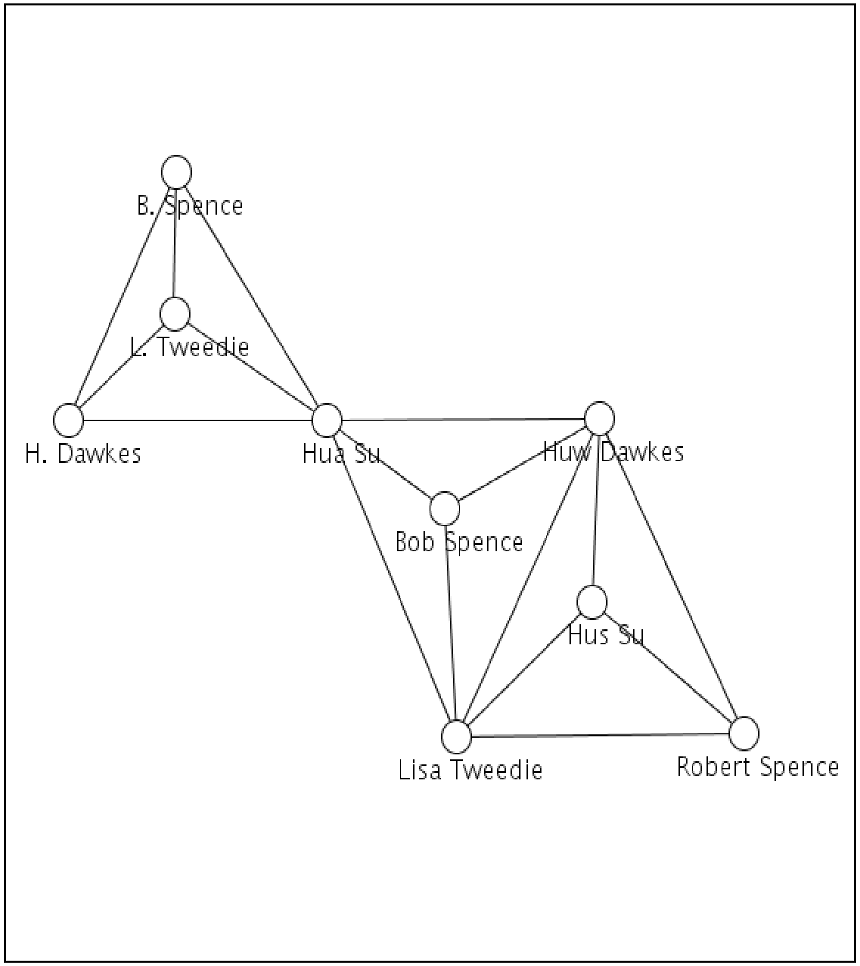
Householding

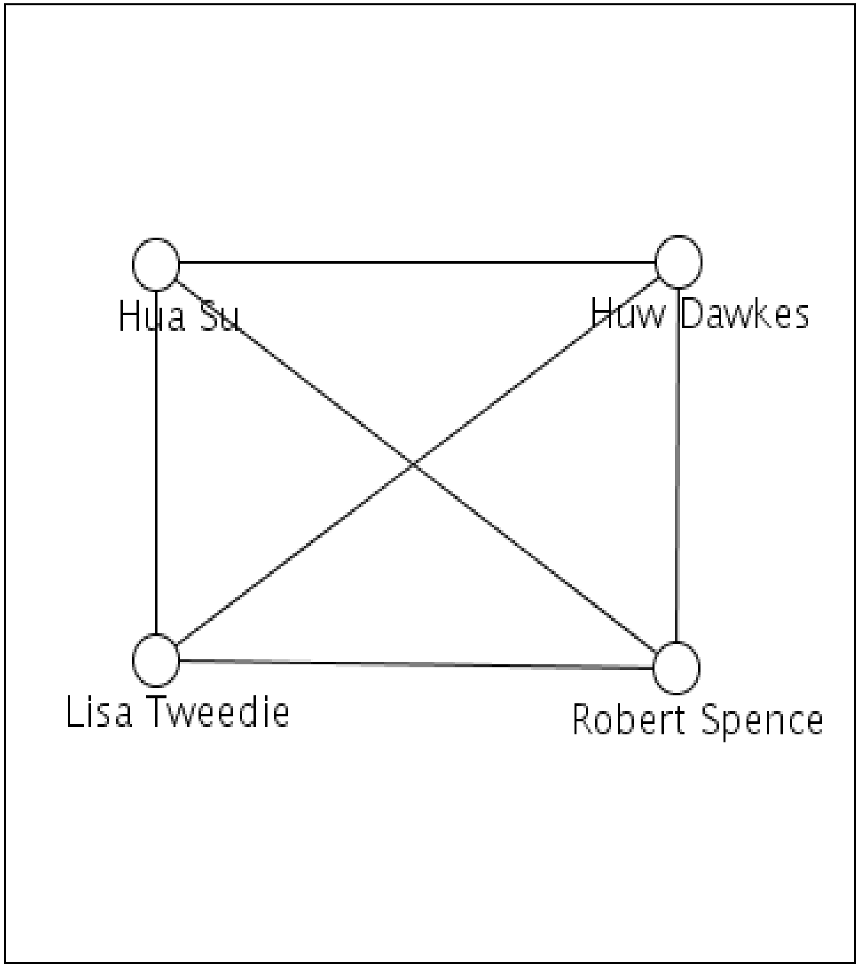
Reference matching

ER Motivating Examples

- *Linking Census Records*
- *Public Health*
- *Web search*
- *Comparison shopping*
- *Counter-terrorism*
- *Spam detection*
- *Machine Reading*
- …

ER and Network Analysis



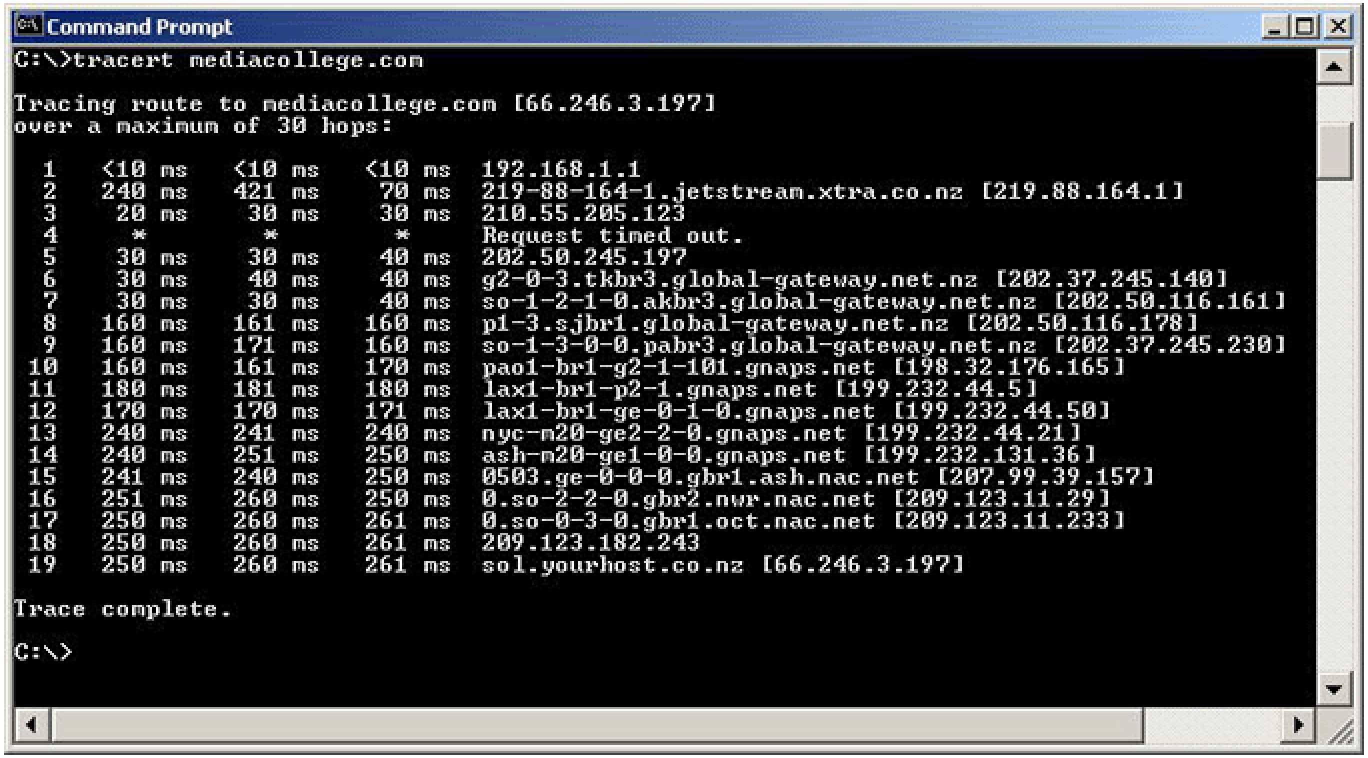


before

after

Motivation: Network Science

- Measuring the topology of the internet … using `traceroute`



```
C:\>tracert mediacollege.com

Tracing route to mediacollege.com [66.246.3.197]
over a maximum of 30 hops:

  1    <10 ms   <10 ms   <10 ms  192.168.1.1
  2    240 ms   421 ms    70 ms  219-88-164-1.jetstream.xtra.co.nz [219.88.164.1]
  3     20 ms    30 ms    30 ms  210.55.205.123
  4      *        *        *     Request timed out.
  5     30 ms    30 ms    40 ms  202.50.245.197
  6     30 ms    40 ms    40 ms  g2-0-3.tkbr3.global-gateway.net.nz [202.37.245.140]
  7     30 ms    30 ms    40 ms  so-1-2-1-0.akbr3.global-gateway.net.nz [202.50.116.161]
  8    160 ms   161 ms   160 ms  p1-3.sjbr1.global-gateway.net.nz [202.50.116.178]
  9    160 ms   171 ms   160 ms  so-1-3-0-0.pabr3.global-gateway.net.nz [202.37.245.230]
 10    160 ms   161 ms   170 ms  pao1-br1-g2-1-101.gnaps.net [198.32.176.165]
 11    180 ms   181 ms   180 ms  lax1-br1-p2-1.gnaps.net [199.232.44.5]
 12    170 ms   170 ms   171 ms  lax1-br1-ge-0-1-0.gnaps.net [199.232.44.50]
 13    240 ms   241 ms   240 ms  nyc-m20-ge2-2-0.gnaps.net [199.232.44.21]
 14    240 ms   251 ms   250 ms  ash-m20-ge1-0-0.gnaps.net [199.232.131.36]
 15    241 ms   240 ms   250 ms  0503.ge-0-0-0.gbr1.ash.nac.net [207.99.39.157]
 16    251 ms   260 ms   250 ms  0.so-2-2-0.gbr2.nwr.nac.net [209.123.11.29]
 17    250 ms   260 ms   261 ms  0.so-0-3-0.gbr1.oct.nac.net [209.123.11.233]
 18    250 ms   260 ms   261 ms  209.123.182.243
 19    250 ms   260 ms   261 ms  sol.yourhost.co.nz [66.246.3.197]

Trace complete.

C:\>
```

IP Aliasing Problem [Willinger et al. 2009]

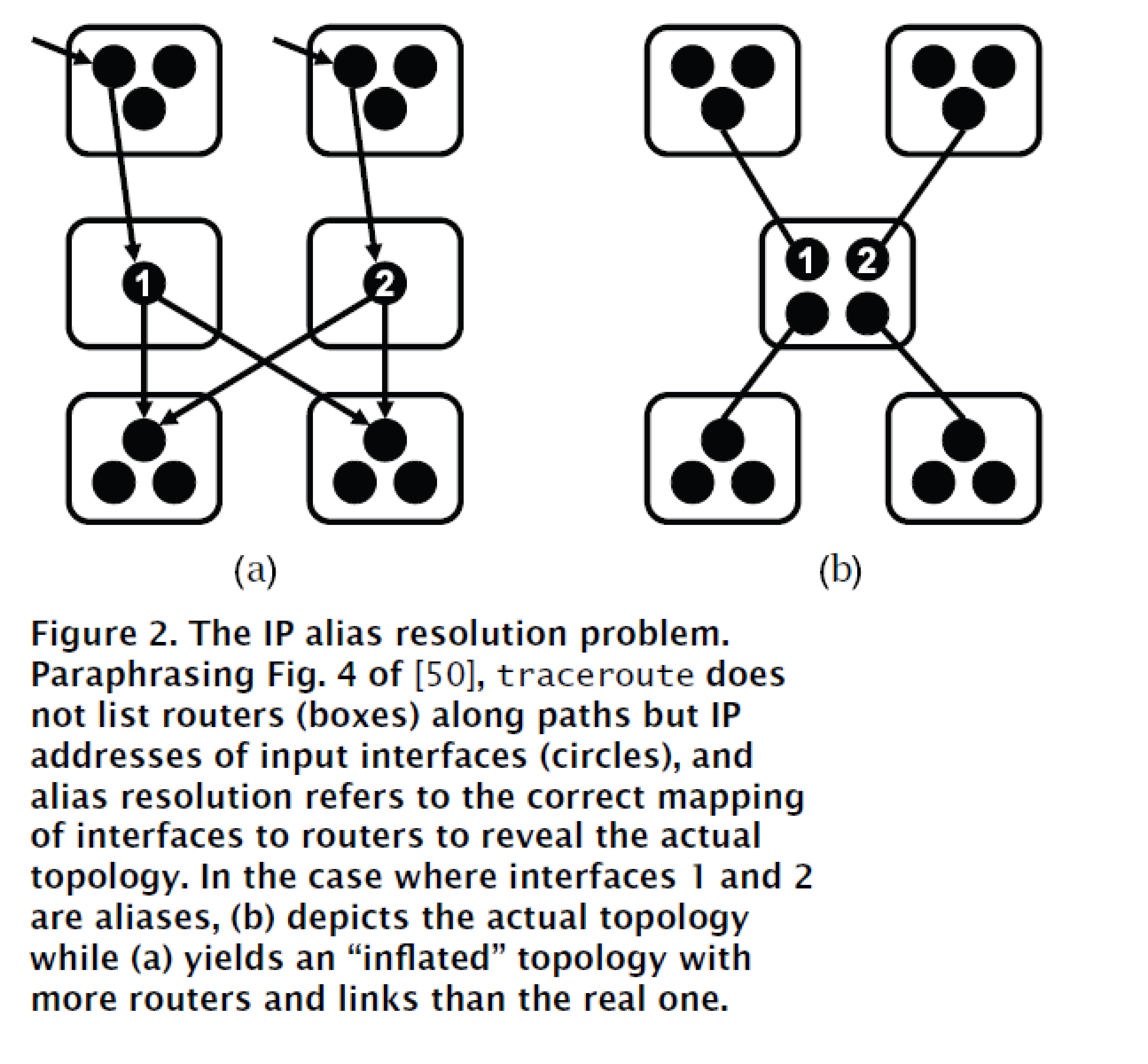


Figure 2. The IP alias resolution problem. Paraphrasing Fig. 4 of [50], `traceroute` does not list routers (boxes) along paths but IP addresses of input interfaces (circles), and alias resolution refers to the correct mapping of interfaces to routers to reveal the actual topology. In the case where interfaces 1 and 2 are aliases, (b) depicts the actual topology while (a) yields an "inflated" topology with more routers and links than the real one.

IP Aliasing Problem [Willinger et al. 2009]

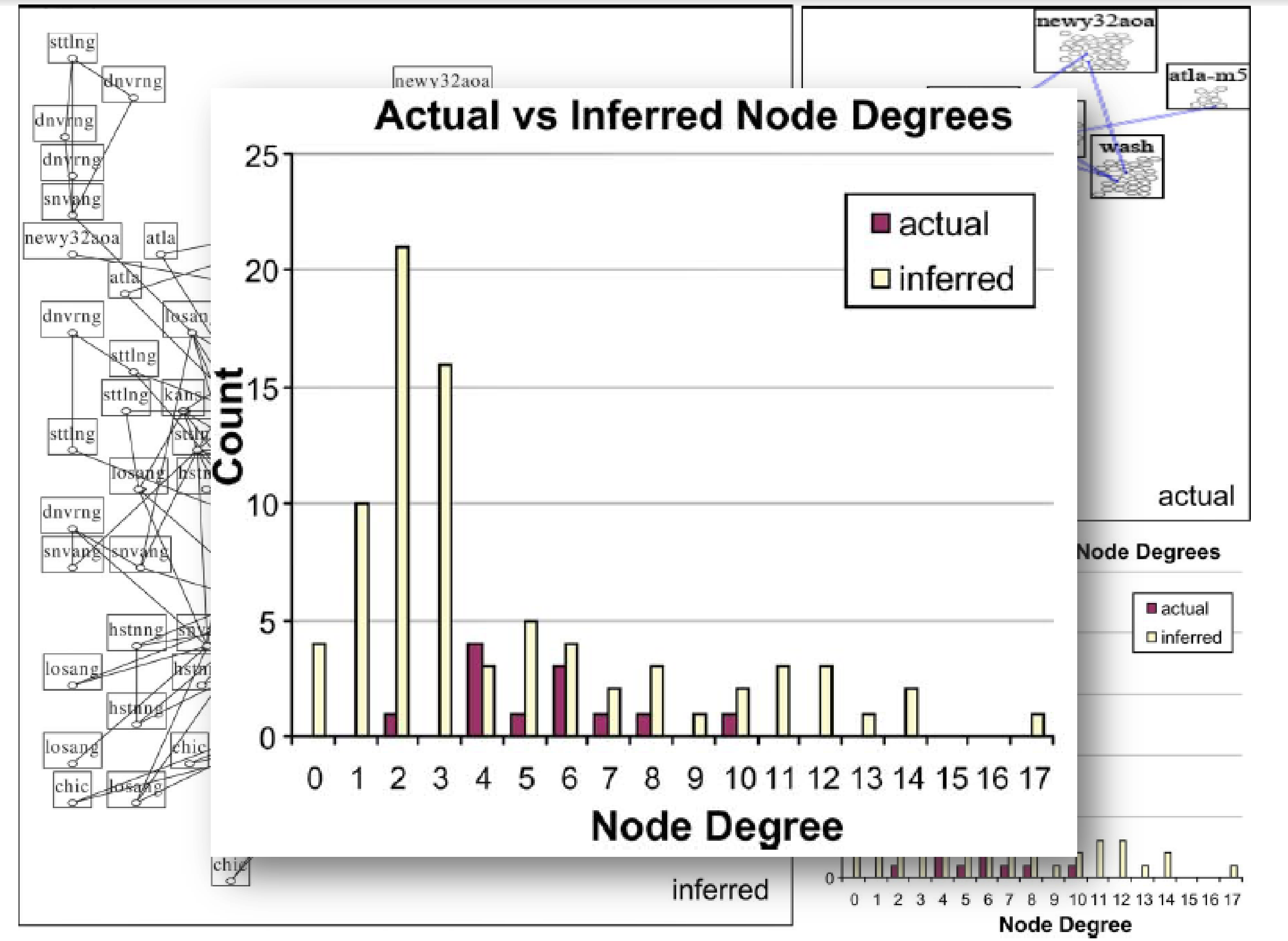
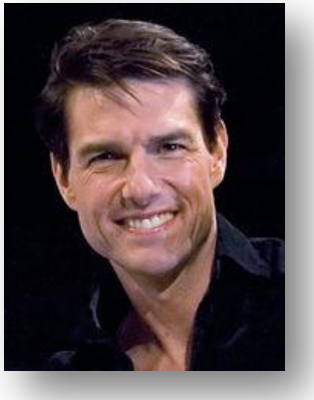


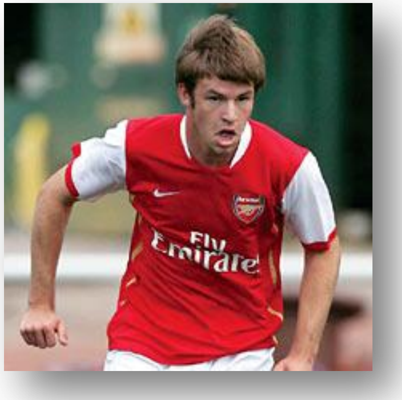
Figure 3. The IP alias resolution problem in practice. This is re-produced from [48] and shows a comparison between the Abilene/Internet2 topology inferred by Rocketfuel (left) and the actual topology (top right). Rectangles represent routers with interior ovals denoting interfaces. The histograms of the corresponding node degrees are shown in the bottom right plot. © 2008 ACM,

Traditional Challenges in ER

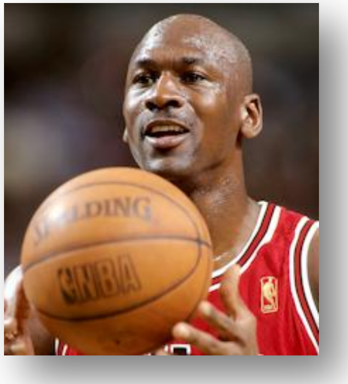
- Name/Attribute ambiguity

Thomas Cruise





Michael Jordan

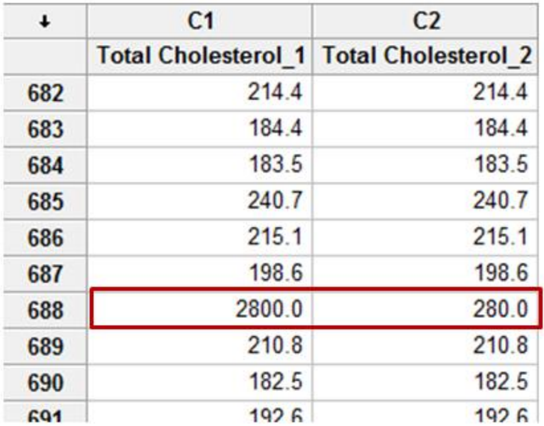


Traditional Challenges in ER

- Name/Attribute ambiguity
- Errors due to data entry

© Original Artist
Reproduction rights obtainable from
www.CartoonStock.com

© AP



↓	C1	C2
	Total Cholesterol_1	Total Cholesterol_2
682	214.4	214.4
683	184.4	184.4
684	183.5	183.5
685	240.7	240.7
686	215.1	215.1
687	198.6	198.6
688	2800.0	280.0
689	210.8	210.8
690	182.5	182.5
691	192.6	192.6

Traditional Challenges in ER

- Name/Attribute ambiguity
- Errors due to data entry
- Missing Values

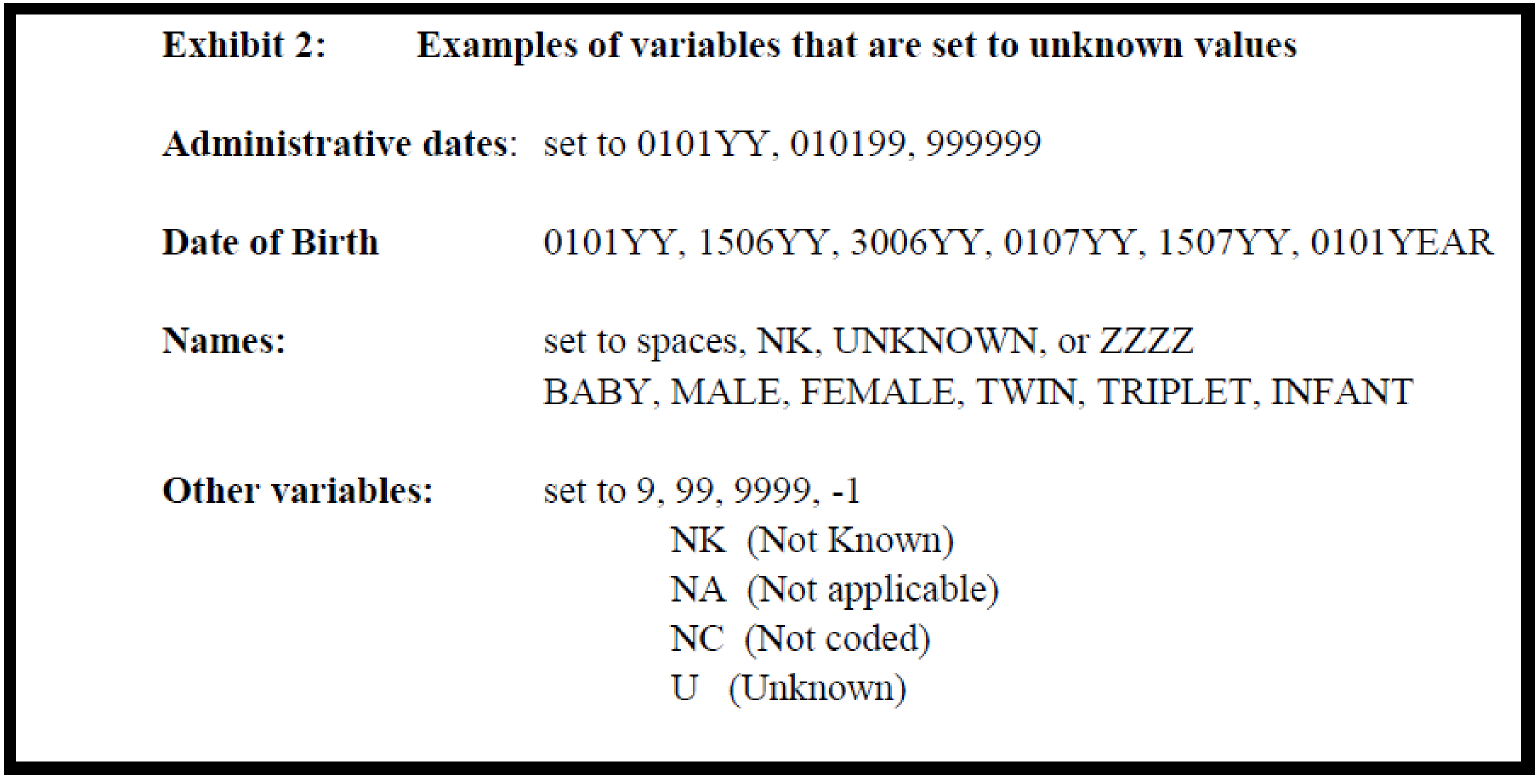


Exhibit 2: Examples of variables that are set to unknown values

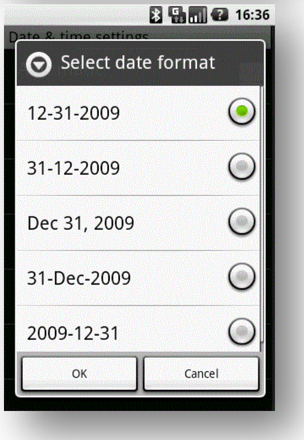
Administrative dates:	set to 0101YY, 010199, 999999
Date of Birth	0101YY, 1506YY, 3006YY, 0107YY, 1507YY, 0101YEAR
Names:	set to spaces, NK, UNKNOWN, or ZZZZ BABY, MALE, FEMALE, TWIN, TRIPLET, INFANT
Other variables:	set to 9, 99, 9999, -1 NK (Not Known) NA (Not applicable) NC (Not coded) U (Unknown)

[Gill et al; Univ of Oxford 2003]

Traditional Challenges in ER

- Name/Attribute ambiguity
- Errors due to data entry
- Missing Values
- Changing Attributes

- Data formatting

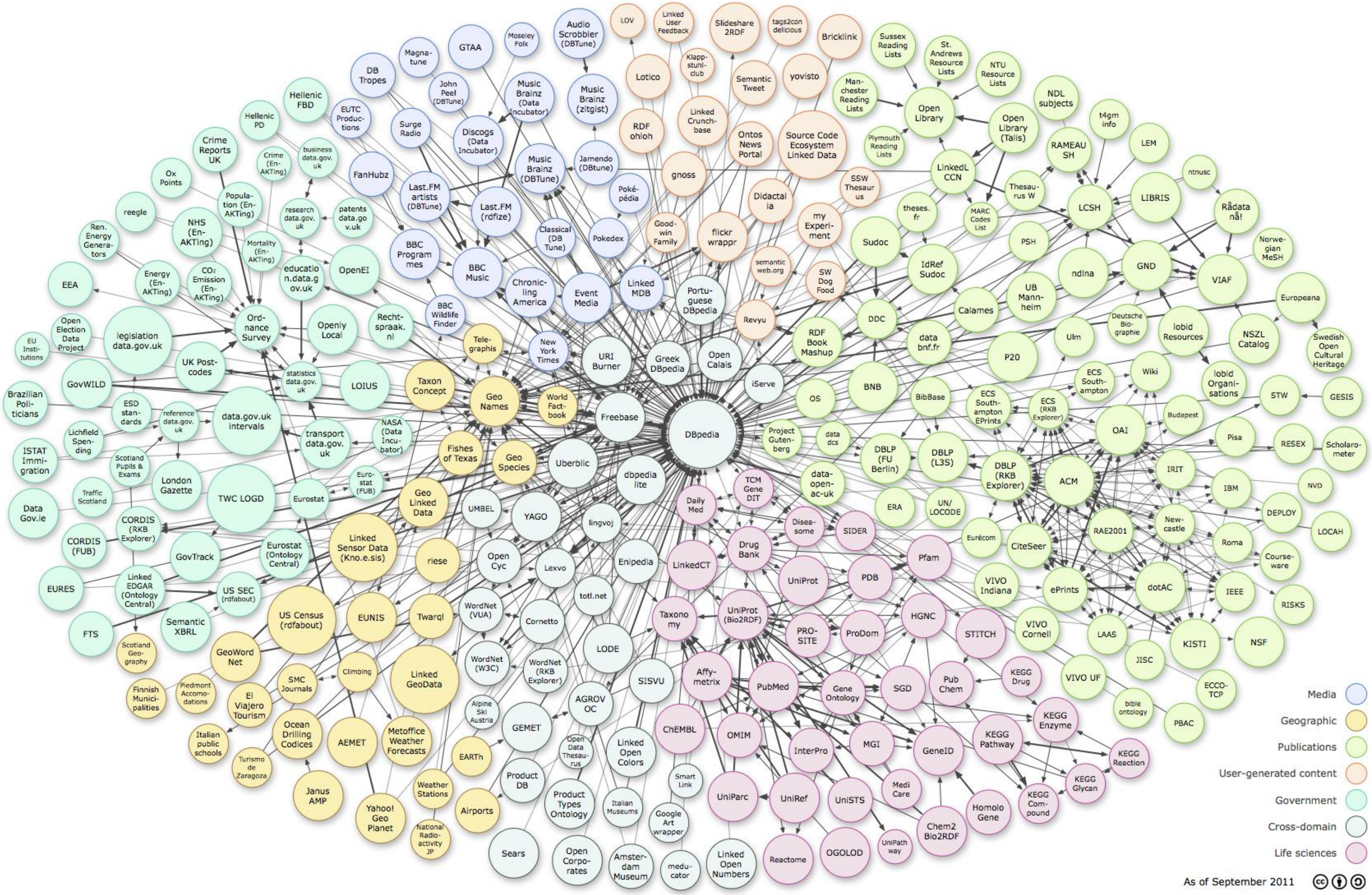


Select date format

- 12-31-2009 ●
- 31-12-2009 ○
- Dec 31, 2009 ○
- 31-Dec-2009 ○
- 2009-12-31 ○

- Abbreviations / Data Truncation

Big-Data ER Challenges



As of September 2011

Big-Data ER Challenges

- Larger and more Datasets
  - Need efficient parallel techniques
- More Heterogeneity
  - Unstructured, Unclean and Incomplete data. Diverse data types.
  - No longer just matching names with names, but Amazon profiles with browsing history on Google and friends network in Facebook.

Big-Data ER Challenges

- Larger and more Datasets
  - Need efficient parallel techniques
- More Heterogeneity
  - Unstructured, Unclean and Incomplete data. Diverse data types.
- More linked
  - Need to infer relationships in addition to “equality”
- Multi-Relational
  - Deal with structure of entities (Are Walmart and Walmart Pharmacy the same?)
- Multi-domain
  - Customizable methods that span across domains
- Multiple applications (web search versus comparison shopping)
  - Serve diverse application with different accuracy requirements

Outline

1. Classical Single Entity ER
2. Relational & MultiEntity ER
3. Efficiency: Blocking/Canopies
4. Challenges & Future Directions

PART 1

CLASSICAL SINGLE ENTITY ER

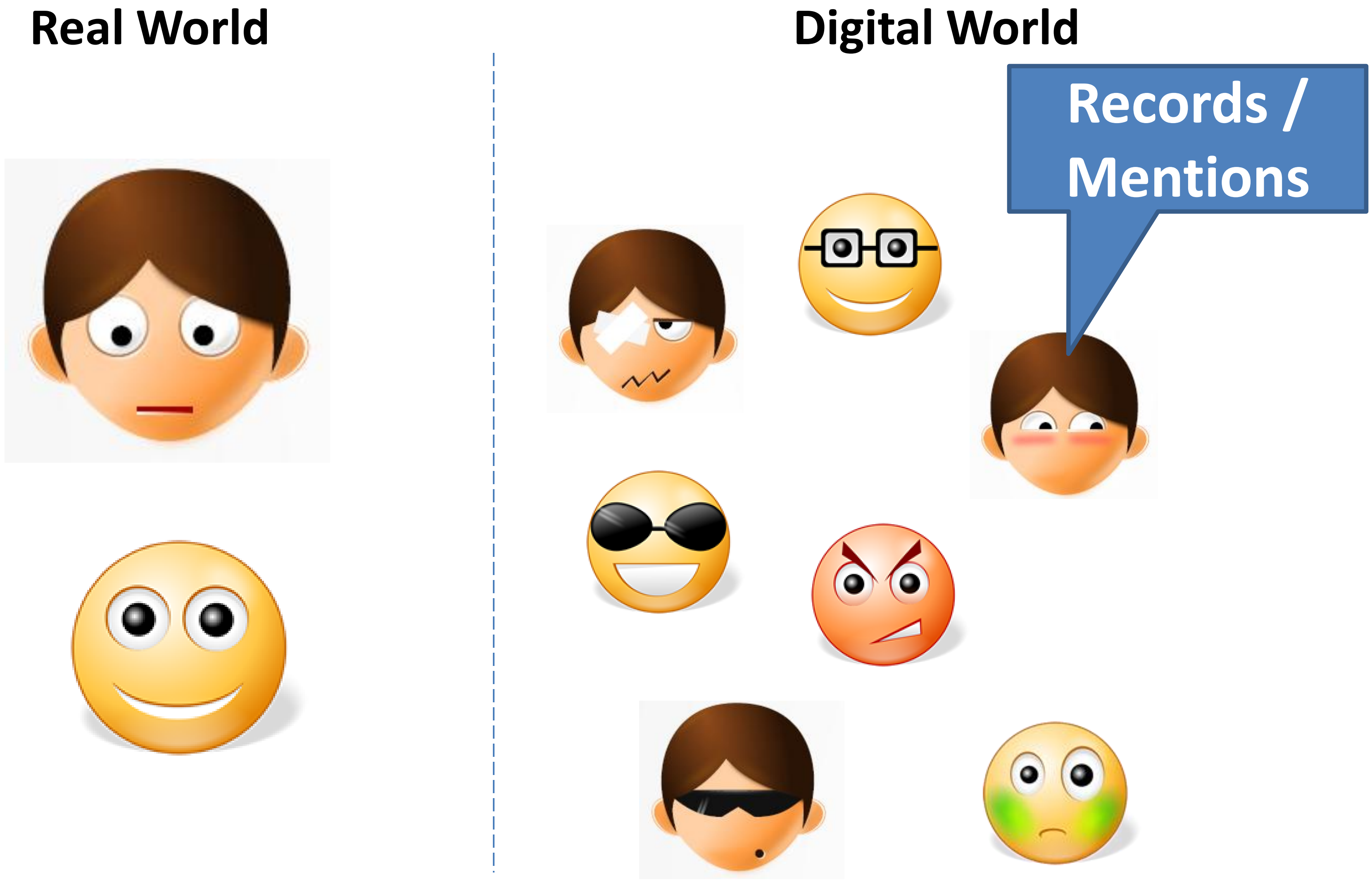
Outline

1. Classical Single Entity ER
    a) Problem Statement
    b) Data Preparation & Matching Features
    c) Algorithms for Single-Entity ER
    d) Canonicalization
2. Relational & MultiEntity ER
3. Efficiency: Blocking/Canopies
4. Challenges & Future Directions

PART 1-a

ER PROBLEM STATEMENT

Abstract Problem Statement



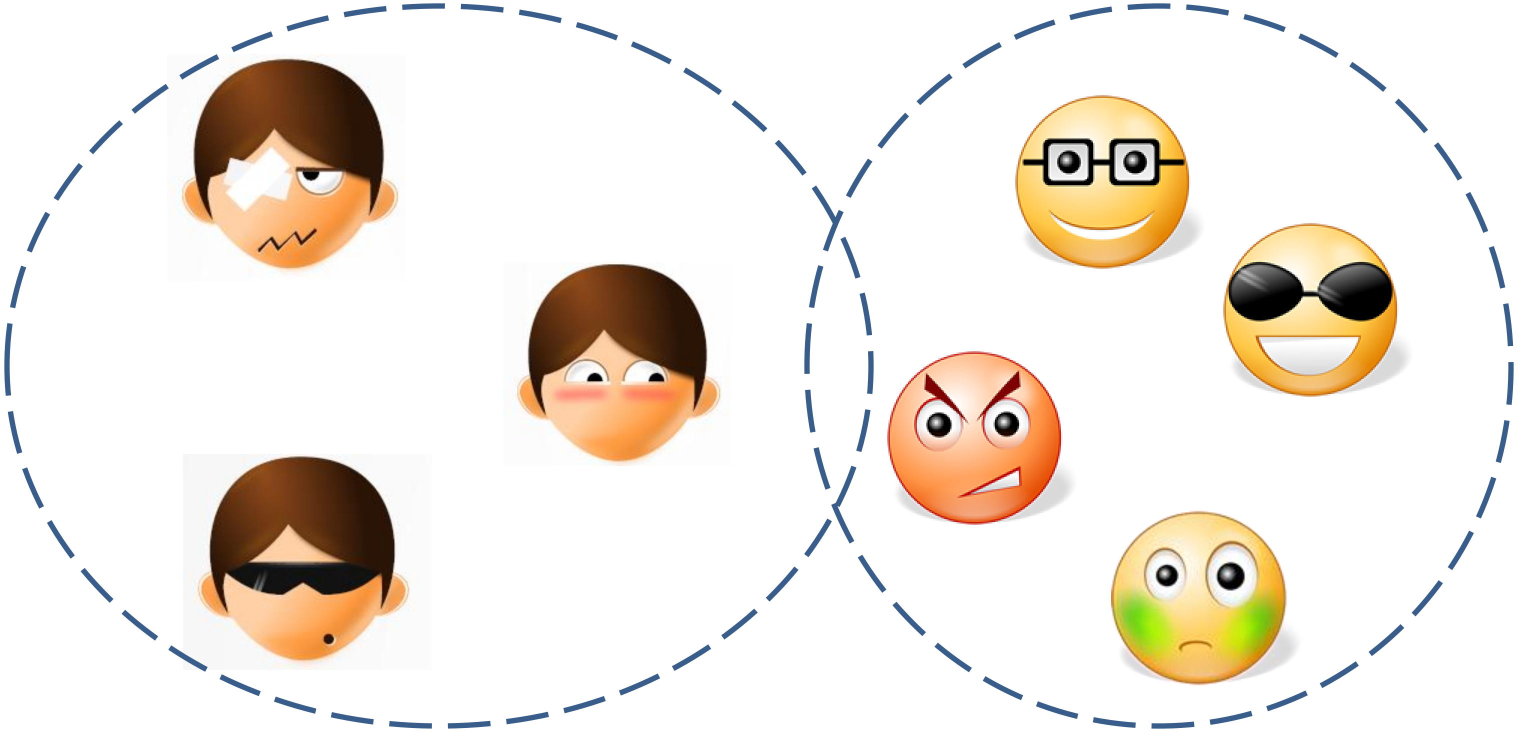
Real World

Digital World

Records / Mentions

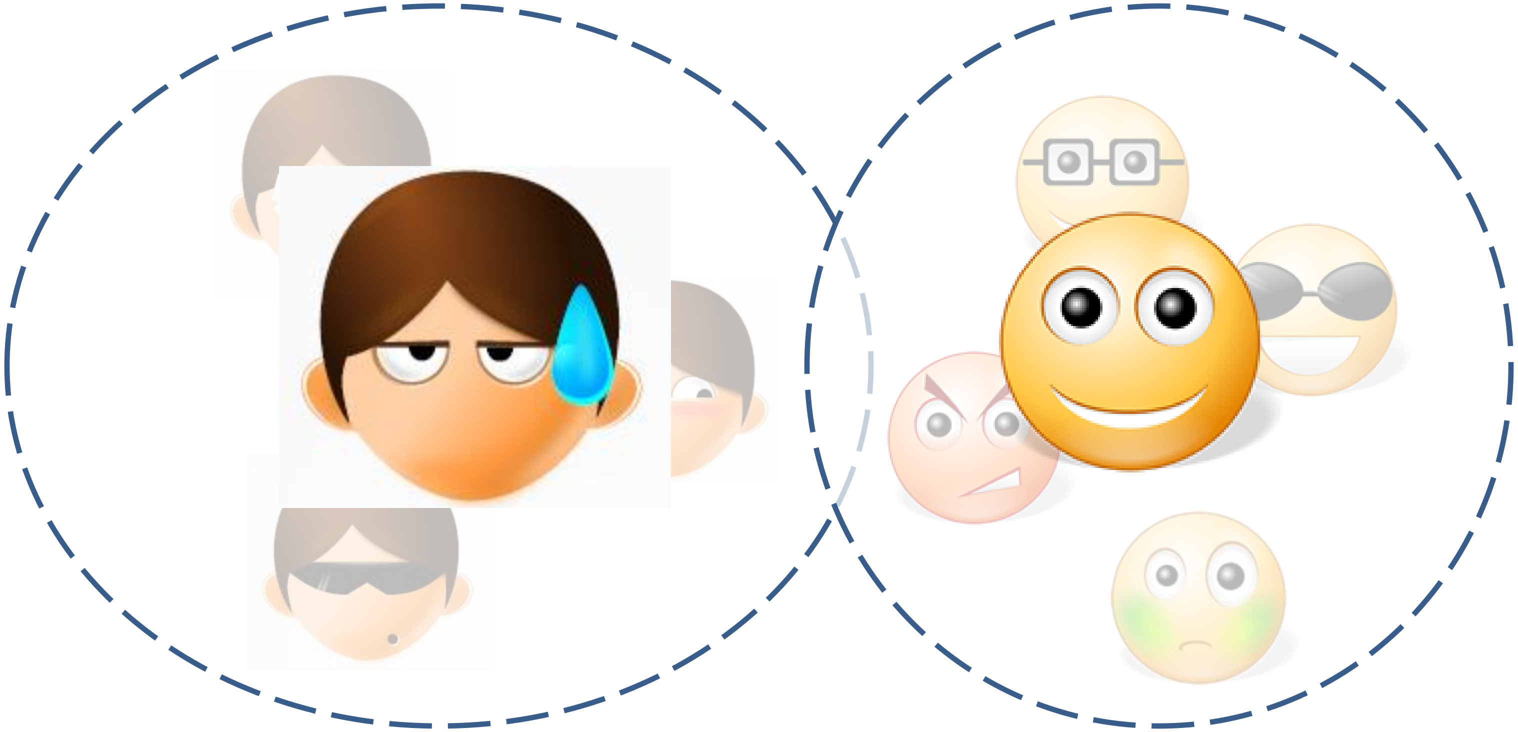
Deduplication Problem Statement

- Cluster the records/mentions that correspond to same entity



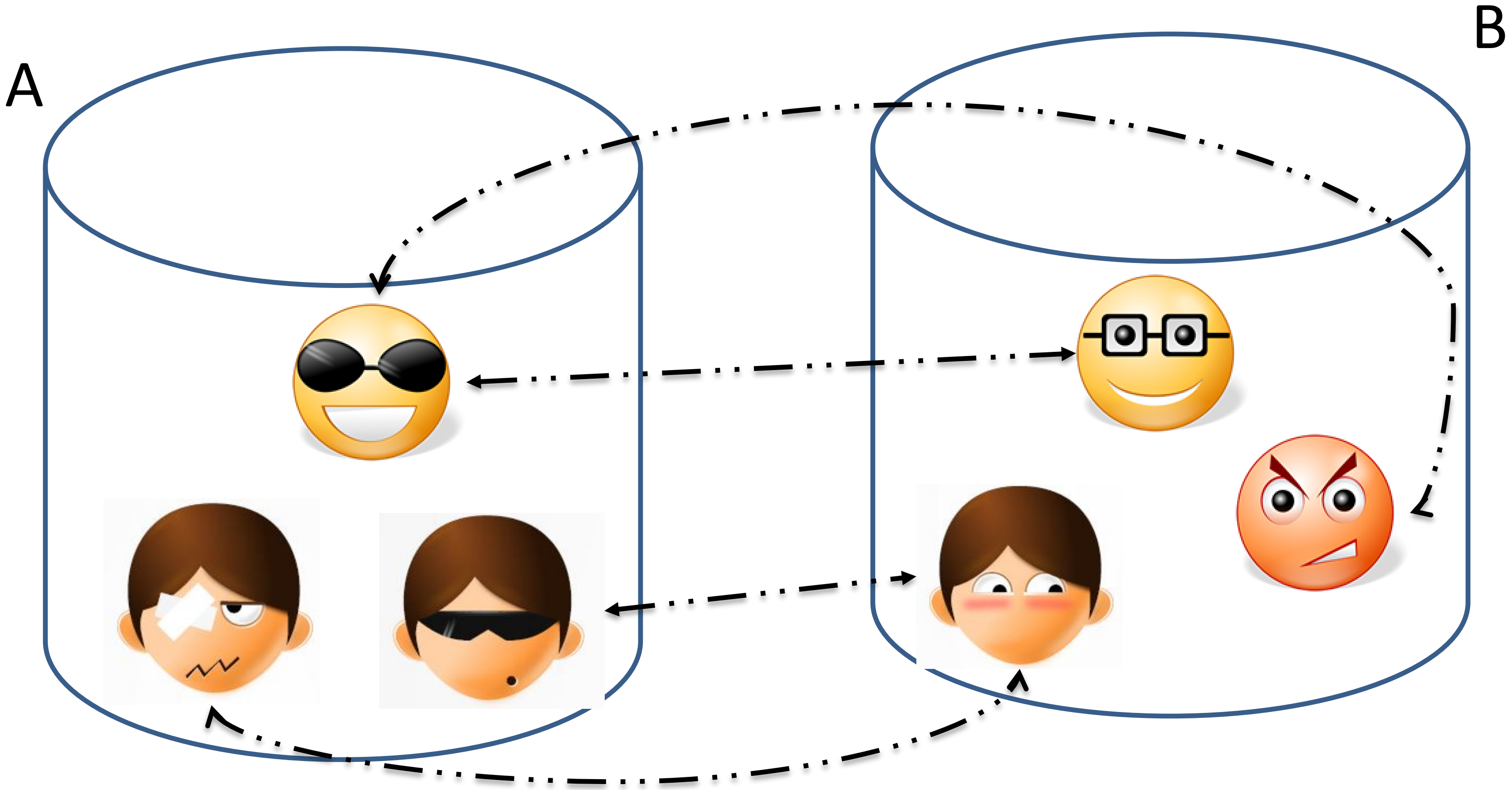
Deduplication Problem Statement

- Cluster the records/mentions that correspond to same entity
  - **Intensional Variant**: Compute cluster representative



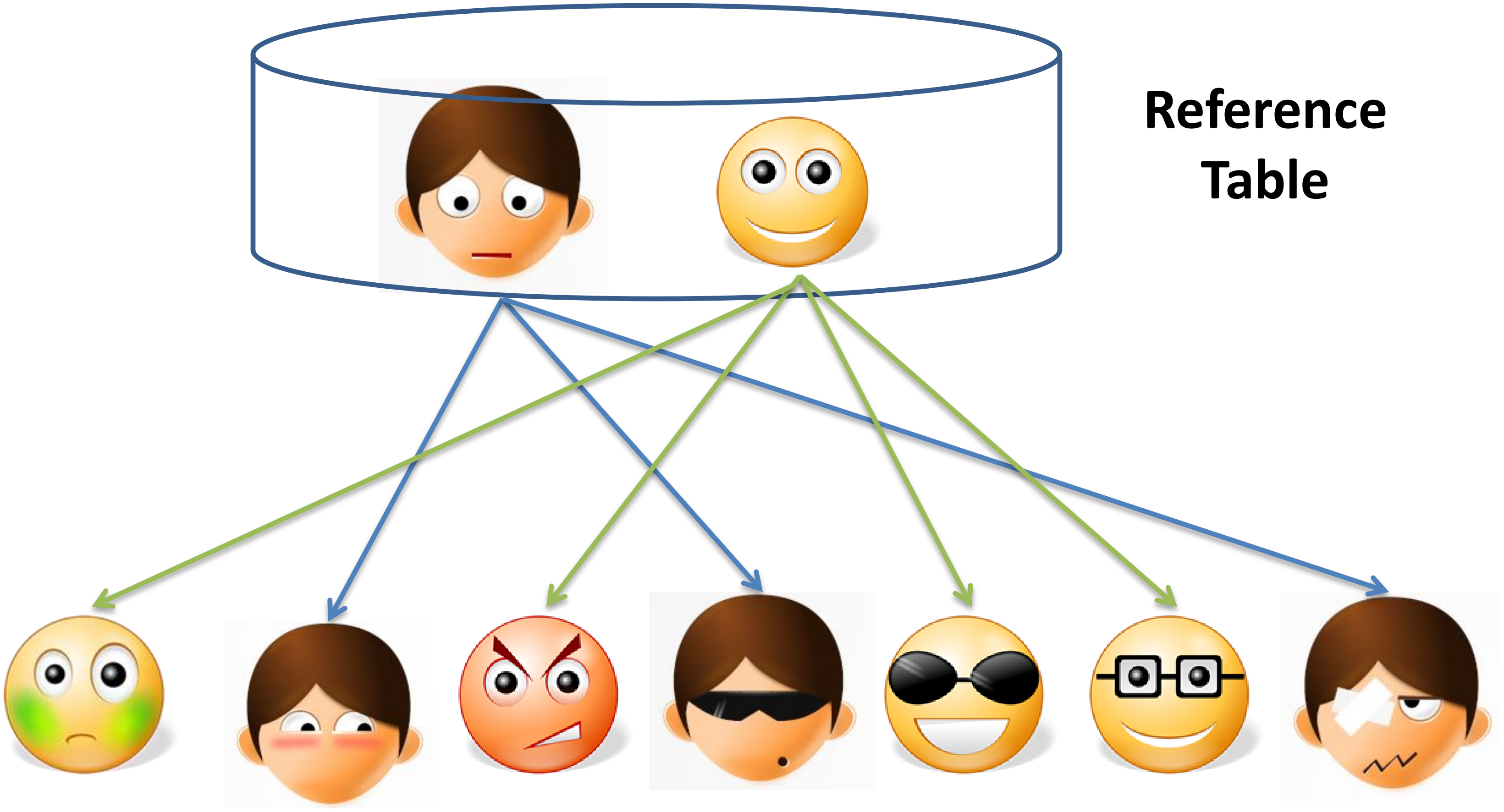
Record Linkage Problem Statement

- Link records that match across databases



Reference Matching Problem

- Match noisy records to clean records in a reference table



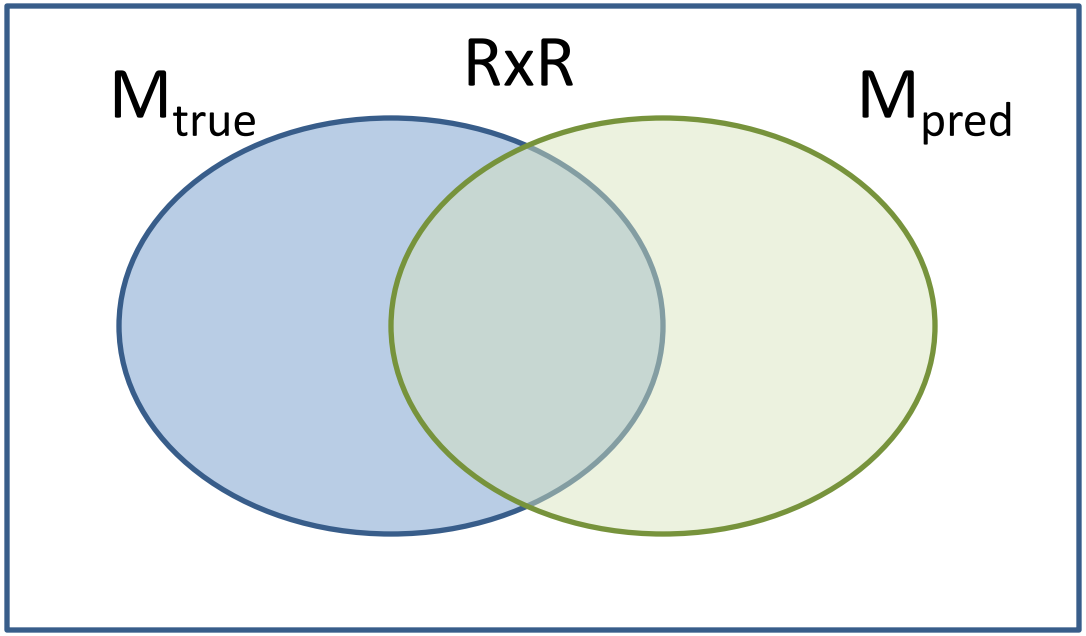
Notation & Assumptions

- *R*: set of records / mentions
- *M*: set of *matches* (record pairs that correspond to same entity)
- *N*: set of *non-matches* (record pairs corresponding to different entities)
- *E*: set of entities

- True (M_{true}, N_{true}, E_{true}): according to real world
  vs Predicted (M_{pred}, N_{pred}, E_{pred}): by algorithm

Relationship between M_{true} and M_{pred}

- M_{true} (SameAs , Equivalence)
- M_{pred} (Similar representations and similar attributes)



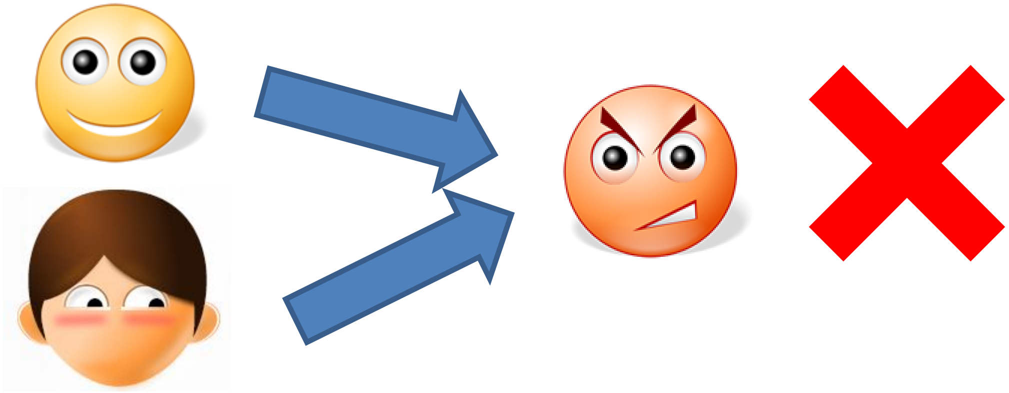
Metrics

- Pairwise metrics
  - Precision/Recall, F1
  - # of predicted matching pairs

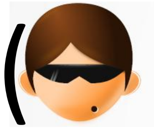
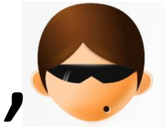
- Cluster level metrics
  - purity, completeness, complexity
  - Precision/Recall/F1: Cluster-level, closest cluster, MUC, B^3 , Rand Index
  - Generalized merge distance [Menestrina et al, PVLDB10]

Typical Assumptions Made

- *Each record/mention is associated with a single real world entity.*



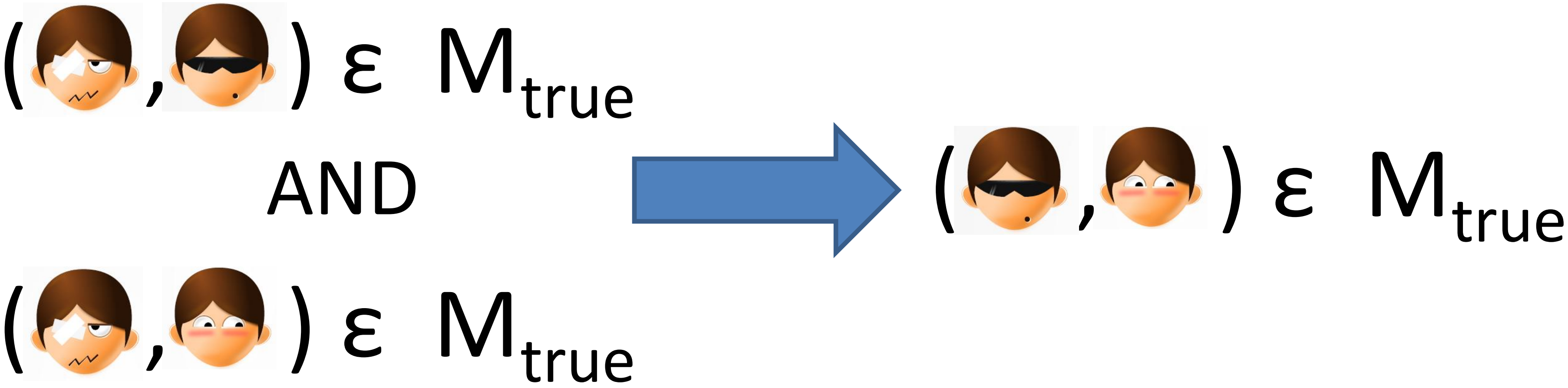
- *In record linkage, no duplicates in the same source*
- *If two records/mentions are identical, then they are true matches*

( , ) ε M_{true}

ER versus Classification

Finding matches vs non-matches is a classification problem

- Imbalanced: typically O(R) matches, O(R^2) non-matches
- Instances are pairs of records. Pairs are not IID



(,) ε M_{true}

AND

(,) ε M_{true}

(,) ε M_{true}

ER vs Clustering

Computing entities from records is a clustering problem

- In typical clustering algorithms (k-means, LDA, etc.) *number of clusters is a constant or sub linear in R.*

- In ER: *number of clusters is linear in R, and average cluster size is a constant. Significant fraction of clusters are singletons.*

PART 1-b

DATA PREPARATION & MATCH FEATURES

Normalization

- Schema normalization
  - Schema Matching – e.g., contact number and phone number
  - Compound attributes – full address vs str,city,state,zip
  - Nested attributes
    - List of features in one dataset (air cond[illegible] each feature a boolean attribute
  - Set valued attributes
    - Set of phones vs [illegible] phone
  - Record seg[illegible]
- Data nor[illegible]
  - Often co[illegible] all lower/all upper; remove whitespace
  - detecting and correcting values that contain known typographical errors or variations,
  - expanding abbreviations and replacing them with standard forms; replacing nicknames with their proper name forms
  - Usually done based on dictionaries (e.g., commercial dictionaries, postal addresses, etc.)

Initial data prep big part of the work; smart normalization can go long way!

Matching Functions

- For two references x and y, compute a “comparison” vector, typically similarity of each component attribute.

- Distance metric:
  - Idempotent
  - Non-negative
  - Symmetric
  - Triangle inequality
- Not all commonly used ER distance functions are metrics
  - non-linear elastic matching (NEM)

- From distance, can convert to similarity:
  - S = 1 / d, or if d is normalized, s = 1-d

Summary of Matching Functions

Handle Typographical errors

- Equality on a boolean predicate
- Edit distance
  - Levenstein, Smith-Waterman, Affine
- Set similarity
  - Jaccard, Dice
- Vector Based
  - Cosine similarity, TFIDF

Good for Text like reviews/ tweets

Good for Names

- Alignment-based or Two-tiered
  - Jaro-Winkler, Soft-TFIDF, Monge-Elkan
- Phonetic Similarity
  - Soundex
- Translation-based
- Numeric distance between values
- Domain-specific

Useful for abbreviations, alternate names.

- Useful packages:
  - SecondString, http://secondstring.sourceforge.net/
  - Simmetrics: http://sourceforge.net/projects/simmetrics/
  - LingPipe, http://alias-i.com/lingpipe/index.html

PART 1-c

ALGORITHMS FOR SINGLE-ENTITY ER

Matching Algorithms

- Pairwise Matching
  - Given a vector of comparison scores, Independently compute a (probability) score indicative of whether a pair of records/mentions match.

- Record Linkage
  - Each record from one database matches at most one record from other database.
  - Weighted k-partite matching

- Deduplication
  - Transitivity constraints must be satisfied.
  - Correlation Clustering

PAIRWISE MATCHING

Pairwise Match Score

Problem: Given a vector of component-wise similarities for a pair of records (x,y), compute P(x and y match).

Solutions:

1. Weighted sum or average of component-wise similarity scores. Threshold determines match or non-match.
   - 0.5*Last-name-match-score + 0.2*address-match-score + 0.3*login-match-score.
   - Hard to pick weights.
     - Match on last name match *more predictive* than login name.
     - Match on “Smith” *less predictive* than match on “Getoor”.
   - Hard to tune a threshold.

Pairwise Match Score

Problem: Given a vector of component-wise similarities for a pair of records (x,y), compute P(x and y match).

Solutions:

1. Weighted sum or average of component-wise similarity scores. Threshold determines match or non-match.
2. Formulate rules about what constitutes a match.
   - (Last-name-score > 0.7 AND address-match-score > 0.8)
     OR (login-match-score > 0.9 AND address-match-score > 0.9)
   - Manually formulating the right set of rules is hard.

ML Pairwise Approaches

- Supervised machine learning algorithms
  - Decision trees
    - [Cochinwala et al, IS01]
  - Support vector machines
    - [Bilenko & Mooney, KDD03]; [Christen, KDD08]
  - Ensembles of classifiers
    - [Chen et al., SIGMOD09]
  - Conditional Random Fields (CRF)
    - [Wellner & McCallum, NIPS04]

- Issues:
  - **Training set generation**
  - Imbalanced classes – many more negatives than positives (even after eliminating obvious non-matches … using *Blocking*)
  - Misclassification cost

Creating a Training Set is a key issue

- Constructing a training set is hard – since most pairs of records are "easy non-matches".
  - 100 records from 100 cities.
  - Only 10^6 pairs out of total 10^8 (1%) come from the same city
- Some pairs are hard to judge even by humans
  - Inherently ambiguous
    - E.g., Paris Hilton (person or business)
  - Missing attributes
    - Starbucks, Toronto vs Starbucks, Queen Street ,Toronto

Avoiding Training Set Generation

- Unsupervised / Semi-supervised Techniques
  - Fellegi-Sunter Model
    - [Newcombe et al Science ‘59, Fellegi & Sunter JASA 69, Winkler ‘06, Herzog et al ’07]
  - Generative Models
    - [Ravikumar & Cohen, UAI04]

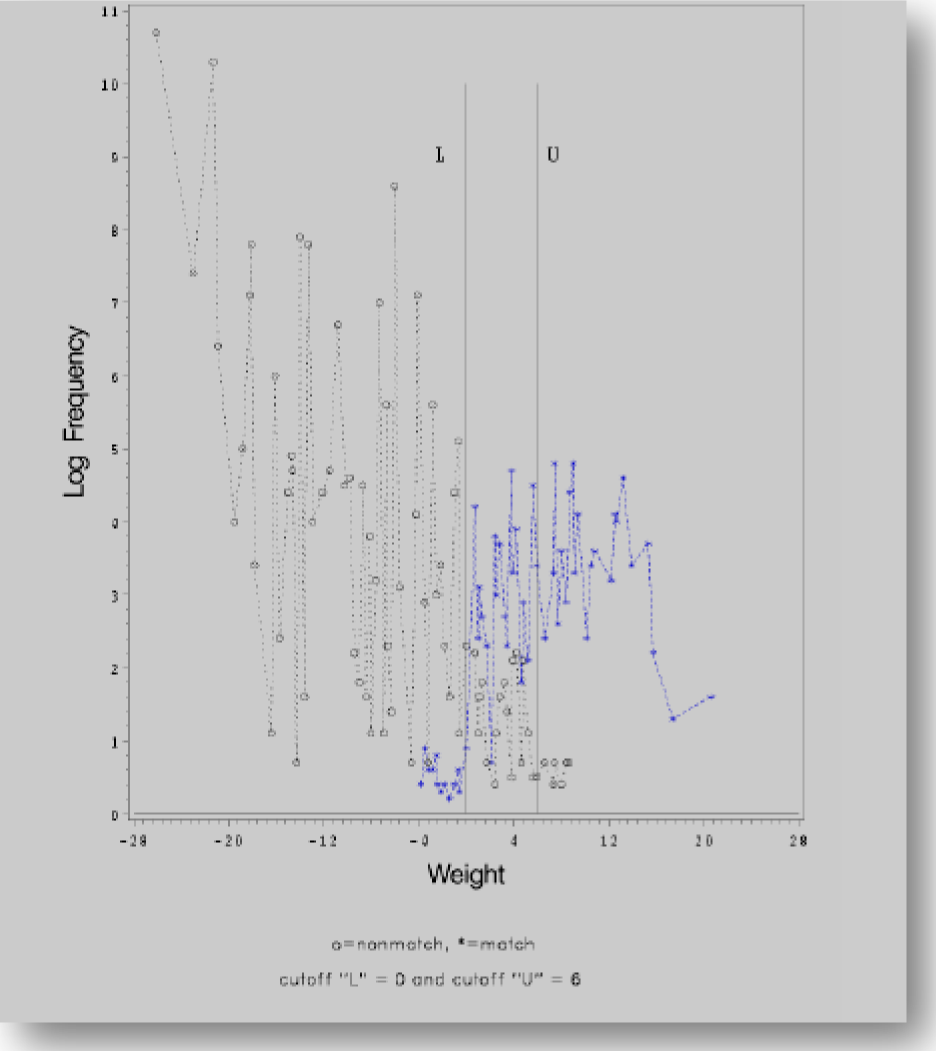
Fellegi & Sunter Model

- *r = (x,y)* is record pair, γ is comparison vector, *M* matches, *U* non-matches
- In the original work, γ is binary, 0/1, match/not match
- Decision rule $$R = \frac{P(\gamma \mid r \in M)}{P(\gamma \mid r \in U)}$$

$$R \geq t_u \Rightarrow r \rightarrow \mathrm{Match}$$
$$t_l < R < t_u \Rightarrow r \rightarrow \mathrm{Potential\ Match}$$
$$R \leq t_u \Rightarrow r \rightarrow \mathrm{Non\text{-}Match}$$

- Thresholds t_u and t_l determined by apriori bounds on false matches and false non-matches

Fellegi & Sunter Model



[Winkler 2006]

Computing Probabilities

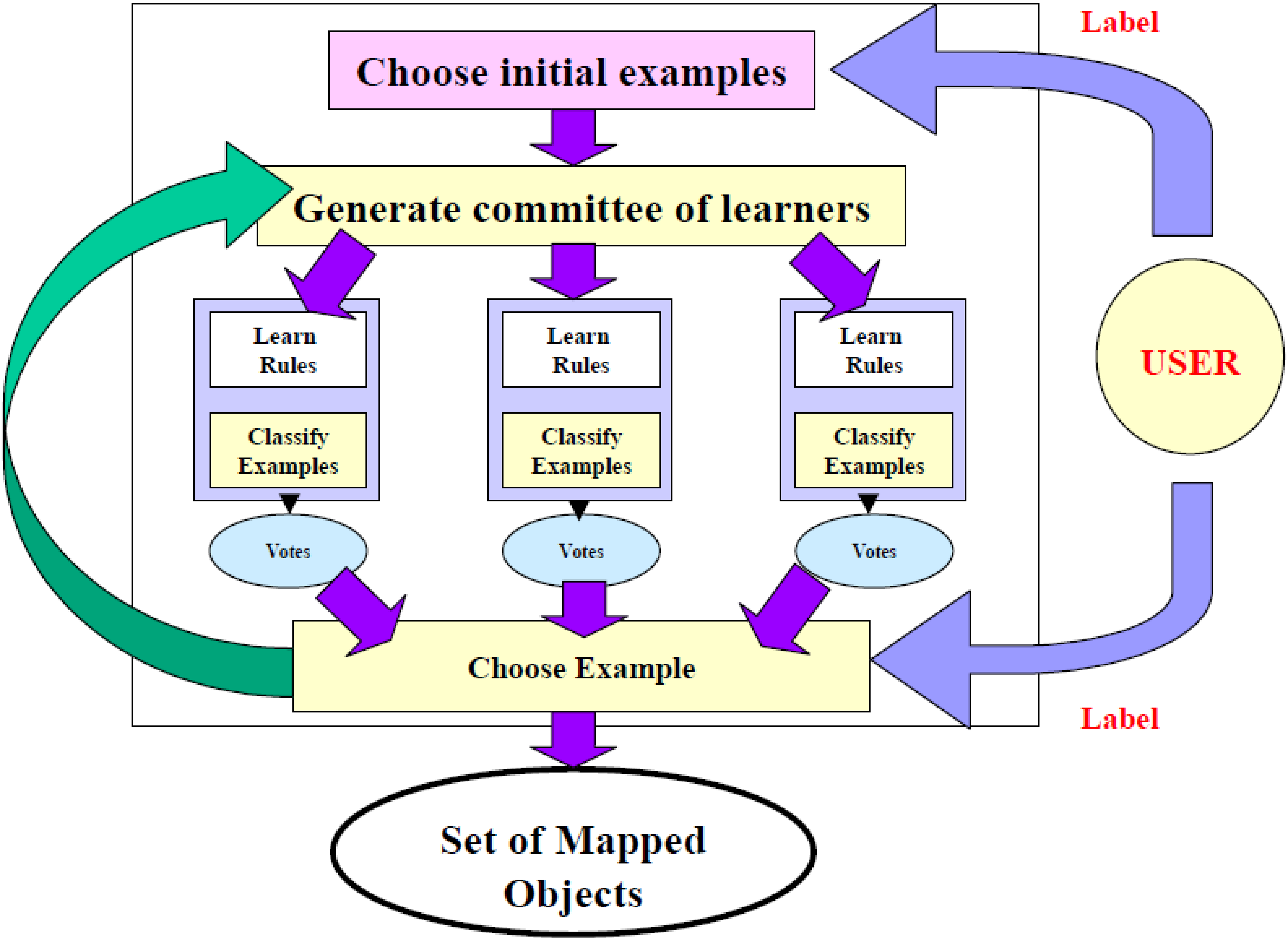
- Typically make an independence assumption
- Agreement weight w_i is calculated for each attribute i based on m and u probabilities:
  - $m_i = P(x_i = y_i \mid r \in M)$
  - $u_i = P(x_i = y_i \mid r \in U)$
- Probabilities can be estimated using EM
  - See [Winkler 2006] for a survey of techniques used in the US Census.

Avoiding Training Set Generation

- Unsupervised / Semi-supervised Techniques
  - Fellegi-Sunter Model
    - [Newcombe Science '59, Fellegi & Sunter JASS 69, Winkler '99, Herzog et al '07]
  - Generative Models
    - [Ravikumar & Cohen, UAI04]

- Active Learning
  - Committee of Classifiers
    - [Sarawagi et al KDD '00, Tajeda et al IS '01]
  - Provably optimizing precision/recall
    - [Arasu et al SIGMOD '10, Bellare et al KDD '12]

Committee of Classifiers [Tejada et al, IS '01]



Active Learning with Provable Guarantees

- Most active learning techniques minimize 0-1 loss

  [Beygelzimer et al NIPS 2010].

$$\text{minimize} \quad \frac{fn(h) + fp(h)}{n}$$

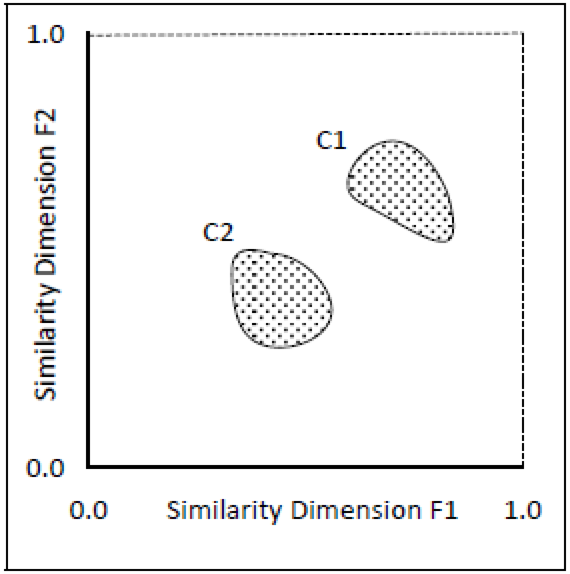
- However, ER is very imbalanced:
  - Number of non-matches >> number of matches.
  - Classifying all pairs as “non-matches” has low 0-1 loss.

- Hence, need active learning techniques that minimize precision/recall.

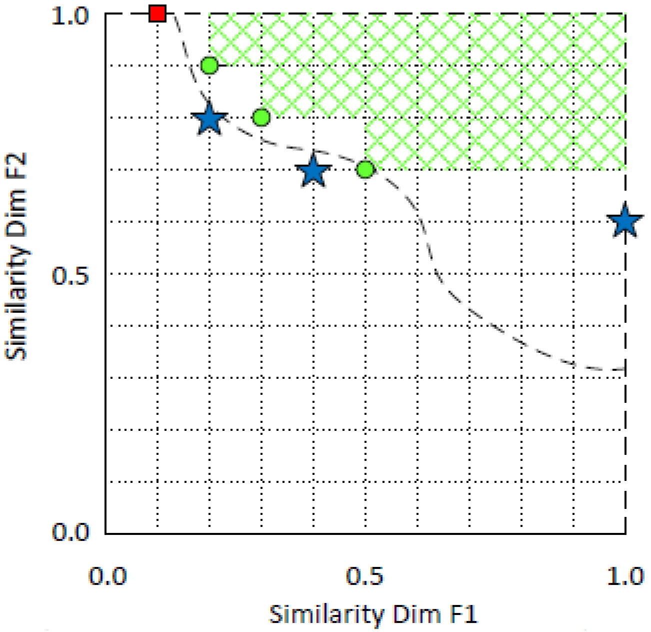
$$\begin{array}{ll} \text{maximize} & recall(h) \\ \text{subject to} & precison(h) \geq \tau \end{array}$$

Active Learning with Provable Guarantees

- Monotonicity of Precision [Arasu et al SIGMOD '10]



There is a larger fraction of matches in C1 than in C2.



Algorithm searches for the optimal classifier using binary search on each dimension

Active Learning with Provable Guarantees

[Bellare et al KDD ‘12]

O ($\log^2$ n) calls to a blackbox 0-1 loss active learning algorithm.

Exponentially smaller label complexity than [Arasu et al SIGMOD ‘10] (in the worst case).

1. Precision Constrained → Weighted 0-1 Loss Problem (using a Lagrange Multiplier λ).
2. Given a fixed value for λ, weighted 0-1 Loss can be optimized by a balckbox active learning classifier.
3. Right value of λ is computed by searching over all optimal classifiers.
   - Classifiers are embedded in a 2-d plane (precision/recall)
   - Search is along the convex hull of the embedded classifiers

Open challenge

- Handling errors in human judgements:
  - In an experiment on Amazon Mechanical Turk:
    - Each pairwise judgment given to 5 different people
  - Majority of workers agreed on truth on only 90% of pairwise judgements.

Using pairwise ER

- ER applications need more than independent classification of pairs of records as match/non-match.

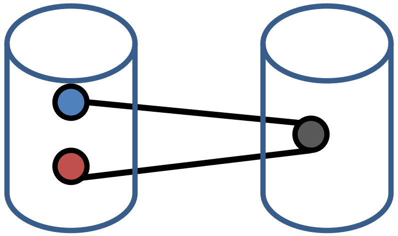
- Record Linkage
- Deduplication

RECORD LINKAGE

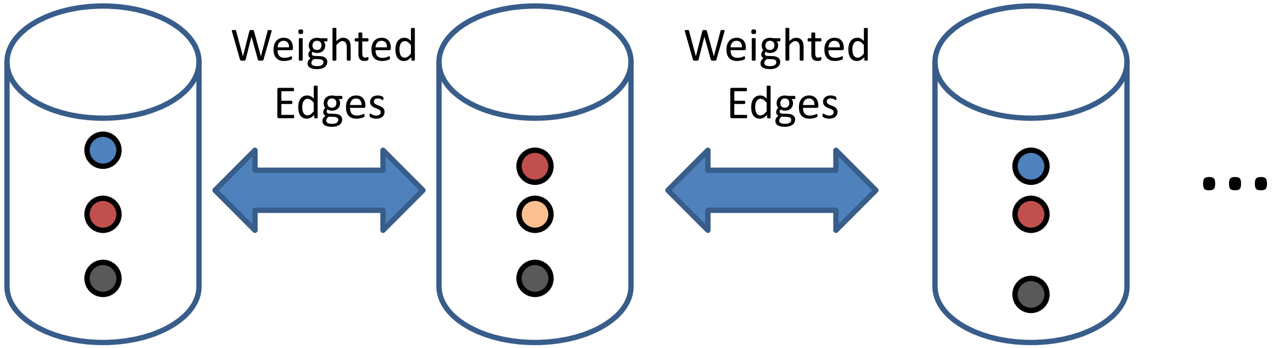
1-1 assumption

- Matching between (almost) deduplicated databases.
- Each record in one database matches at most one record in another database.

- Pairwise ER may match a record in one database with more than one record in second database

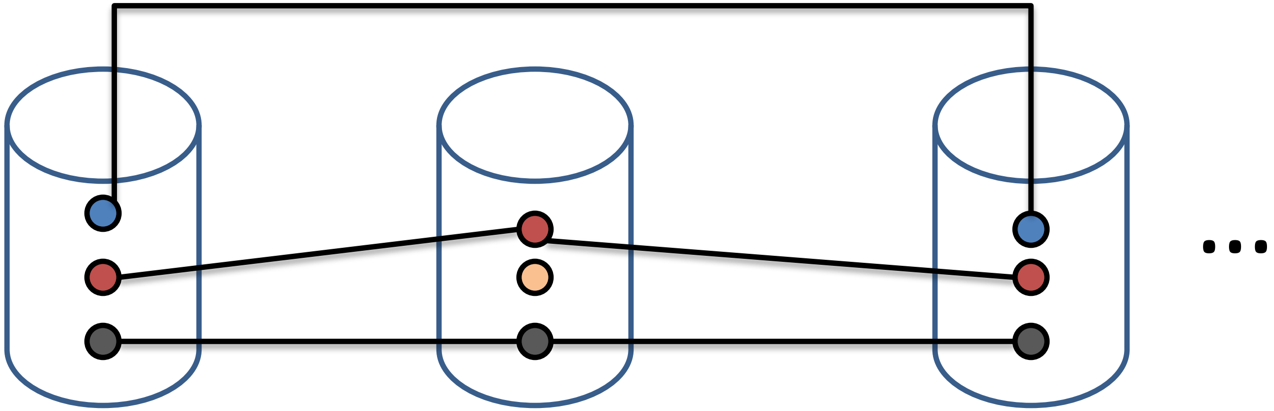


Weighted K-Partite Matching



- Edges between pairs of records from different databases
- Edge weights
  - Pairwise match score
  - Log odds of matching

Weighted K-Partite Matching

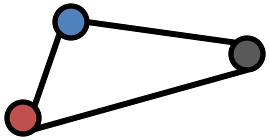


- Find a matching (each record matches at most one other record from other database) that maximize the sum of weights.
- General problem is NP-hard (3D matching)
- Successive bipartite matching is typically used. [Gupta & Sarawagi, VLDB ‘09]

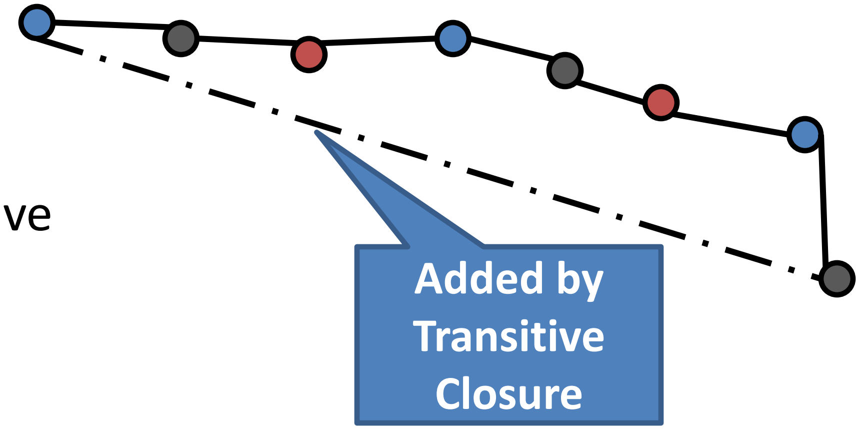
DEDUPLICATION

Deduplication => Transitivity

- Often pairwise ER algorithm output “inconsistent” results
  - (x, y) ε M_{pred} , (y,z) ε M_{pred} , but (y,z) $\not\varepsilon$ M_{pred}



- Idea: Correct this by adding additional matches using transitive closure



- In certain cases, this is a bad idea.
  - Graphs resulting from pairwise ER have diameter > 20
    [Rastogi et al Corr‘12]

- Need clustering solutions that deal with this problem directly by reasoning about records jointly.

Clustering-based ER

- Resolution decisions are not made independently for each pair of records

- Based on variety of clustering algorithms, but
  - Number of clusters unknown aprioiri
  - Many, many small (possibly singleton) clusters

- Often take a pair-wise similarity graph as input

- May require the construction of a *cluster representative* or *canonical entity*

Clustering Methods for ER

- Hierarchical Clustering
  - [Bilenko et al, ICDM 05]
- Nearest Neighbor based methods
  - [Chaudhuri et al, ICDE 05]
- **Correlation Clustering**
  - [Soon et al CL’01, Bansal et al ML’04, Ng et al ACL’02, Ailon et al JACM’08, Elsner et al ACL’08, Elsner et al ILP-NLP’09]

Integer Linear Programming view of ER

- $r_{xy}\ \varepsilon\ \{0,1\}$, $r_{xy} = 1$ if records x and y are in the same cluster.
- $w^{+}_{xy}\ \varepsilon\ [0,1]$, cost of clustering x and y together
- $w^{-}_{xy}\ \varepsilon\ [0,1]$, cost of placing x and y in different clusters

$$minimize \sum r_{xy} w^{+}_{xy} + (1 - r_{xy}) w^{-}_{xy}$$

$$s.t.\ \forall\, x, y, z \in R,$$

$$r_{xy} + r_{xz} + r_{yz} \neq 2$$

Transitive closure

Correlation Clustering

$$minimize \sum r_{xy} w^{+}_{xy} + (1 - r_{xy}) w^{-}_{xy}$$

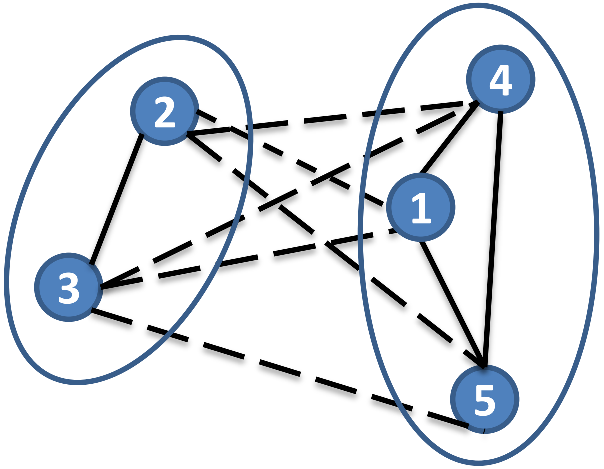
$$s.t. \; \forall \, x, y, z \in R,$$

$$r_{xy} + r_{xz} + r_{yz} \neq 2$$

- Cluster mentions such that total cost is minimized

  Solid edges contribute w^{+}_{xy} to the objective

  Dashed edges contribute w^{-}_{xy} to the objective



- Cost based on pairwise similarities

$$\{p_{xy} \mid \forall \, (x, y) \in R \times R\}$$

  - Additive: $w^{+}_{xy} = p_{xy}$ and $w^{-}_{xy} = (1\text{-}p_{xy})$
  - Logarithmic: $w^{+}_{xy} = \log(p_{xy})$ and $w^{-}_{xy} = \log(1\text{-}p_{xy})$

Correlation Clustering

- Solving the ILP is NP-hard [Ailon et al 2008 JACM]

- A number of heuristics [Elsner et al 2009 ILP-NLP]
  - Greedy BEST/FIRST/VOTE algorithms
  - Greedy PIVOT algorithm (5-approximation)
  - Local Search

Greedy Algorithms

Step 1: Permute the nodes according a random π

Step 2: Assign record *x* to the cluster that maximizes *Quality*

Start a new cluster if *Quality* < 0

Quality:

- BEST: Cluster containing the closest match $max_{y \in C} \; w_{xy}^{+}$
  - [Ng et al 2002 ACL]
- FIRST: Cluster contains the most recent vertex y with $w^{+}_{xy} > 0$
  - [Soon et al 2001 CL]
- VOTE: Assign to cluster that minimizes objective function.
  - [Elsner et al 08 ACL]

Practical Note:

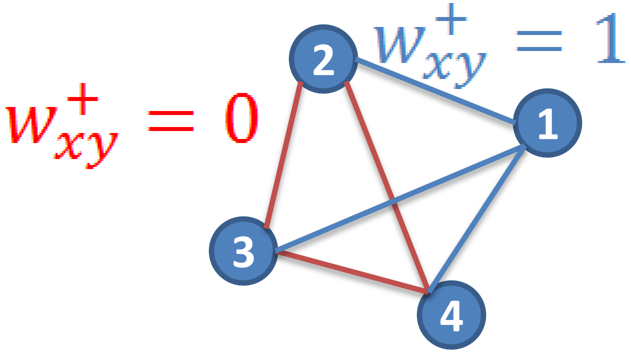
- Run the algorithm for many random permutations , and pick the clustering with best objective value (better than average run)

Greedy with approximation guarantees

PIVOT Algorithm [Ailon et al 2008 JACM]

- Pick a random *(pivot)* record *p*.
- New cluster = $\{x \mid w^+_{px} > 0\}$

- π = {1,2,3,4} C = {{1,2,3,4}}
- π = {2,4,1,3} C = {{1,2}, {4}, {3}}
- π = {3,2,4,1} C = {{1,3}, {2}, {4}}



When weights are 0/1, E(cost(greedy)) < **3** OPT

For $w^+_{xy} + w^-_{xy} = 1$, E(cost(greedy)) < **5** OPT

[Elsner et al, ILP-NLP '09] : Comparison of various correlation clustering algorithms

Summary of Single-Entity ER Algorithms

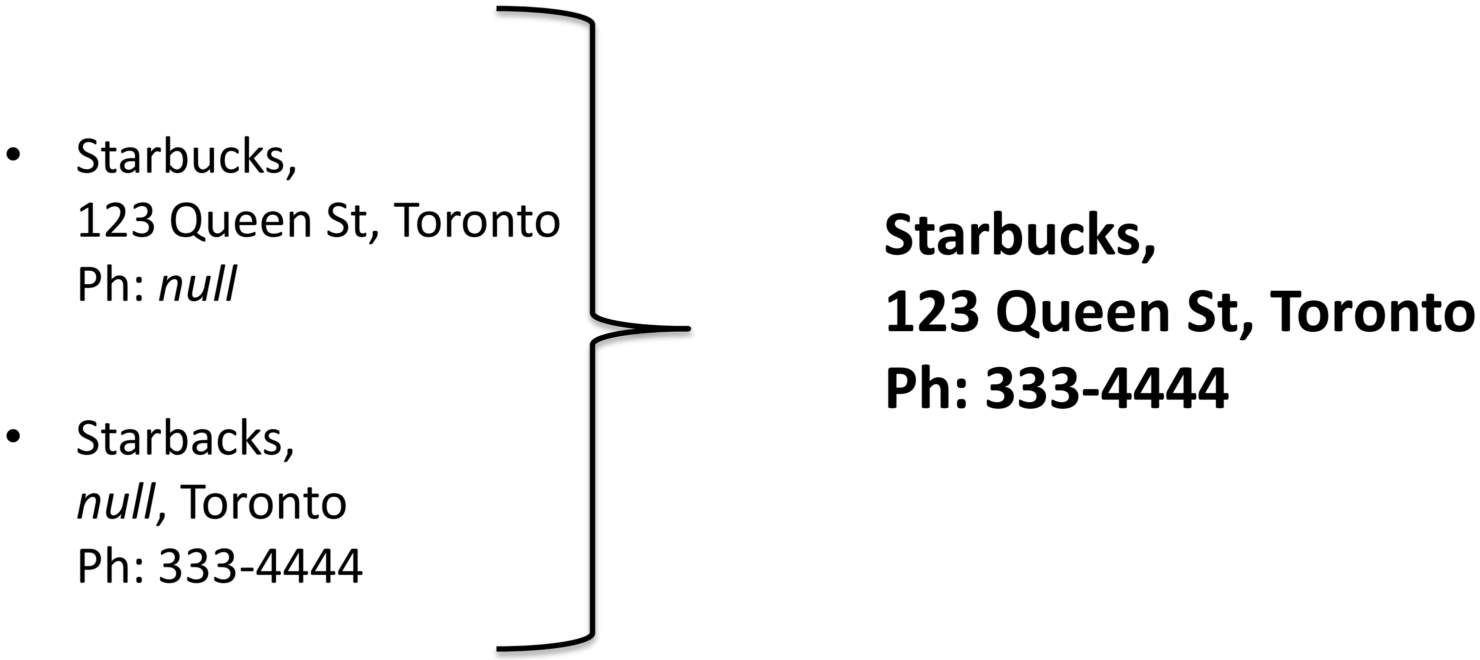
- Many algorithms for independent classification of pairs of records as match/non-match
  - ML based classification & Fellegi-Sunter
  - **Pro: Advanced state of the art**
  - **Con: Building a training set is an open problem**
  - Active learning is becoming popular
- ER applications need more than pairwise classification
  - **Record linkage**: each record matched to at most one record from other database.
    - **Weighted K-Partite Matching**
  - **Deduplication**: transitivity requires clustering based algorithms.
    - **Correlation Clustering**

PART 1-d

CANONICALIZATION

Canonicalization

- Merge information from duplicate mentions to construct a cluster representative with *maximal* information



- Starbucks,
  123 Queen St, Toronto
  Ph: *null*

- Starbacks,
  null, Toronto
  Ph: 333-4444

Starbucks,
123 Queen St, Toronto
Ph: 333-4444

Canonicalization

- Critically important in Web portals where users must be shown a consolidated view of the duplicate cluster.
  - Each mention only contains a subset of the attributes.
  - Mentions contain variations (of names, addresses).
  - Some of the mentions can have wrong values.

Canonicalization Algorithms

- Rule based:
  - For names: typically longest names are used.
  - For set values attributes: UNION is used.

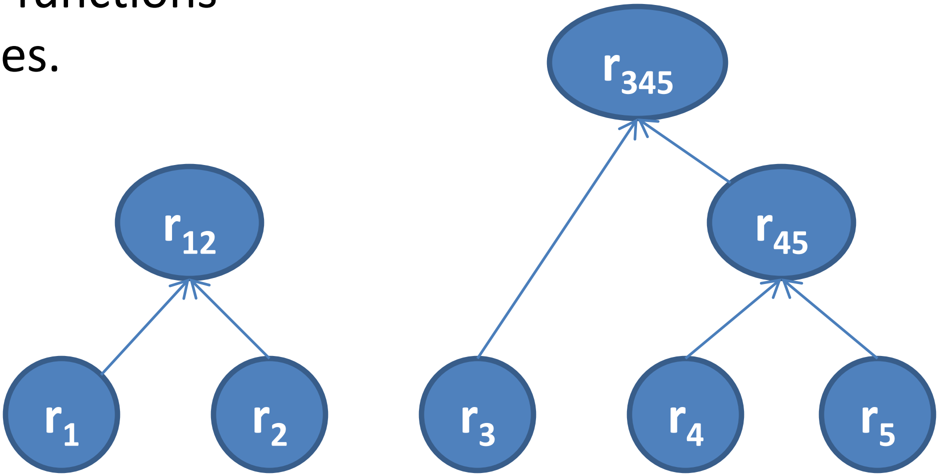
- For strings, [Culotta et al KDD07] learn an edit distance for finding the most representative “centroid”.

- Can use “majority rule” to fix errors
  (if 4 out of 5 say a business is closed, then business is closed).
  - **This may not always work due to copying [Dong et al VLDB09], or when underlying data changes [Pal et al WWW11]**

Canonicalization for Efficiency

- Stanford Entity Resolution Framework [Benjelloun VLDBJ09]
  - Consider a blackbox match and merge function
  - Match is a pairwise boolean operator
  - Merge: construct canonical version of a matching pair

- Can minimize time to compute matches by interleaving matching and merging
  - esp., when match and merge functions satisfy **monotonicity** properties.



Summary of Canonicalization

- Critically important in Web portals where users must be shown a consolidated view of the duplicate cluster.

- Canonicalization can also help speed up ER in certain cases.

PART 2

RELATIONAL & MULTIENTITY ER

Outline

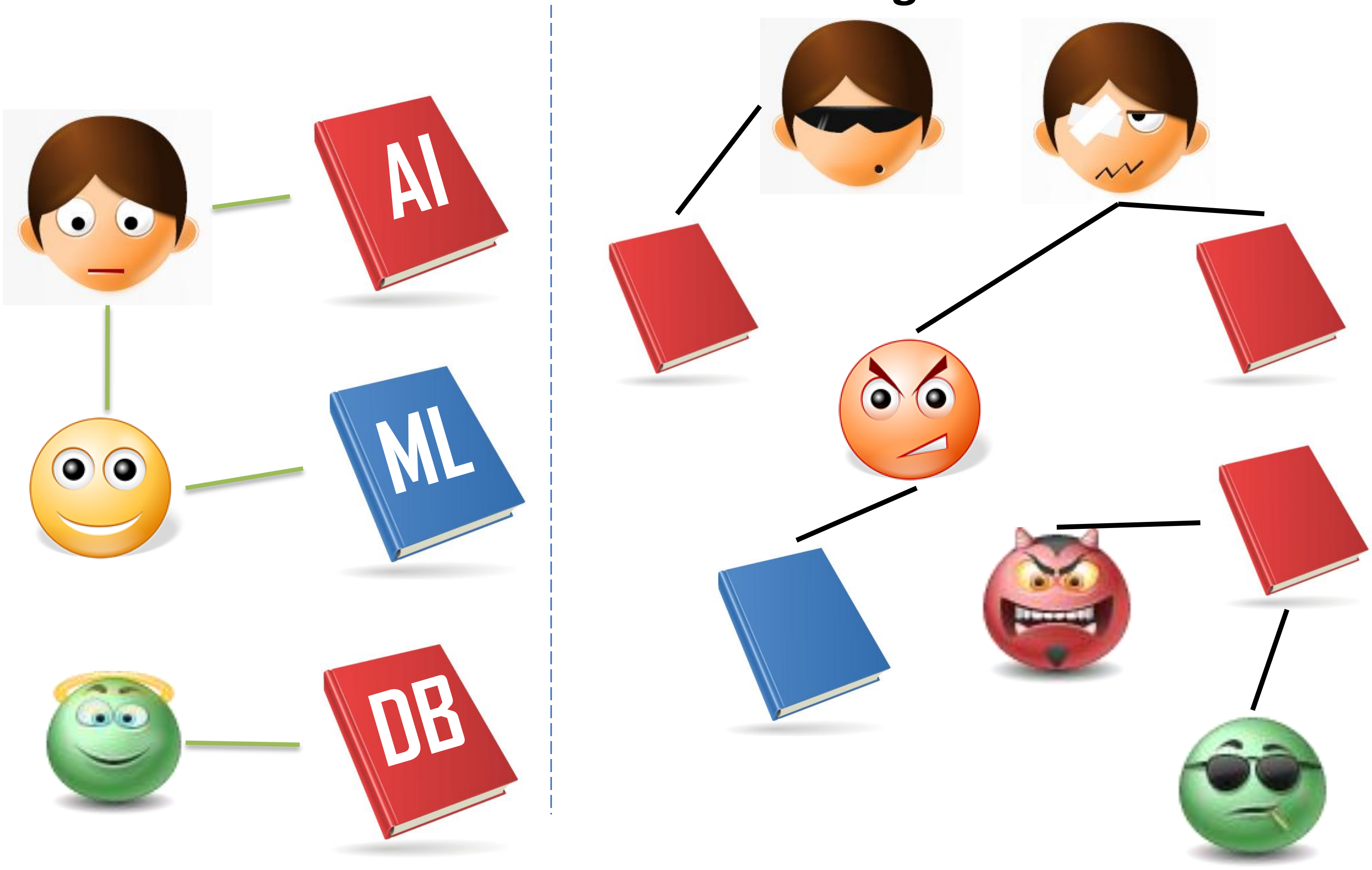
1. Classical Single Entity ER
2. Relational & MultiEntity ER
    a) Problem Statement
    b) Relational Features & Constraints
    c) Non-Probabilistic Approaches: Similarity Propagation
    d) Non-Probabilistic Approaches: Constraint Optimization
    e) Probabilistic Approaches: Generative Models
    f) Probabilistic Approaches: Undirected Models
3. Efficiency: Blocking/Canopy Generation
4. Challenges & Future Directions

PART 2-a

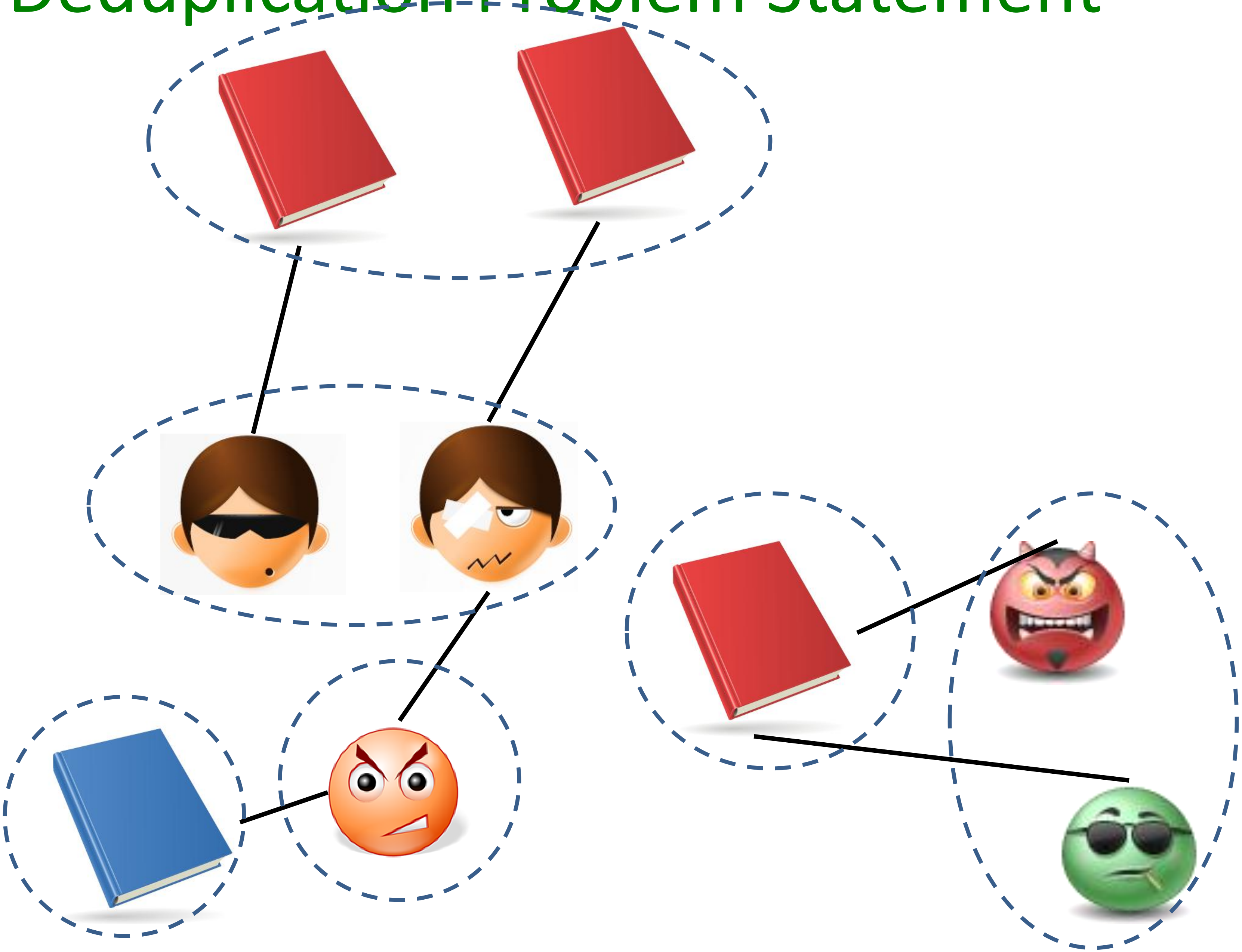
PROBLEM DEFINITION

Abstract Problem Statement

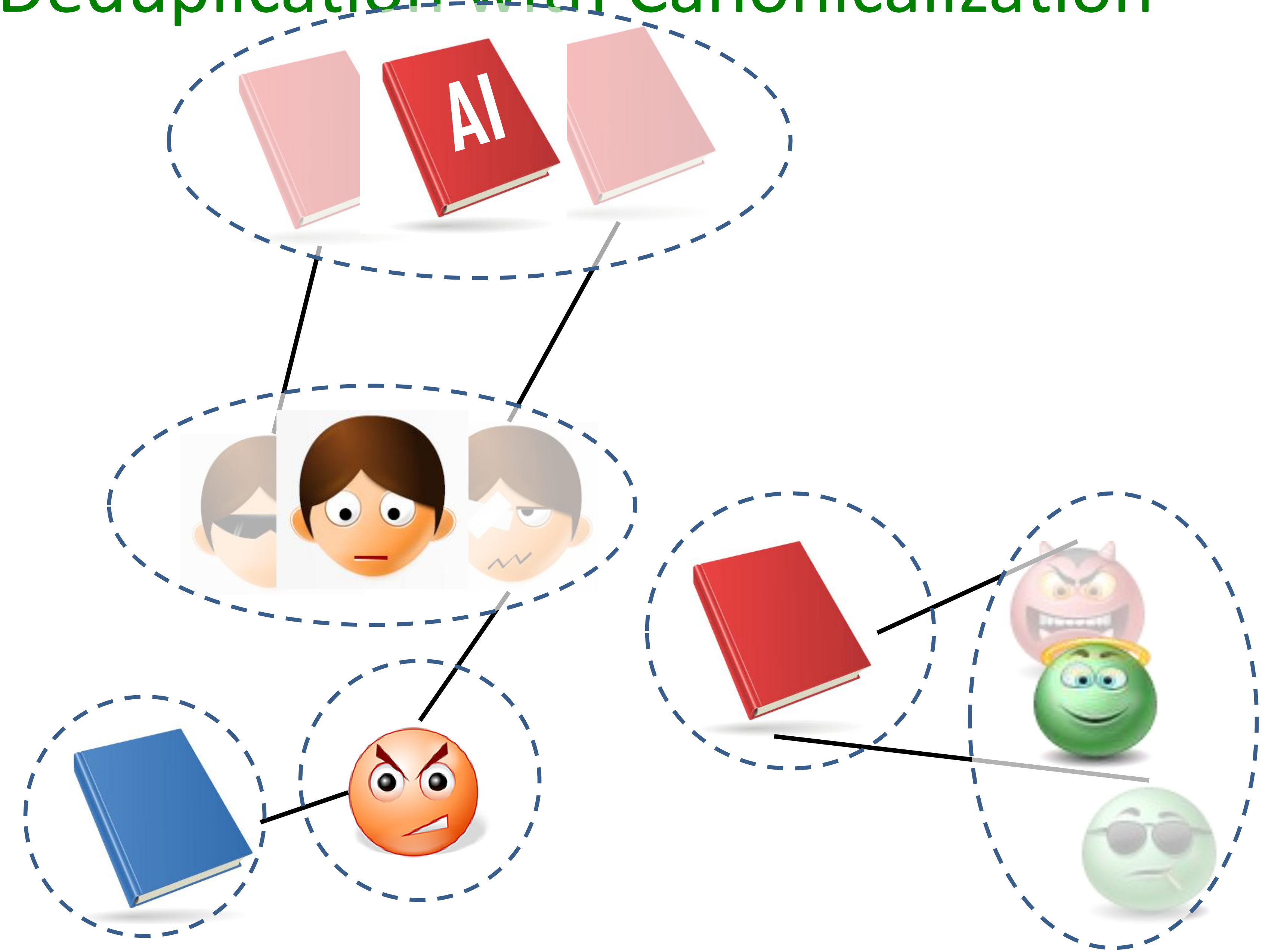
Real World **Digital World**



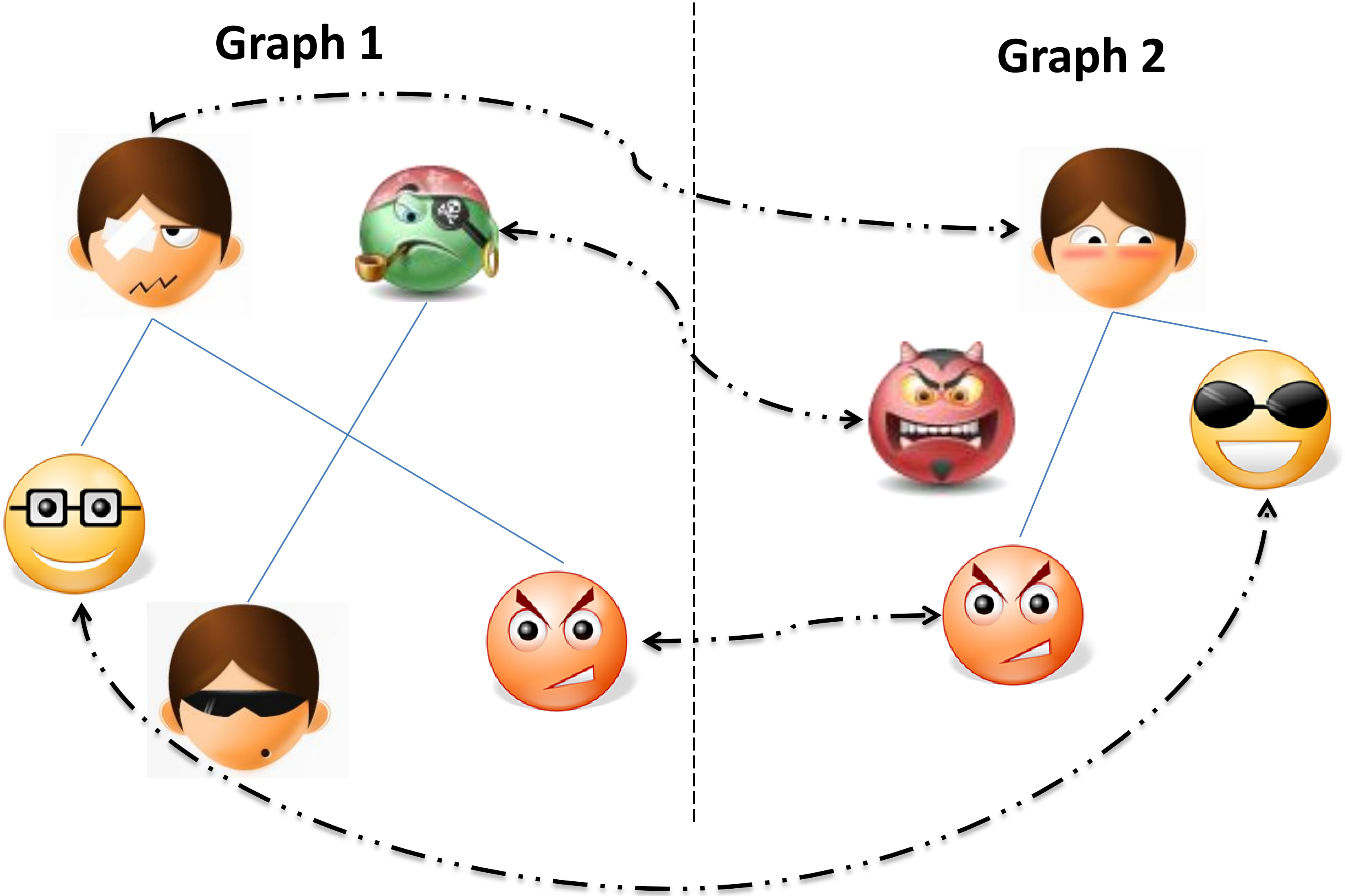
Deduplication Problem Statement



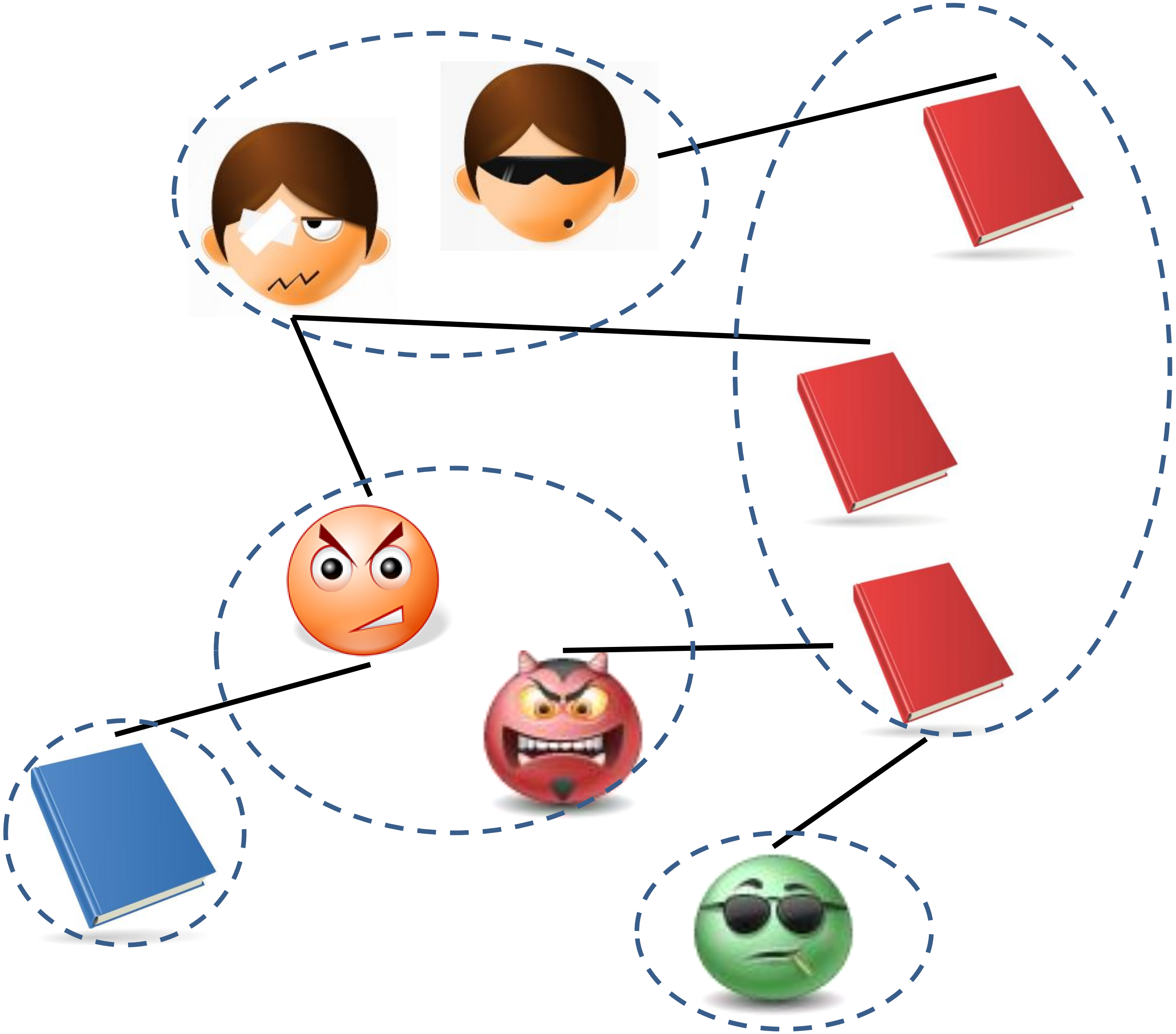
Deduplication with Canonicalization



Graph Alignment (& motif search)



Relationships are crucial



Notation & Assumptions

- R: set of records / mentions (typed)
- H: set of relations / hyperedges (typed)
- M: set of *matches* (record pairs that correspond to same entity)
- N: set of *non-matches* (record pairs corresponding to different entities)
- E: set of entities
- L: set of links

- True (M_{true}, N_{true}, E_{true}, L_{true}): according to real world
  vs Predicted (M_{pred}, N_{pred}, E_{pred}, L_{pred}): by algorithm

Metrics

- Most algorithms use pairwise and cluster-based measures on each entity type
- Little work that evaluations correct prediction of links

MOTIVATING EXAMPLE: BIBLIOGRAPHIC DOMAIN

Bibliography Domain

- Entities:
  - Papers
  - Authors
  - Organizations/Author Affiliations
  - Venues
  - Conference Locations
- Relations:
  - Author-Of
  - Associated-With
  - AppearsIn
  - Cites
- Co-occurrence relationships
  - Co-authors
  - Papers in same conference
  - Papers by same author
  - etc.

PART 2-b

RELATIONAL FEATURES & CONSTRAINTS

Relational Features

- There are a variety of ways of improving ER performance when data is richer than a single table/entity type
- One of the simplest is to use additional information, to *enrich* model with *relational features* that will provide richer context for matching
  - This will often lead to increased precision
    - Relational information can help to distinguish references, add avoid false positives
  - It may also lead to increased recall
    - The best threshold will be different, and it may be, with the additional information, one can get increased recall as well.

Set-based Relational Features

- Relational features are often set-based
  - Set of coauthors for a paper
  - Set of cities in a country
  - Set of products manufactured by manufacturer
- Can use set similarity functions mentioned earlier
  - Common Neighbors: Intersection size
  - Jaccard's Coefficient: Normalize by union size
  - Adar Coefficient: Weighted set similarity
- Can reason about similarity in sets of values
  - Average or Max
  - Other aggregates

Constraints

- In single entity case, we already saw two important forms of constraints:
  - Transitivity: If M1 and M2 match, M2 and M3 match, then M1 and M3 match
  - Exclusivity: If M1 matches with M2, then M3 cannot match with M2

- Transitivity is key to deduplication
- Exclusivity is key to record linkage

Relational Constraints

- In multi-relational domains, matching decisions often propagate
  - Constraints may be hard constraints
    - If M1, M2 match then M3, M4 must match
      - If two papers match, their venues match
      - If two cities match, then their countries match
    - If M1, M2 don't match then M3, M4 cannot match
      - If two venues don't match, then their papers don't match
  - Or soft constraints
    - If M1, M2 match then M3, M4 more likely to match
      - If two venues match, then their papers are more likely to match
    - If M1, M2 don't match then M3, M4 less likely to match
      - If institutions don't match, then authors less likely to match

Terminology

- **Positive evidence:** If M1, M2 match then M3, M4 match
- **Negative evidence:** If M1, M2 match then M3, M4 don’t match

- When matching decisions depend on other matching decisions (in other words, matching decisions are not made independently), we refer to the approach as ***collective***

Match Propagation

- **Global:** In two papers match, then their venues match
  - This constraint can be applied to all instances of venue mentions
    - All occurrences of ‘SIGMOD’ can be matched to ‘International Conference on Management of Data’
- **Local:** If two papers match, then their authors match
  - This constraint can only be applied locally
    - Don’t want to match all occurrences of ‘J. Smith’ with ‘Jeff Smith’, only in the context of the current paper

Additional Relational Constraints

- Constraints can also encode a variety of additional forms of integrity constraints
  - Uniqueness Constraints
    - Mention M1 and M2 must refer to distinct entities
      - Coauthors are distinct
  - Count Constraints
    - Entity A can link to at most N Bs
      - Authors have at most 5 papers at any conference

- Again, these can be either hard or soft constraints

Ex. Semantic Integrity Constraints

Type	Example
Aggregate	C1 = No researcher has published more than five AAAI papers in a year
Subsumption	C2 = If a citation X from DBLP matches a citation Y in a homepage, then each author mentioned in Y matches some author mentioned in X
Neighborhood	C3 = If authors X and Y share similar names and some co-authors, they are likely to match
Incompatible	C4 = No researcher exists who has published in both HCI and numerical analysis
Layout	C5 = If two mentions in the same document share similar names, they are likely to match
Key/Uniqueness	C6 = Mentions in the PC listing of a conference is to different researchers
Ordering	C7 = If two citations match, then their authors will be matched in order
Individual	C8 = The researcher with the name "Mayssam Saria" has fewer than five mentions in DBLP (new graduate student)

[Shen, Li & Doan, AAAI05]

COLLECTIVE APPROACHES

Collective Approaches

- Decisions for cluster-membership depends on other clusters
    - Non-probabilistic approaches
        - Similarity Propagation
        - Constraint Optimization
    - Probabilistic Models
        - Generative Models
        - Undirected Models

PART 2-c

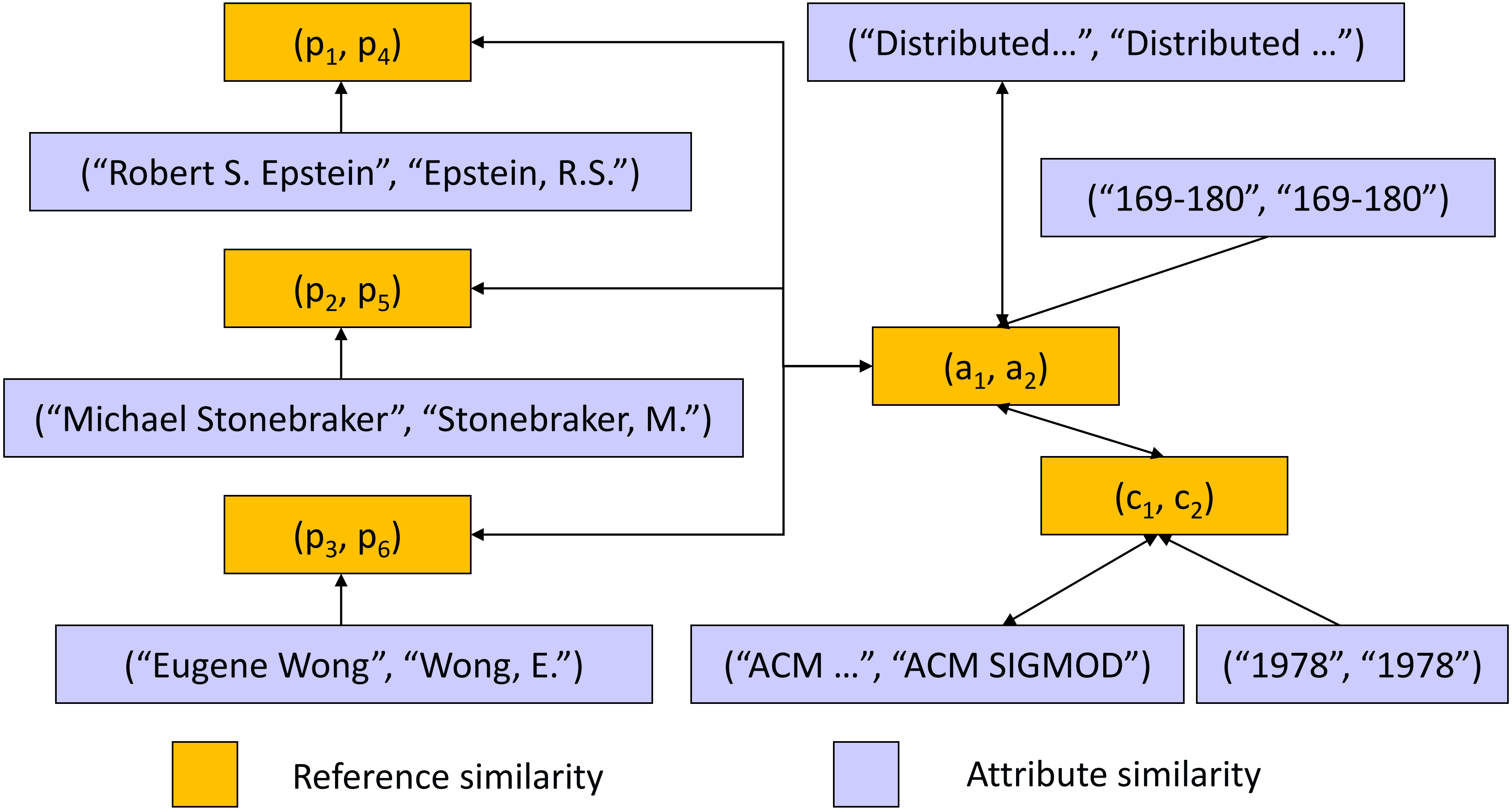
NON-PROBABILISTIC APPROACHES: SIMILARITY PROPAGATION

Similarity Propagation Approaches

- Similarity propagation algorithms define a graph which encodes the entity mentions and matching decisions, and compute matching decisions by propagating similarity values.
  - Details of what type of graph is constructed, and how the similarity is computed varies
  - Algorithms are usually defined procedurally
  - While probabilities may be encoded in various ways in the algorithms, there is no global probabilistic model defined
- Approaches often more scalable than global probabilistic models
- Examples
  - Dependency Graphs [Dong et al, SIGMOD05]
  - Collective Relational Clustering [Bhattacharya & Getoor, TKDD07]

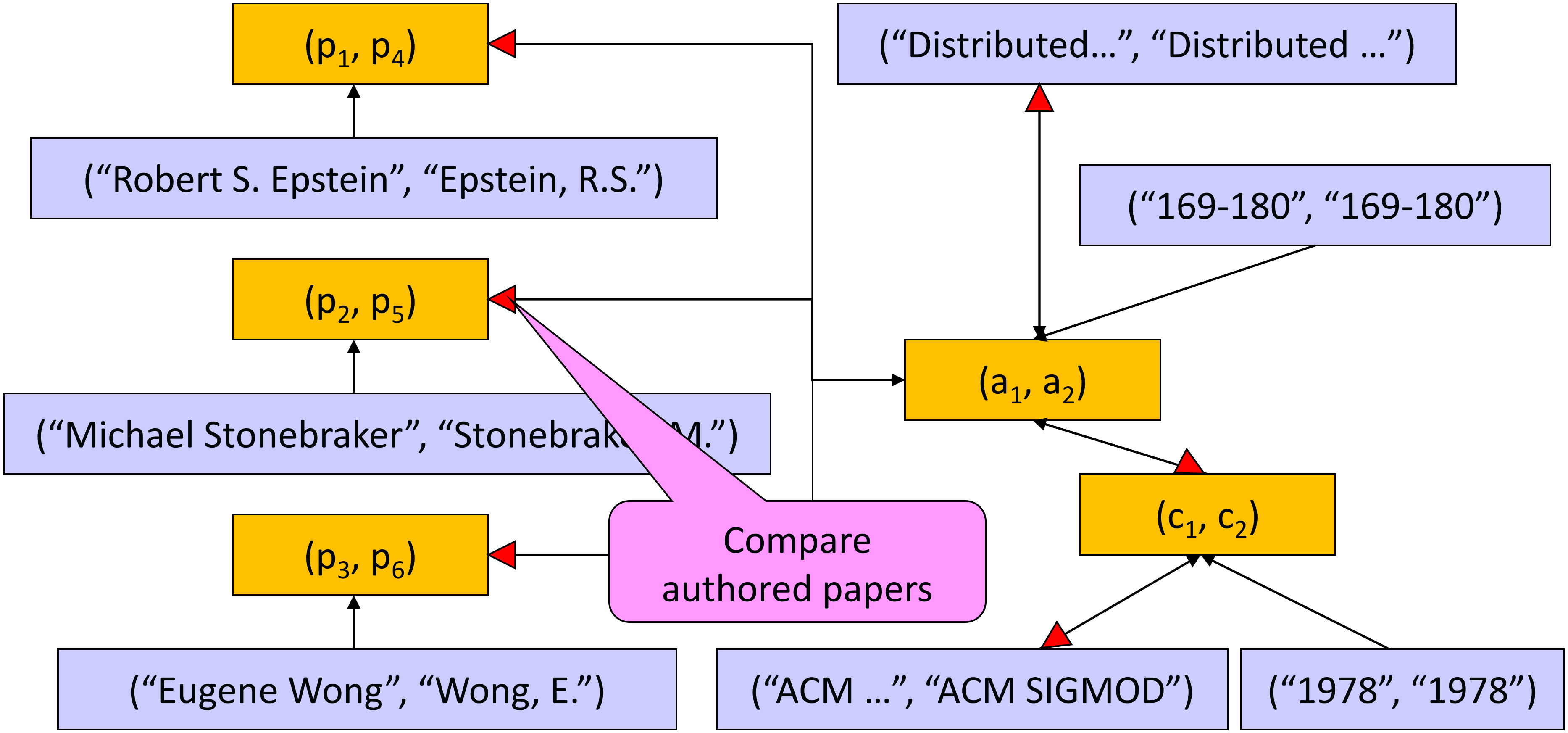
Dependency Graph

[Dong et al., SIGMOD05]



Slides courtesy of [Dong et al.]

Dependency Graph Example II



(p_1, p_4)

(“Robert S. Epstein”, “Epstein, R.S.”)

(p_2, p_5)

(“Michael Stonebraker”, “Stonebrak M.”)

(p_3, p_6)

(“Eugene Wong”, “Wong, E.”)

(“Distributed…”, “Distributed …”)

(“169-180”, “169-180”)

(a_1, a_2)

(c_1, c_2)

(“ACM …”, “ACM SIGMOD”)

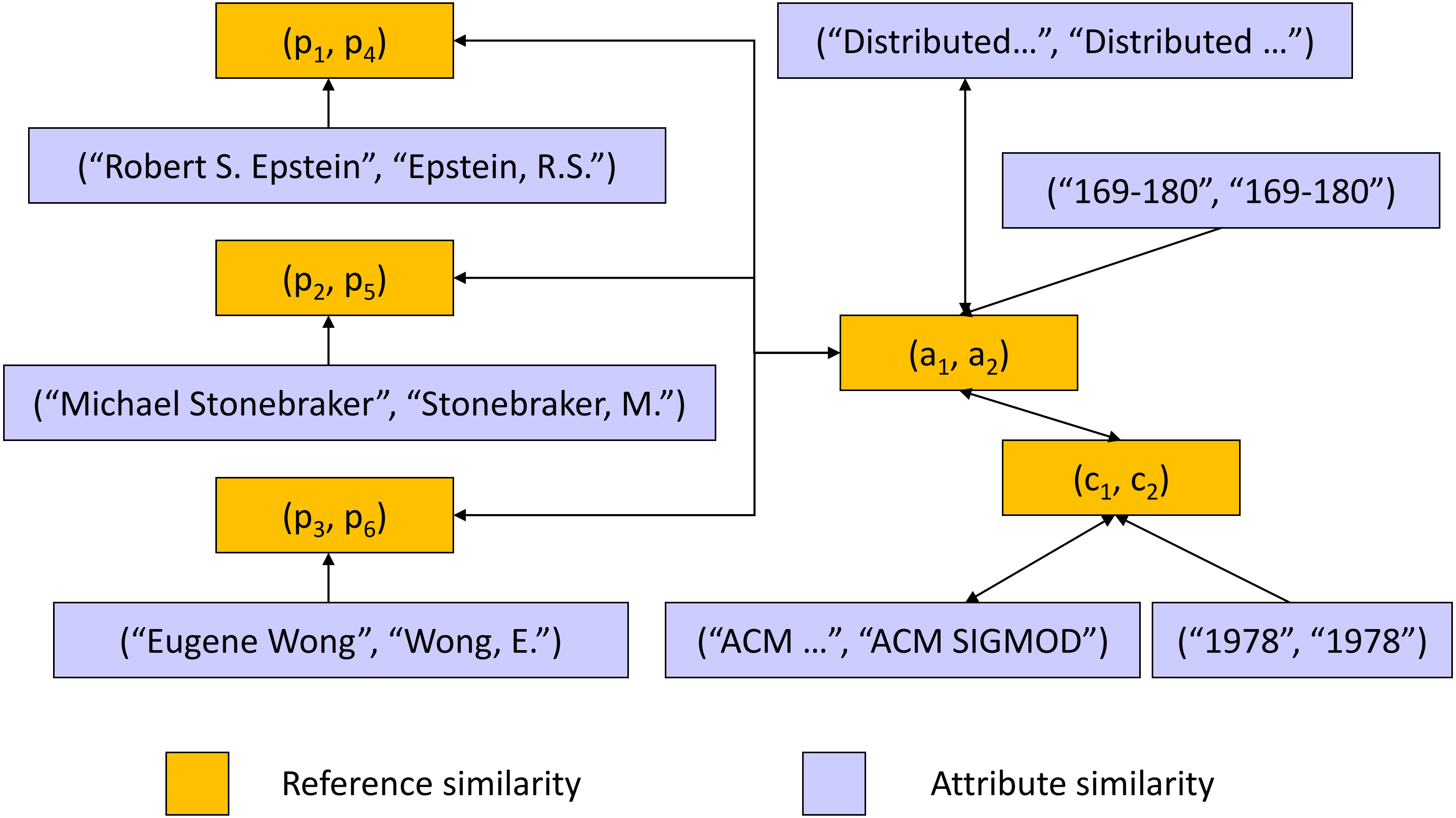
(“1978”, “1978”)

Compare authored papers

🟧 Reference similarity

🟪 Attribute similarity

Exploit the Dependency Graph



(p_1, p_4)

("Robert S. Epstein", "Epstein, R.S.")

(p_2, p_5)

("Michael Stonebraker", "Stonebraker, M.")

(p_3, p_6)

("Eugene Wong", "Wong, E.")

("Distributed…", "Distributed …")

("169-180", "169-180")

(a_1, a_2)

(c_1, c_2)

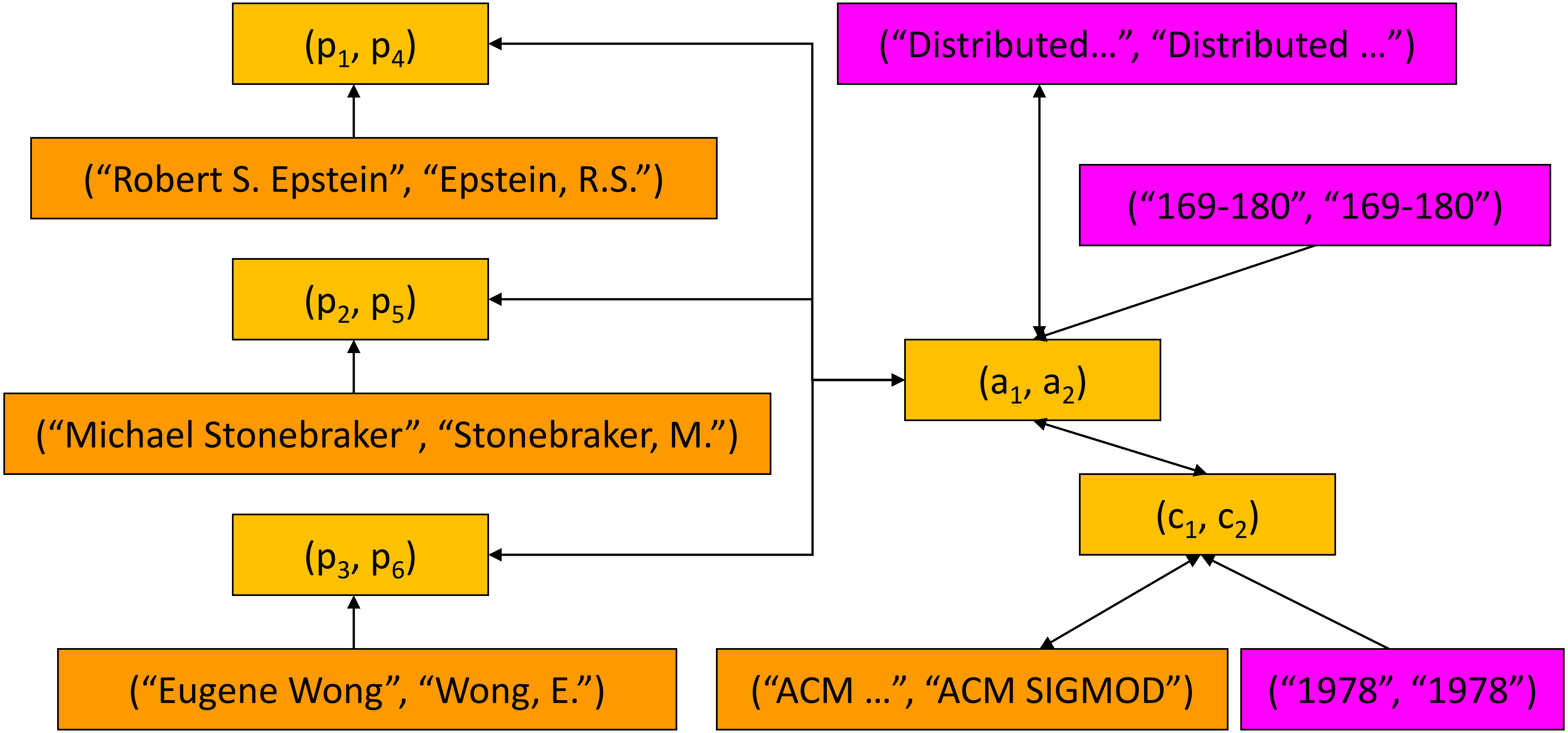
("ACM …", "ACM SIGMOD")

("1978", "1978")

Reference similarity

Attribute similarity

Exploit the Dependency Graph



(p_1, p_4)

("Robert S. Epstein", "Epstein, R.S.")

(p_2, p_5)

("Michael Stonebraker", "Stonebraker, M.")

(p_3, p_6)

("Eugene Wong", "Wong, E.")

("Distributed…", "Distributed …")

("169-180", "169-180")

(a_1, a_2)

(c_1, c_2)

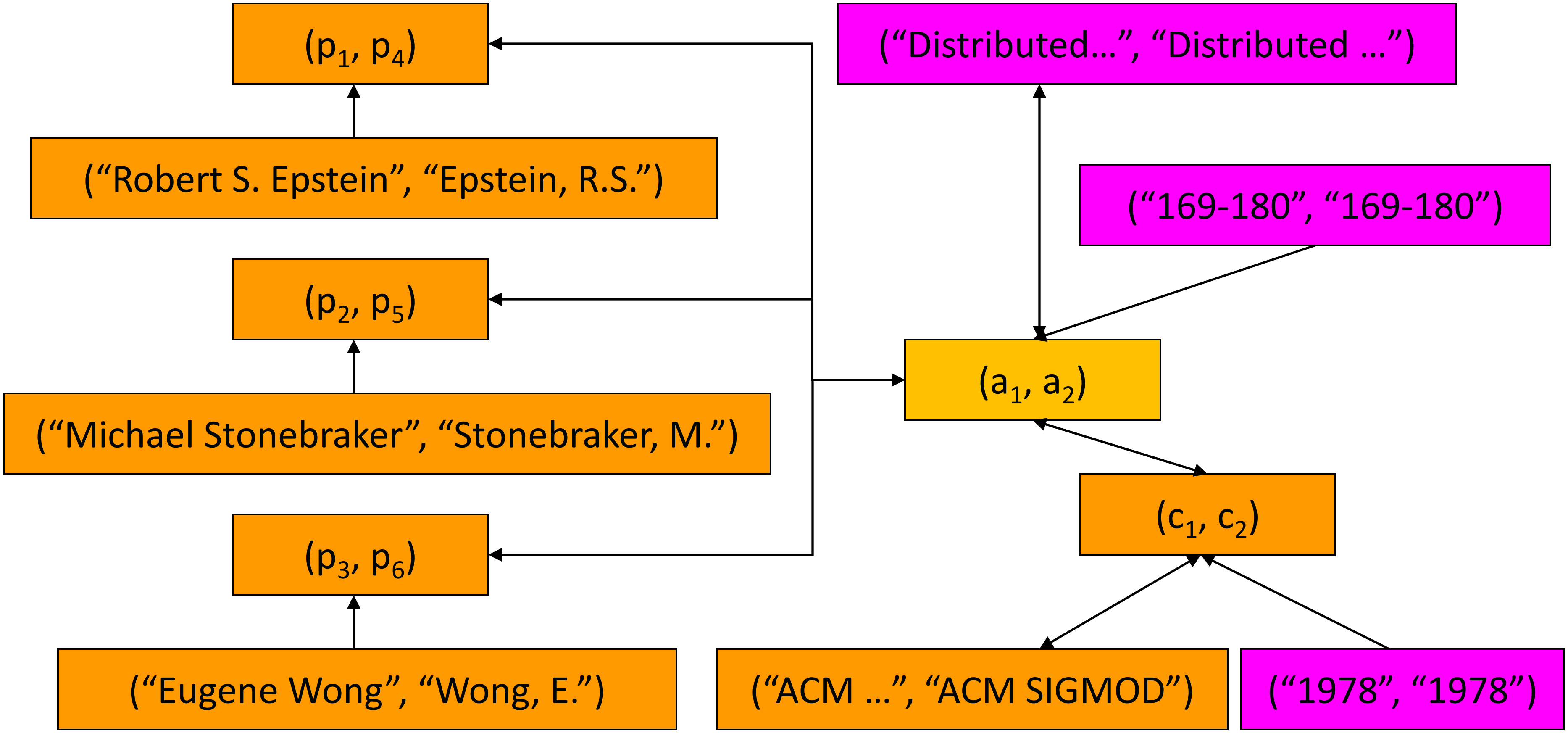
("ACM …", "ACM SIGMOD")

("1978", "1978")

🟪 Reconciled

🟧 Similar

Exploit the Dependency Graph



(p_1, p_4)

(“Robert S. Epstein”, “Epstein, R.S.”)

(p_2, p_5)

(“Michael Stonebraker”, “Stonebraker, M.”)

(p_3, p_6)

(“Eugene Wong”, “Wong, E.”)

(“Distributed…”, “Distributed …”)

(“169-180”, “169-180”)

(a_1, a_2)

(c_1, c_2)

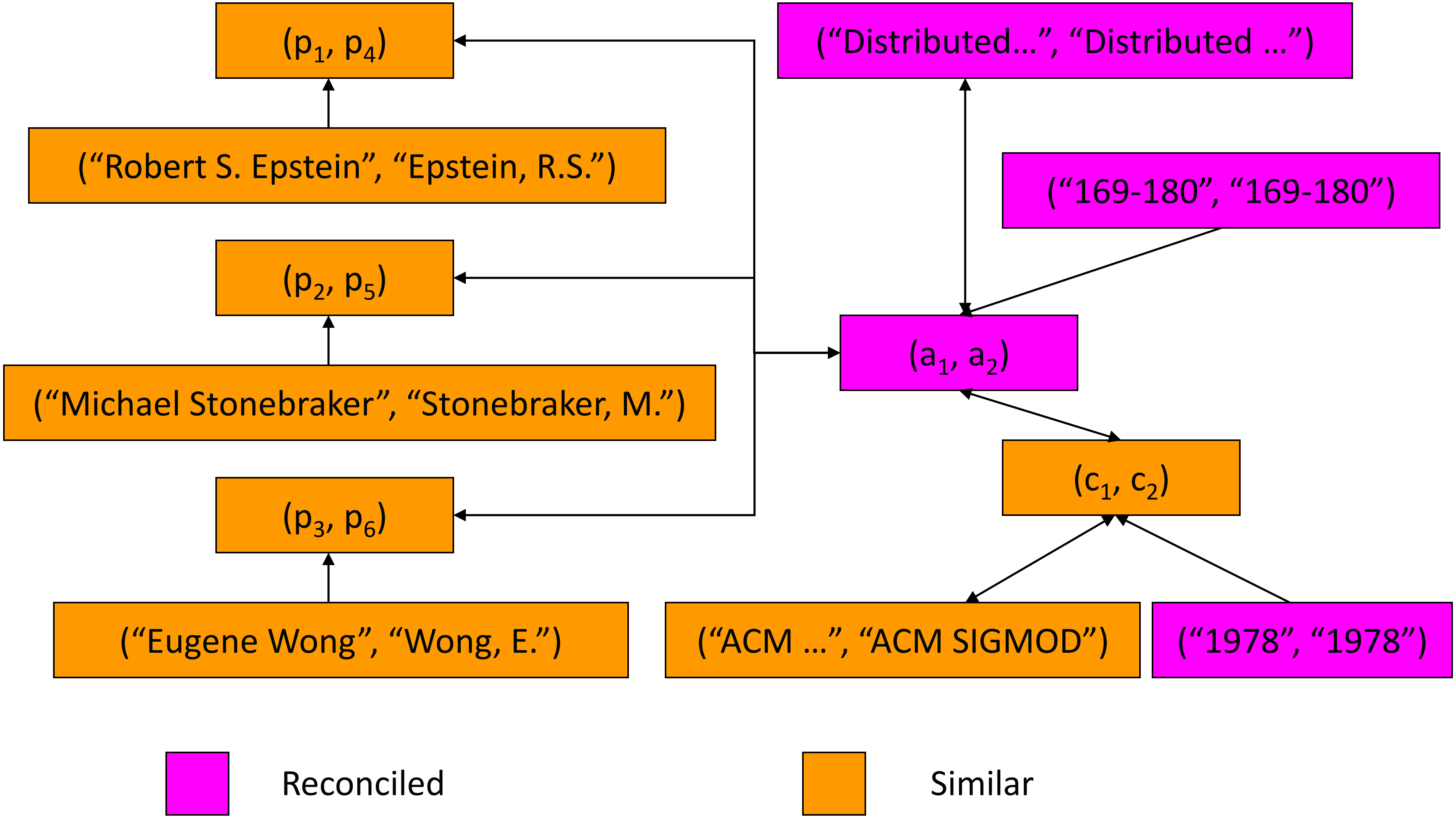
(“ACM …”, “ACM SIGMOD”)

(“1978”, “1978”)

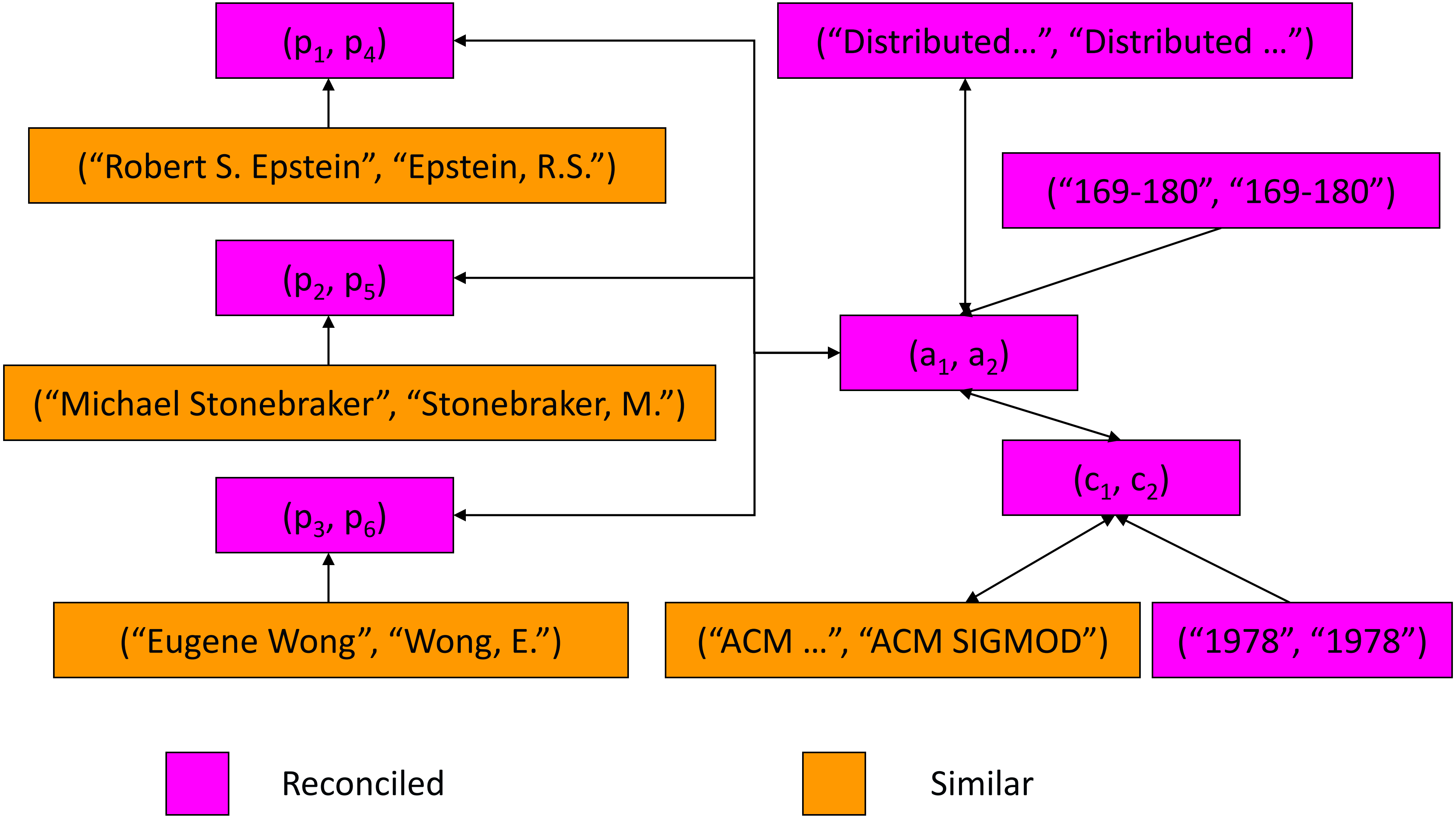
🟪 Reconciled

🟧 Similar

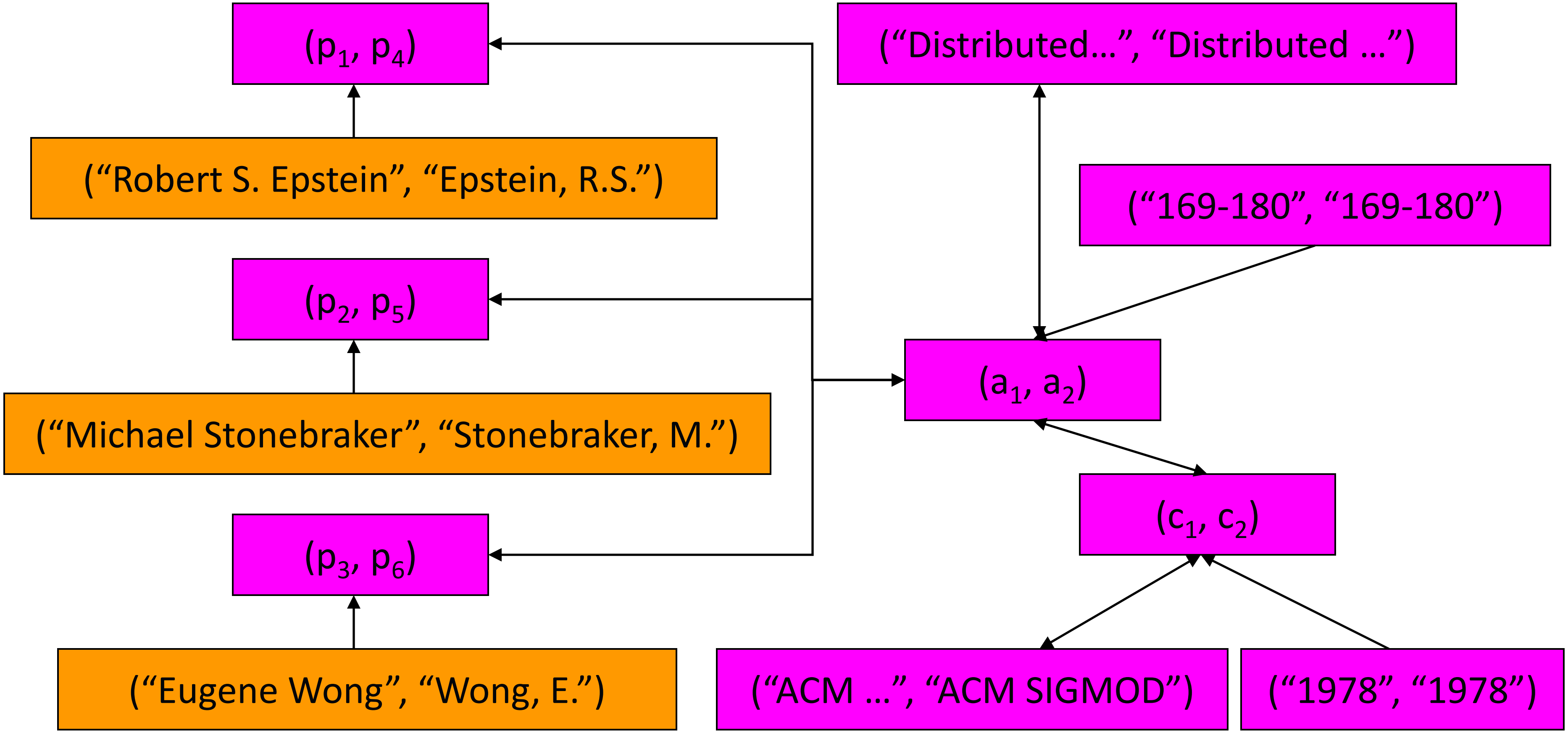
Exploit the Dependency Graph


(p_1, p_4)
("Distributed…", "Distributed …")
("Robert S. Epstein", "Epstein, R.S.")
("169-180", "169-180")
(p_2, p_5)
(a_1, a_2)
("Michael Stonebraker", "Stonebraker, M.")
(c_1, c_2)
(p_3, p_6)
("Eugene Wong", "Wong, E.")
("ACM …", "ACM SIGMOD")
("1978", "1978")
Reconciled
Similar

Exploit the Dependency Graph


(p_1, p_4)
("Distributed…", "Distributed …")
("Robert S. Epstein", "Epstein, R.S.")
("169-180", "169-180")
(p_2, p_5)
(a_1, a_2)
("Michael Stonebraker", "Stonebraker, M.")
(c_1, c_2)
(p_3, p_6)
("Eugene Wong", "Wong, E.")
("ACM …", "ACM SIGMOD")
("1978", "1978")
Reconciled
Similar

Exploit the Dependency Graph



(p_1, p_4)

("Robert S. Epstein", "Epstein, R.S.")

(p_2, p_5)

("Michael Stonebraker", "Stonebraker, M.")

(p_3, p_6)

("Eugene Wong", "Wong, E.")

("Distributed…", "Distributed …")

("169-180", "169-180")

(a_1, a_2)

(c_1, c_2)

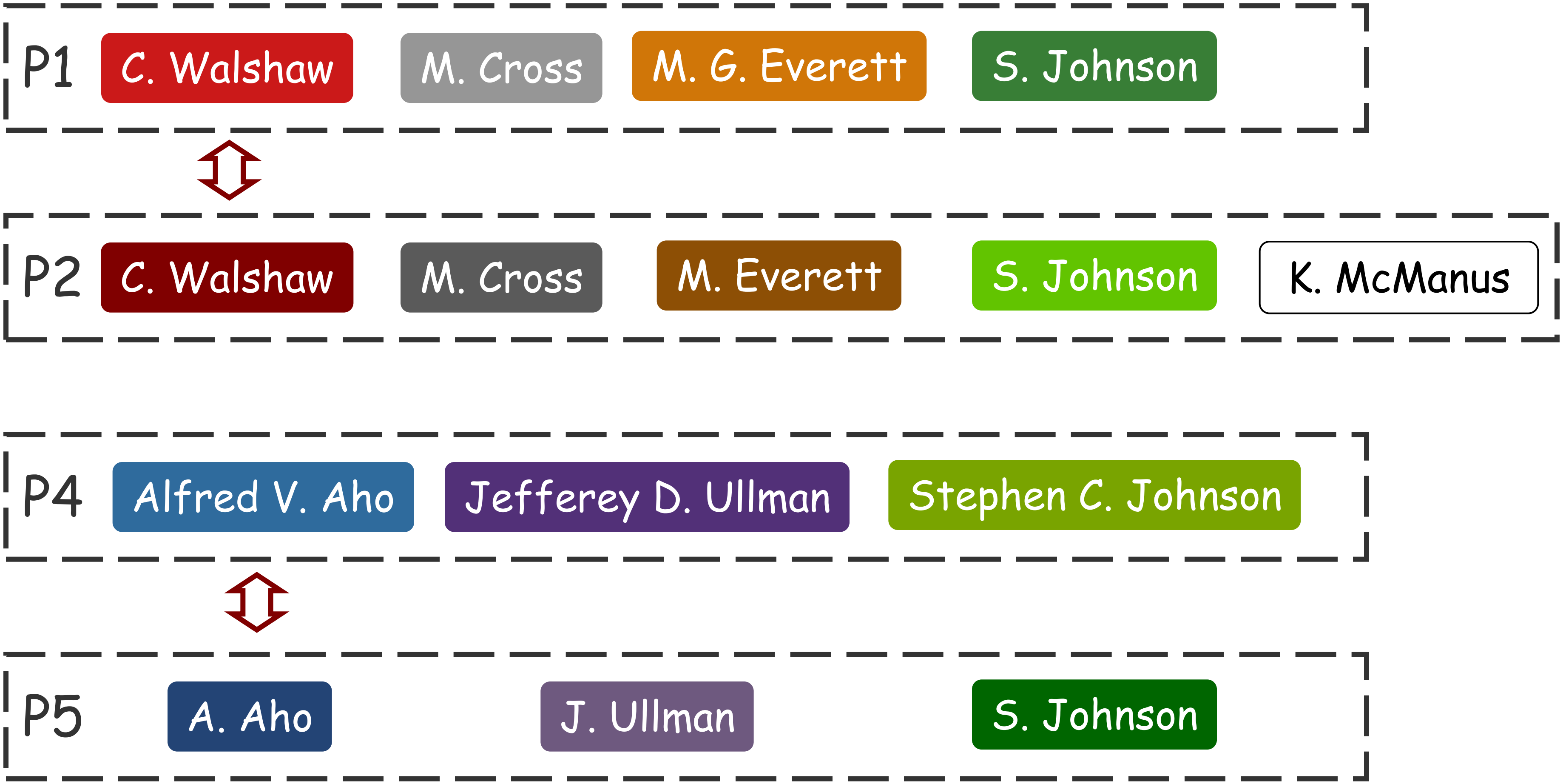
("ACM …", "ACM SIGMOD")

("1978", "1978")

🟪 Reconciled

🟧 Similar

Relational Clustering for ER (RC-ER)



P1 C. Walshaw M. Cross M. G. Everett S. Johnson

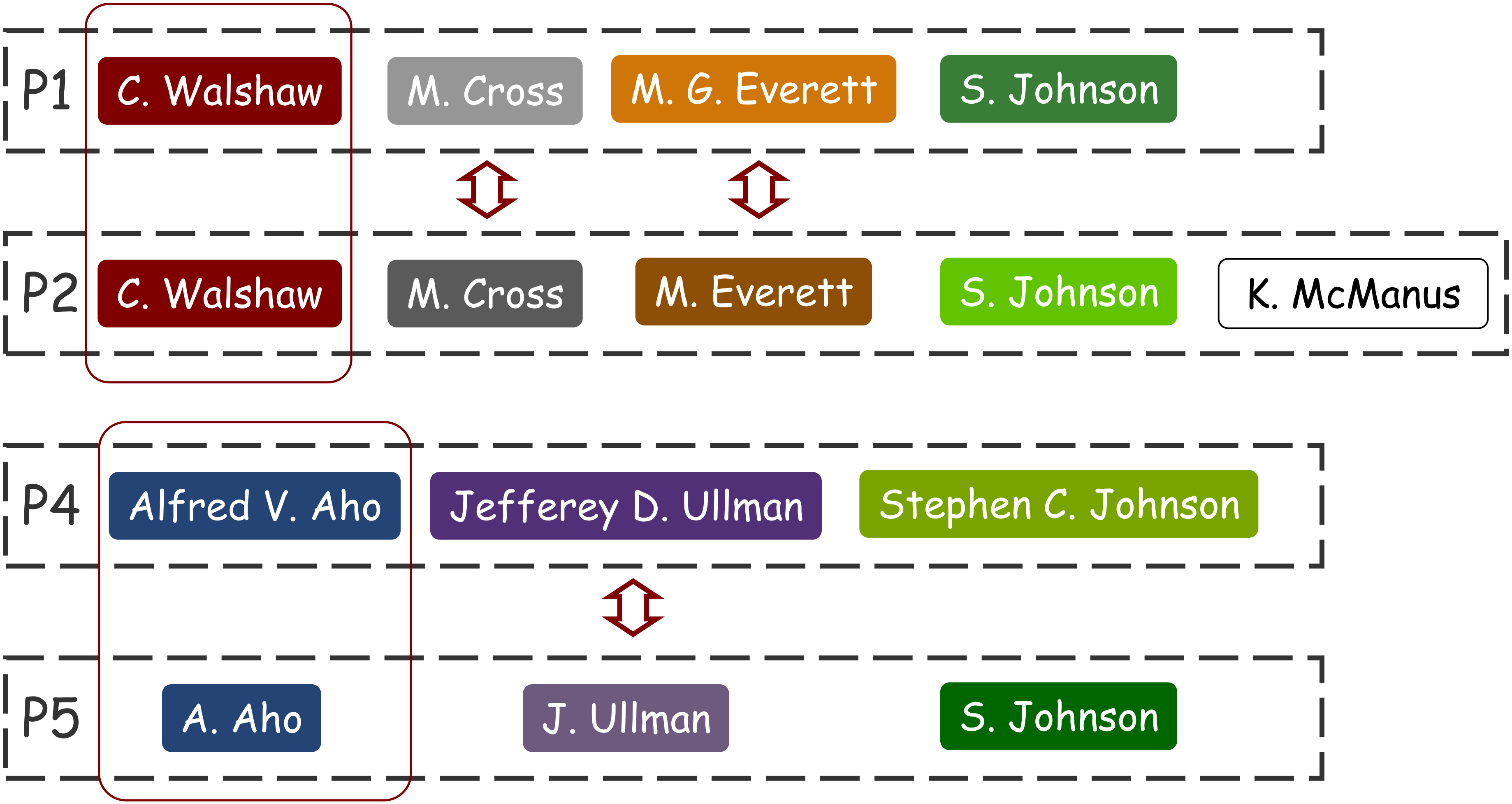
P2 C. Walshaw M. Cross M. Everett S. Johnson K. McManus

P4 Alfred V. Aho Jefferey D. Ullman Stephen C. Johnson

P5 A. Aho J. Ullman S. Johnson

[Bhattacharya & Getoor, TKDD07]

Relational Clustering for ER (RC-ER)



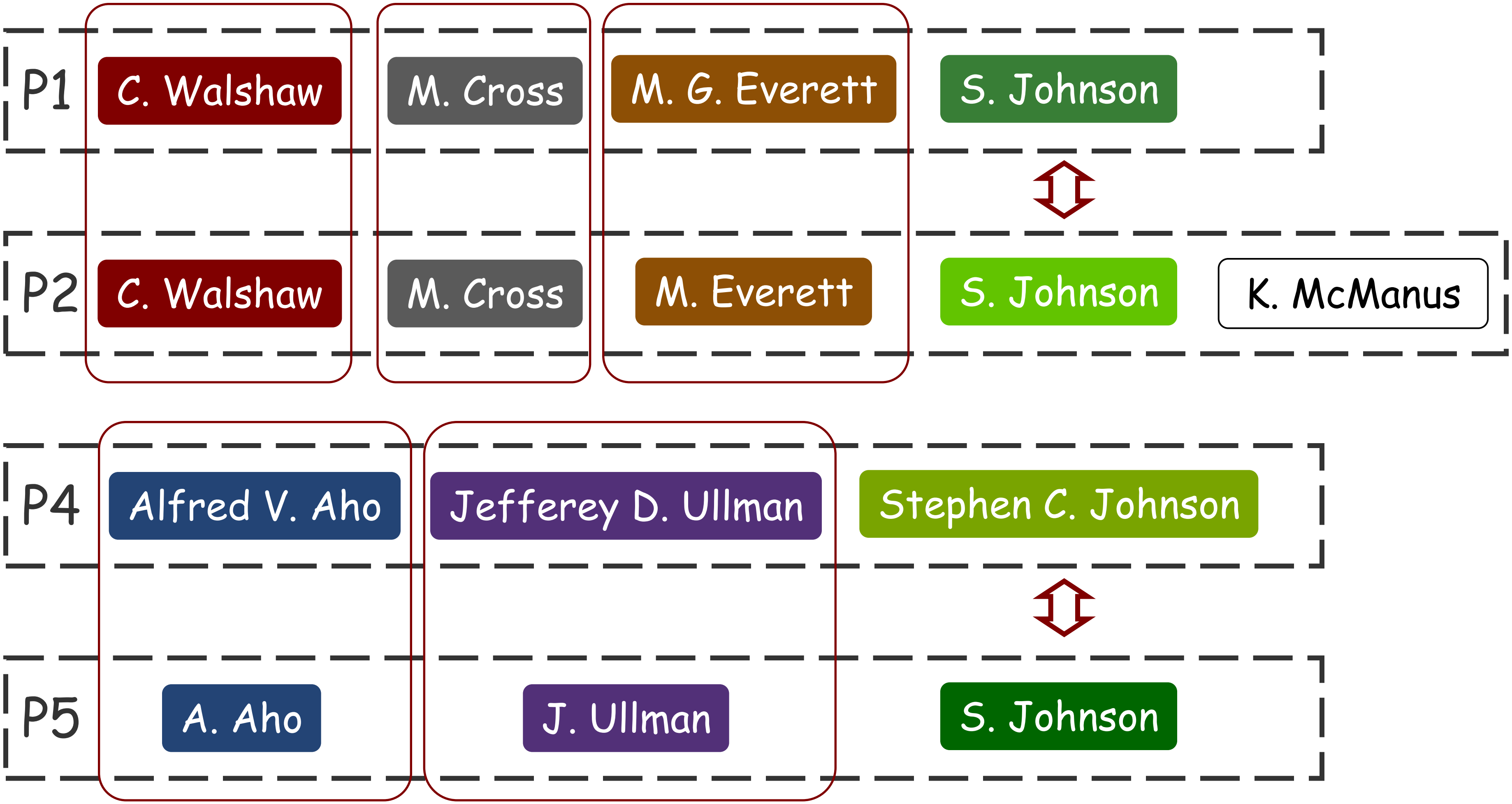
P1 C. Walshaw M. Cross M. G. Everett S. Johnson

P2 C. Walshaw M. Cross M. Everett S. Johnson K. McManus

P4 Alfred V. Aho Jefferey D. Ullman Stephen C. Johnson

P5 A. Aho J. Ullman S. Johnson

Relational Clustering for ER (RC-ER)



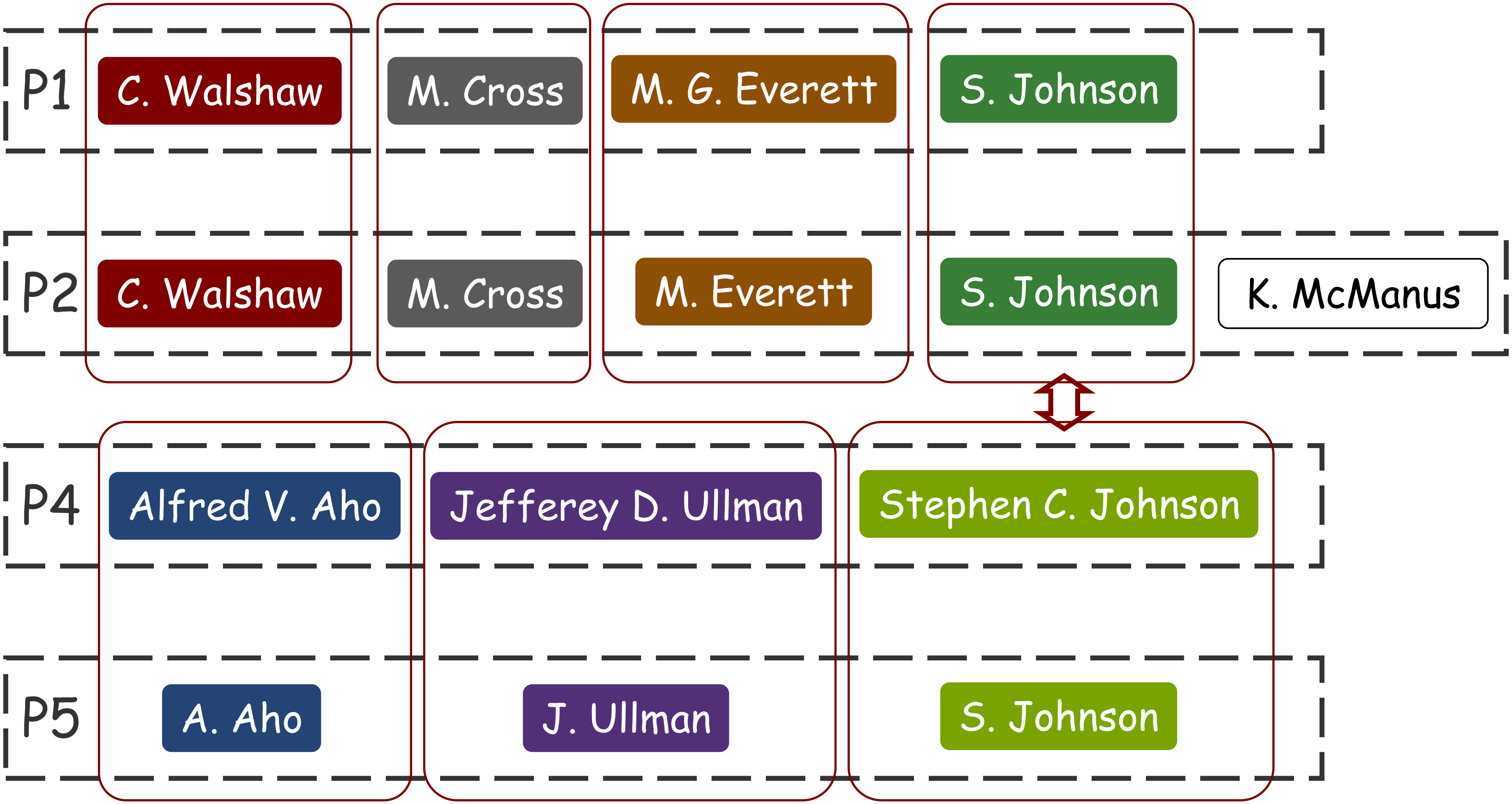
P1 C. Walshaw M. Cross M. G. Everett S. Johnson

P2 C. Walshaw M. Cross M. Everett S. Johnson K. McManus

P4 Alfred V. Aho Jefferey D. Ullman Stephen C. Johnson

P5 A. Aho J. Ullman S. Johnson

Relational Clustering for ER (RC-ER)



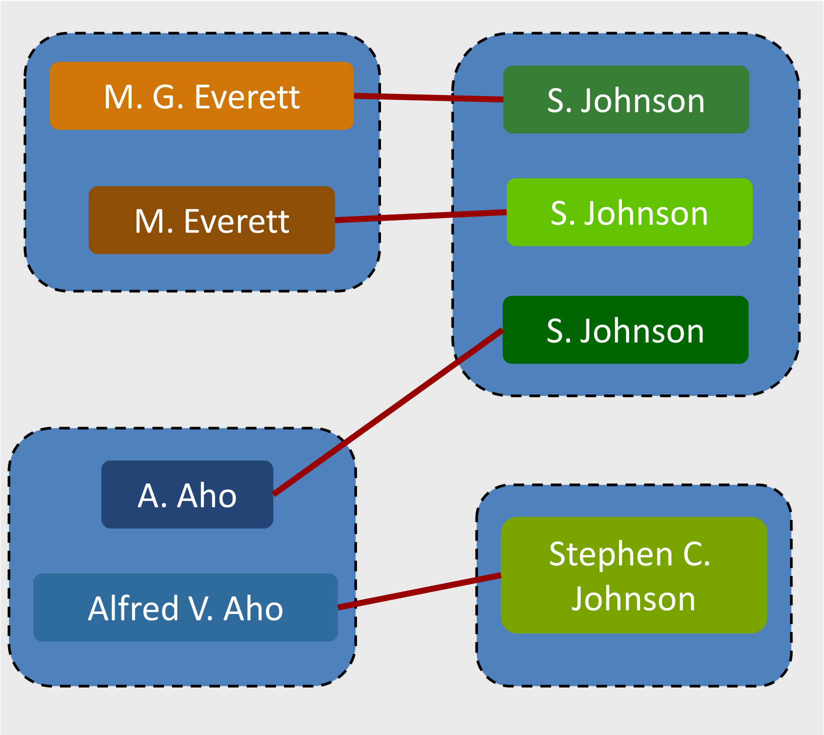
P1 C. Walshaw M. Cross M. G. Everett S. Johnson

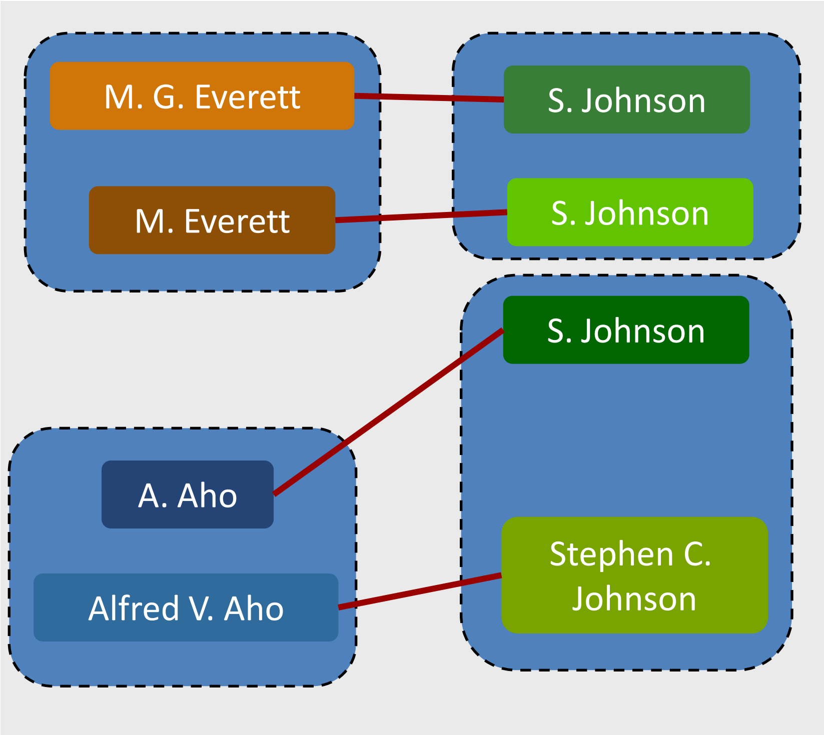
P2 C. Walshaw M. Cross M. Everett S. Johnson K. McManus

P4 Alfred V. Aho Jefferey D. Ullman Stephen C. Johnson

P5 A. Aho J. Ullman S. Johnson

Collective Relational Clustering: Motivation





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

Similarity Measures

- Attribute Similarity
  - Use best available measure for each attribute
  - Name Strings: *Soft TF-IDF, Levestein, Jaro*
  - Textual Attributes: *TF-IDF*

- Aggregate to find similarity between clusters
  - Single link, Average link, Complete link
  - Cluster representative

- Relational Similarity
  - Measures of set similarity
  - Higher order similarity: Consider nbrs of nbrs
  - Can also consider neighborhood as multi-set

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

Similarity-propagation Approaches

	Method	Notes	Constraints	Evaluation
RelDC [Kalashnikov et al, TODS06]	Reference disambiguation using using Relationship-based data cleaning (RelDC)	Model choice nodes identified using feature-based similarity	Context attraction measures the relational similarity	Accuracy and runtime for Author resolution and director resolution in Movie database
Reference Reconciliation [Dong et al, SIGMOD05]	Dependency Graph for propagating similarities + enforce non-match constraints	Reference enrichment Explicitly handle missing values Parameters set by hand	Both positive and negative constraints	Precision/Recall, F1 on personal information management data (PIM), Cora dataset
Collective Relational Clustering [Bhattacharya & Getoor, TKDD07]	Modified hierarchical agglomerative clustering approach	Constructs canonical entity as merges are made	Focus on coauthor resolution and propagation	F1 on three bibliographic datasets: CiteSeer, ArXiv, and BioBase

PART 2-d

CONSTRAINT OPTIMIZATION APPROACHES

Constraint-based Approaches

- Constraint-based approaches explicitly encode relational constraints
  - They can be formulated as hybrid of constraints and probabilistic models
  - Or as constraint optimization problem
- Examples
  - Constraint-based Entity Matching [Shen, Li & Doan, AAAI05]
  - Dedupalog [Arasu, Re, Suciu, ICDE09]

CME

- Two layer model:
  - Layer 1: Generative model for data sets that satisfy constraints; builds on (Li, Morie, & Roth, AI Mag 2004).
  - Layer 2: EM algorithm and the relaxation labeling algorithm to perform matching. Matching process is carried out in multiple iterations. In each iteration, use EM to estimate parameters of the generative model and a matching assignment, then employs relaxation labeling to exploit the constraints
- First layer clusters mentions into groups (such that all matching mentions belong to the same group) and exploits constraints at the *group level. Once this is done,* the second layer exploits additional constraints at the level of *individual matching mention pairs.*

[Shen, Li & Doan, AAAI05]

Clustering with Dedupalog

PaperRef(id, title, conference, publisher, year)
Wrote(id, authorName, Position)

Data to be deduplicated

TitleSimilar(title1,title2)
AuthorSimilar(author1,author2)

(Thresholded) Fuzzy-Join Output

Step (0) Create Fuzzy Matches; this is input to Dedupalog.

Step (1) Declare the entities

"Cluster Papers, Publishers, & Authors"

```
Paper!(id)     :- PaperRef(id,-,-,-)
Publisher!(p) :- PaperRef(-,-,-,p,-)
Author!(a)     :- Wrote(-,a,-)
```

Dedupalog is *flexible*:
Unique **N**ames **A**ssumption (**UNA**)

Publishers (UNA) and Papers (NOT UNA)

Slides from [Arasu, Re, Suciu, ICDE09]

Step (2) Declare Clusters

Input in the DB

PaperRef(id, title, conference, publisher, year)
Wrote(id, authorName, Position)

"Cluster papers, publishers, and authors"

TitleSimilar(title1,title2)
AuthorSimilar(author1,author2)

```
Paper!(id)     :- PaperRef(id,-,-,-)
Publisher!(p) :- PaperRef(-,-,-,p,-)
Author!(a)     :- Wrote(-,a,-)
```

Clusters are *declared* using * (like IDBs or Views): These are output

Author*(a_1,a_2) <-> AuthorSimilar(a_1,a_2)

"Cluster authors with similar names"

Author1	Author2
AA	Arvind Arasu
Arvind A	Arvind Arasu

*IDBs are **equivalence relations**: Symmetric, Reflexive , & Transitively-Closed Relations: i.e., *Clusters*

A **Dedupalog program** is a set of datalog-like rules

Simple Constraints

“Papers with similar titles should likely be clustered together”

Paper*(id_1,id_2) <-> PaperRef$(id_1,t_1,-)$, PaperRef$(id_2,t_2,-)$,TitleSimilar(t_1,t_2)

Author*(a_1,a_2) <-> AuthorSimilar(a_1,a_2)

(<->) Soft-constraints:
Pay a cost if violated.

Paper*(id_1,id_2) <= PaperEq(id_1,id_2)

¬ **Paper***(id_1,id_2) <= PaperNeq(id_1,id_2)

(<=) Hard-constraints: *Any clustering must satisfy these*

*“Papers in PaperEQ **must** be clustered together,
those in PaperNEQ **must not** be clustered together”*

Hard constraints
are challenging!

1. PaperEQ, PaperNEQ are relations (EDBS)
2. ¬ denotes Negation here.

Advanced Constraints

“Clustering two papers, then must cluster their first authors”

$\textbf{Author}*(a_1,a_2) <= \textbf{Paper}*(id_1,id_2), \text{Wrote}(id_1,a_1,1), \text{Wrote}(id_2,a_2,1)$

“Clustering two papers makes it likely we should cluster their publisher”

$\textbf{Publisher}*(x,y) <- \text{Publishes}(x,p_1), \text{Publishes}(x,p_2),\textbf{Paper}*(p_1,p_2)$

“if two authors do not share coauthors, then do not cluster them”

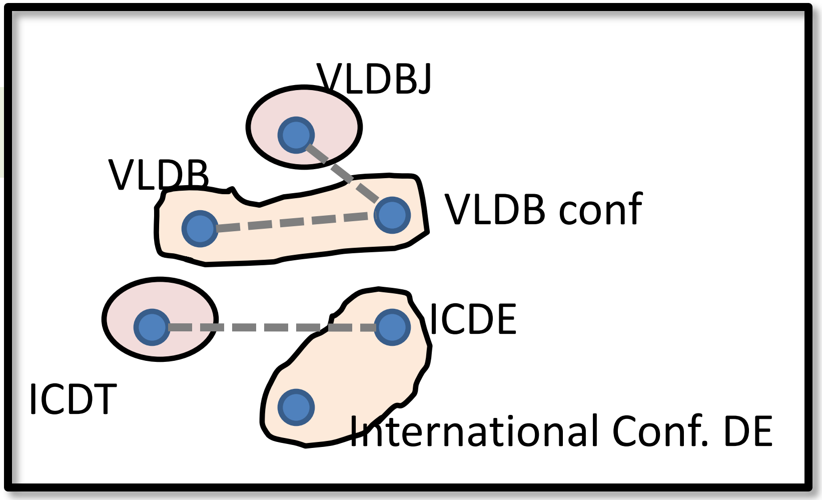
$\neg\ \textbf{Author}* (x, y) <- \neg\ (\text{Wrote}(x, p_1,-), \text{Wrote}(y, p_2,-), \text{Wrote}(z, p_1,-),$
$\text{Wrote}(z, p_2,-), \textbf{Author}*(x, y))$

Dedupalog via CC

Semantics: Translate a Dedupalog Program to a set of graphs

Nodes are references (in the ! Relation)

Entity References: Conference!(c)



Conference*(c_1,c_2) <-> ConfSim(c_1,c_2)

- - - Positive edges

[-] Negative edges are implicit

For a single graph w.o. hard constraints we can reuse prior work for O(1) apx.

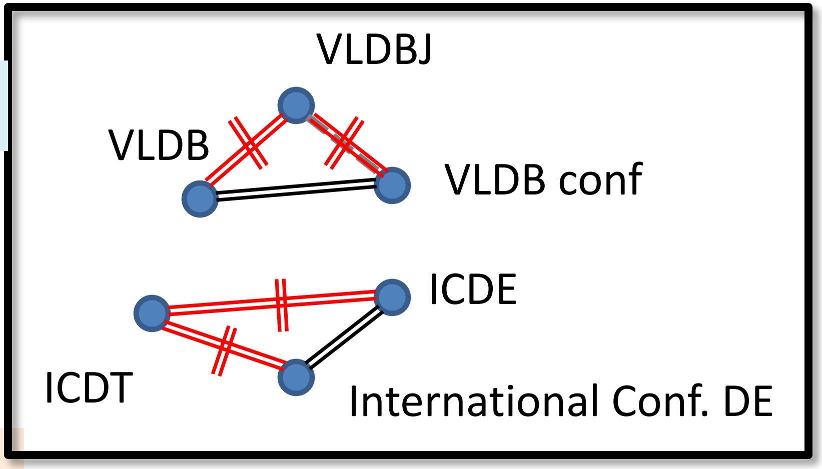
Correlation Clustering

Conference*(c_1,c_2) <- ConfSim(c_1,c_2)

Conference*(c_1,c_2) <= ConfEQ(c_1,c_2)

¬**Conference***(c_1,c_2) <= ConfNEQ(c_1,c_2)

Soft		Hard	
⬚	Positive	☰	Equal
[-]	Negative	▦	Not Equal



```
1. Pick a random order of edges
2. While there is a soft edge do
    1. Pick first soft edge in order
    2. If ⬚ turn into ☰
    3. Else is [-] turn into ▦
    4. Deduce labels
3. Return Transitively closed subsets
```

Simple, Combinatorial algorithm is easy to scale!

Thm: This is a 3-apx!

Voting

Extend algorithm to **whole** language via *voting technique.*
Support many entities, recursive programs, etc.

Many dedupalog programs have an O(1)-apx

Thm: A recursive-hard constraints no O(1) apx

Thm: All "soft" programs O(1)

Expert: multiway-cut hard

System properties:

(1) Streaming algorithm
(2) linear in # of matches (not n^2)
(3) User interaction

Features: Support for weights, reference tables (partially), and corresponding hardness results.

PART 4-d

PROBABILISTIC MODELS: GENERATIVE APPROACHES

Generative Probabilistic Approaches

- Probabilistic semantics based on Directed Models
  - Advantage: generative semantics, can “generate” new instances
  - Disadvantage: acyclicity requirement
- Variety of approaches
  - Based on Bayesian Network semantics, Latent Dirichlet Allocation, etc.
- Examples
  - Latent Dirichlet Allocation [Bhattacharya & Getoor, SDM07]
  - Probabilistic Relational Models [Pasula et al, NIPS02]

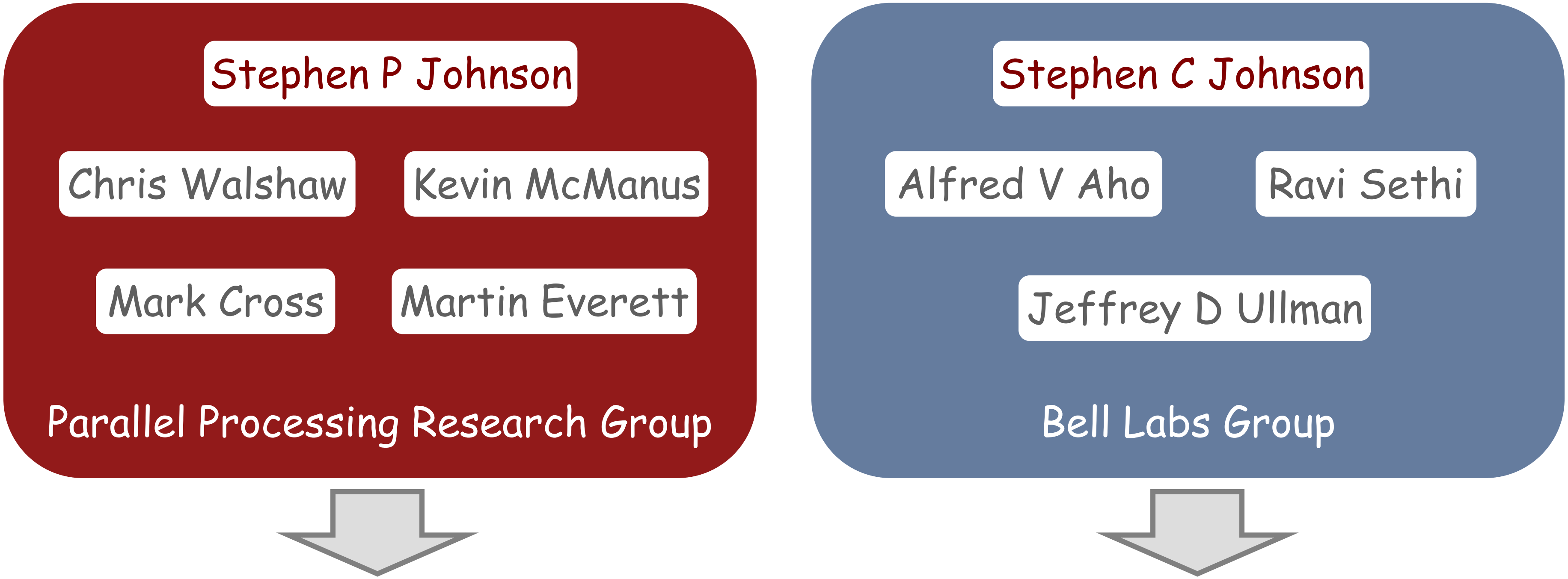
LDA-ER Probabilistic Generative Model

- Model how entity 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



Stephen P Johnson

Chris Walshaw

Kevin McManus

Mark Cross

Martin Everett

Parallel Processing Research Group

Stephen C Johnson

Alfred V Aho

Ravi Sethi

Jeffrey D Ullman

Bell Labs Group

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

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

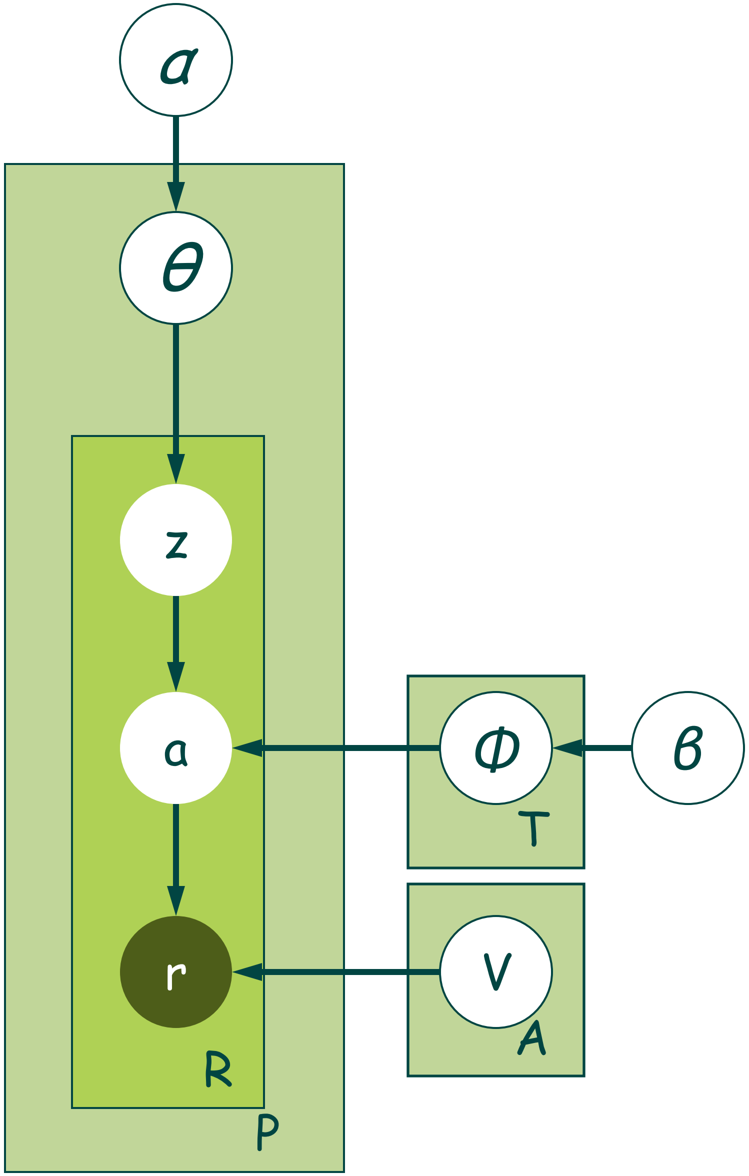
P3: C. Walshaw, M. Cross, M. G. Everett

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



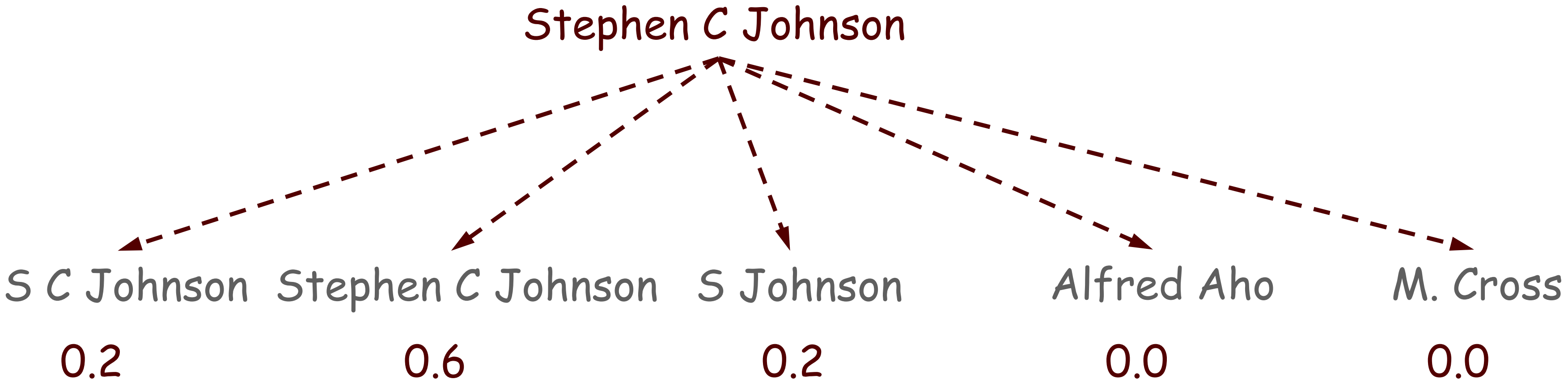
- Entity label *a* and group label *z* for each reference *r*
- *Θ*: 'mixture' of groups for each co-occurrence
- *Φz*: multinomial for choosing entity *a* for each group *z*
- *Va*: multinomial for choosing reference *r* from entity *a*
- Dirichlet priors with α and β

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

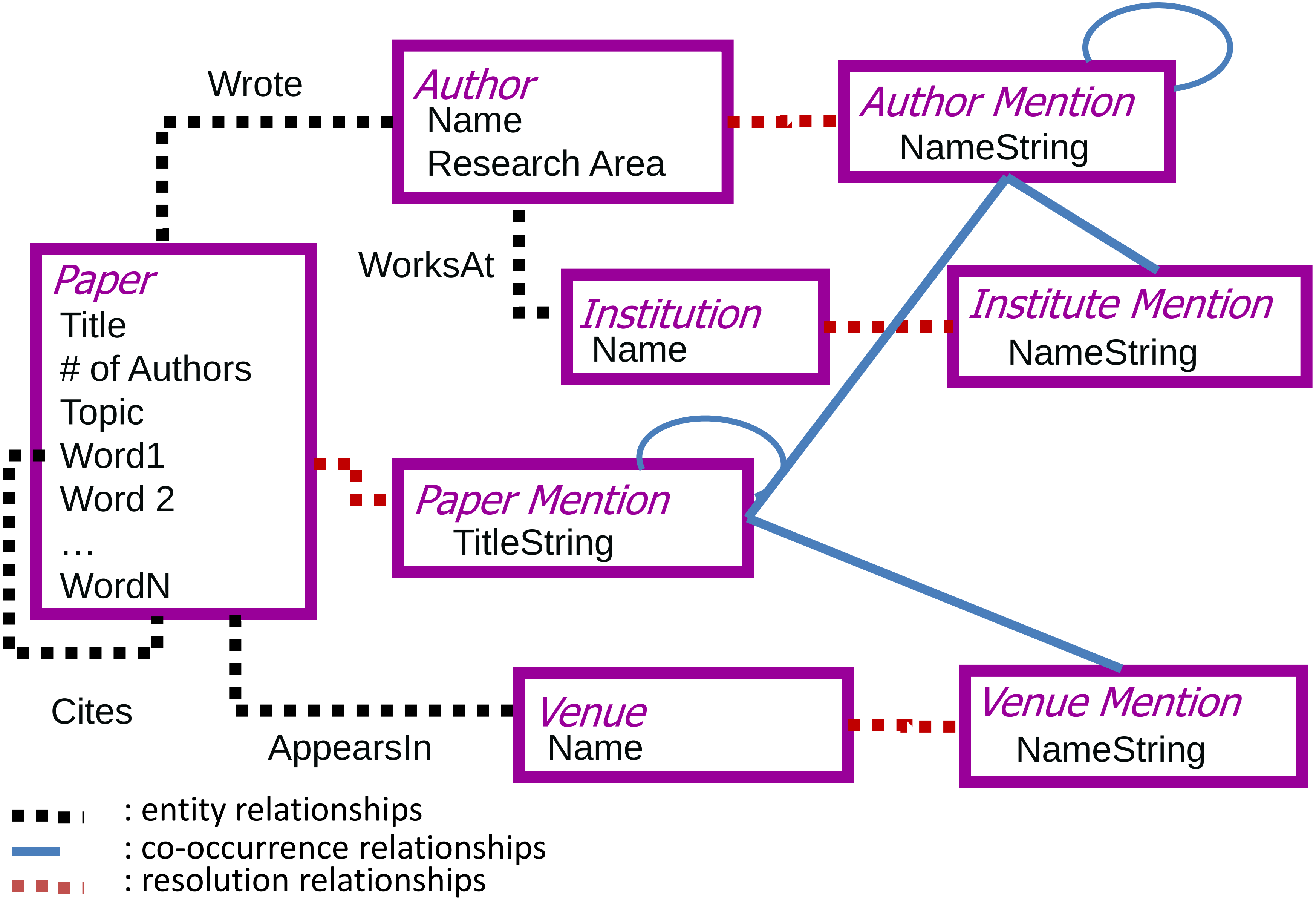
- Alternative: allow entities to merge or split

- For entity a_i, find conditional probabilities for
  1. Merging with existing entity a_j
  2. Splitting back to last merged entities
  3. Remaining unchanged

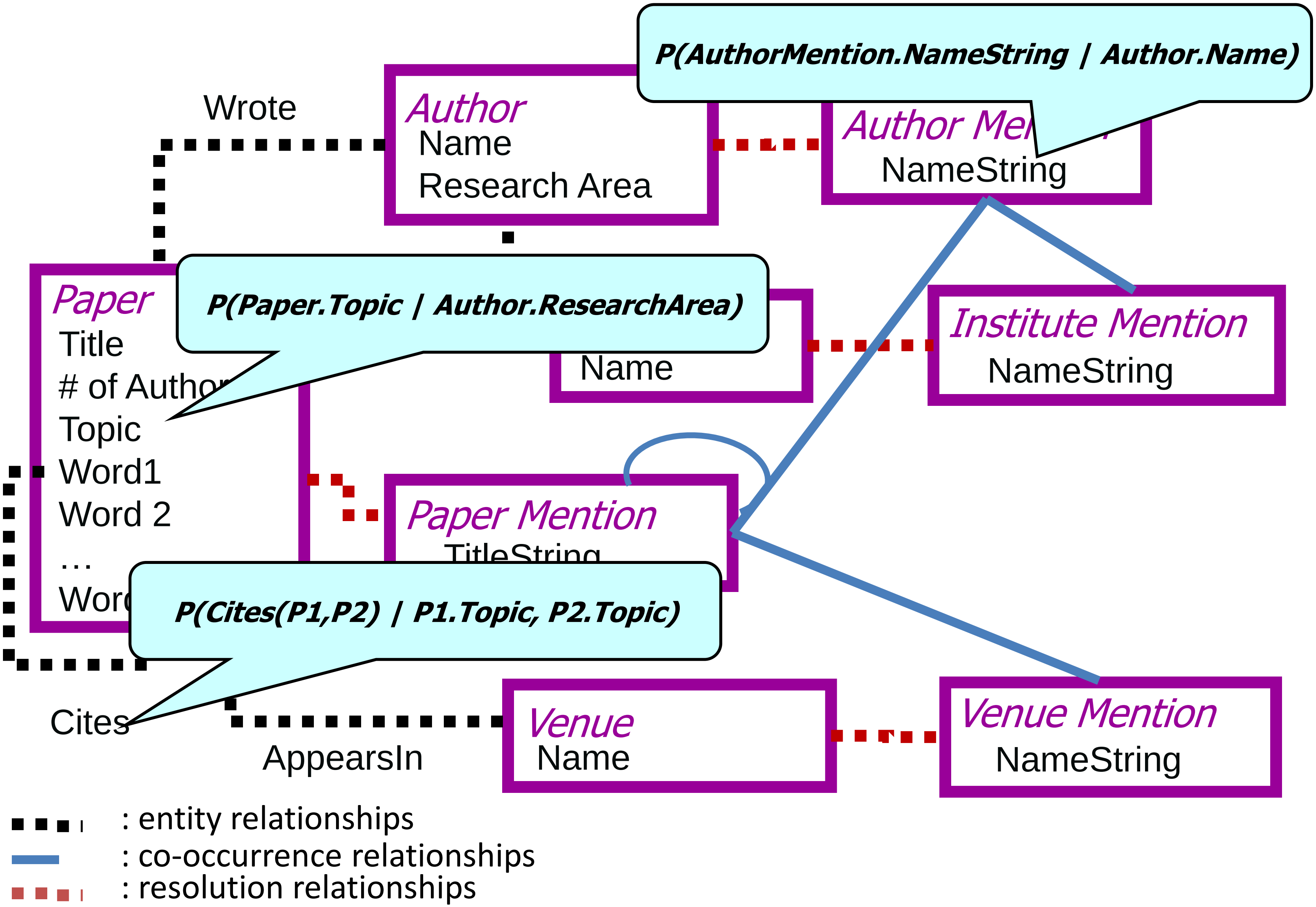
- Sample next state for a_i from distribution

- $O(n\ g + e)$ time per iteration compared to $O(n\ g + n\ e)$

Probabilistic Relational Models for ER



Probabilistic Relational Models



P(AuthorMention.NameString | Author.Name)

P(Paper.Topic | Author.ResearchArea)

P(Cites(P1,P2) | P1.Topic, P2.Topic)

Wrote

Author
Name
Research Area

Author Me
NameString

Paper
Title
# of Autho
Topic
Word1
Word 2
…
Word

Name

Institute Mention
NameString

Paper Mention
TitleString

Cites

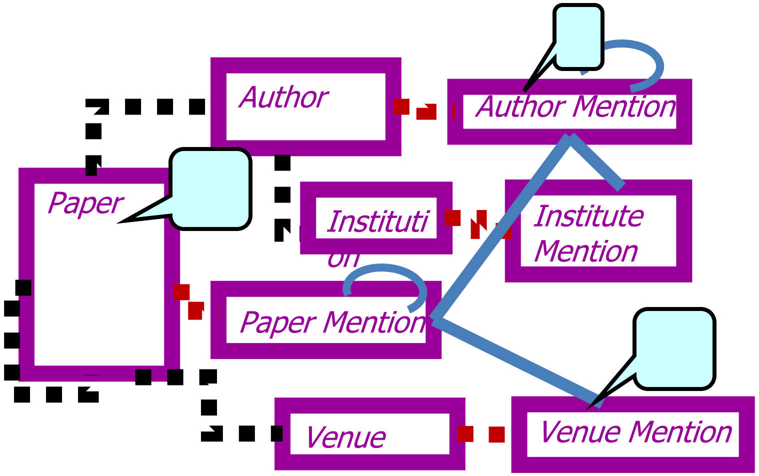
AppearsIn

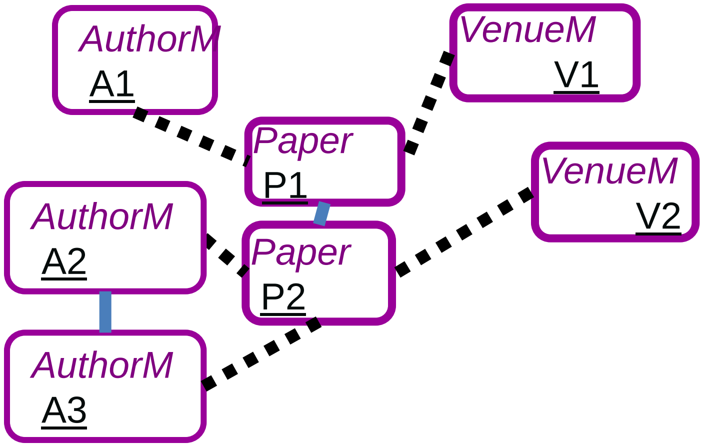
Venue
Name

Venue Mention
NameString

▪ ▪ ▪ : entity relationships
▬ : co-occurrence relationships
▪ ▪ ▪ : resolution relationships

PRM Semantics





PRM + relational skeleton σ =

probability distribution over completions I:

$$P(\mathrm{I} \mid \sigma, S, \Theta) = \prod_{x \in \sigma} \prod_{x.A} P(x.A \mid parents_{S,\sigma}(x.A))$$

Objects Attributes

Inference in PRMs for Citation Matching

[Pasula *et al*., NIPS 2002]

- Parameter estimation
  - Priors for names, titles, citation formats learned offline from labeled data
  - String corruption parameters learned with Monte Carlo EM
- Inference
  - MCMC with cluster recombination proposals
  - Guided by “canopies” of similar citations
  - Accuracy stabilizes after ~20 minutes

Generative Approaches

	Method	Learning/Inference Method	Evaluation
[Li, Morie, & Roth, AAAI 04]	Generative model for mentions in documents	Truncated EM to learn parameters and MAP inference for entities (unsupervised)	F1 on person names, locations and organizations in TREC dataset
Probabilistic Relational Models [Pasula et al., NIPS03]	Probabilistic Relational Models	Parameters learned on separated corpora, inference done using MCMC	% of correctly identified clusters on subsets of CiteSeer data
Latent Dirichlet Allocation [Bhattacharya & Getoor, SDM06]	Latent-Dirichlet Allocation Model	Blocked Gibbs Sampling	Precision/Recall /F1 on CiteSeer and HEP data

PART 4-e

PROBABILISTIC MODELS: UNDIRECTED APPROACHES

Undirected Probabilistic Approaches

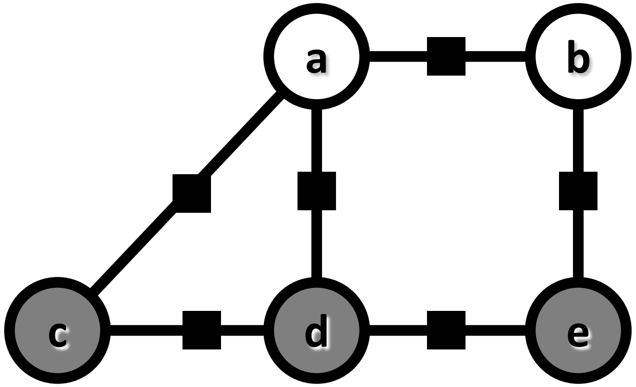
- Probabilistic semantics based on Markov Networks
  - Advantage: no acyclicity requirements
- In some cases, syntax based on first-order logic
  - Advantage: declarative

- Examples
  - Conditional Random Fields (CRFs) [McCallum & Wellner, NIPS04]
  - Markov Logic Networks (MLNs) [Singla & Domingos, ICDM06]
  - Probabilistic Similarity Logic [Broecheler & Getoor, UAI10]

[Lafferty, McCallum, Pereira, ICML01]

Conditional Random Field (CRF)

Undirected graphical model, conditioned on some data variables



$\mathbf{y}$ output predicted variables

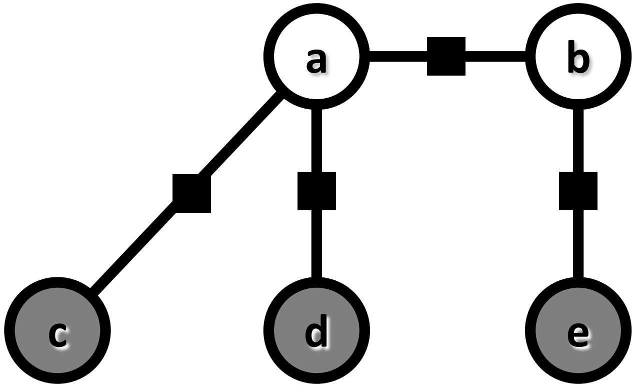
$\mathbf{x}$ input observed variables

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z_{\mathbf{x}}} \prod_{f} \phi(\mathbf{x}_{\in f}, \mathbf{y}_{\in f})$$

[Slides coutesy of Andrew McCallum]

Conditional Random Field (CRF)

Undirected graphical model, conditioned on some data variables



$\mathbf{y}$ output predicted variables

$\mathbf{x}$ input observed variables

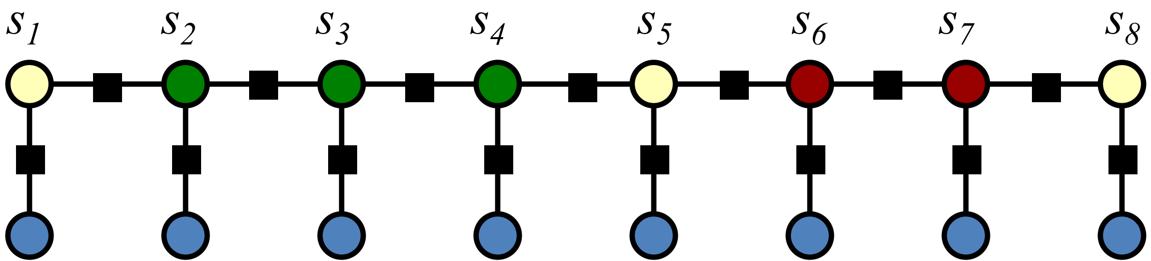
$$p(\mathbf{y}|\mathbf{x}) \; = \; \frac{1}{Z_{\mathbf{x}}} \prod_{f} \phi(\mathbf{x}_{\in f}, \mathbf{y}_{\in f})$$

+ Tremendous freedom to use arbitrary features of input.
+ Predict multiple dependent variables (“structured output”)

Information Extraction with Linear-chain CRFs

Logistic Regression analogue of a hidden Markov model

Graphical model

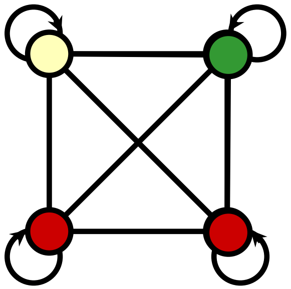


state sequence

observation sequence

Today Morgan Stanley Inc announced Mr. Friday's appointment.

Finite state model



🟥 person name
🟩 organization name
🟨 background

CRF for ER

[McCallum & Wellner, NIPS04]

- CRF with random variables for each mention pair
- Factors capture dependence among mentions assigned to the same cluster
- Show that inference in above CRF is equivalent to graph partitioning in graph where nodes are mentions and edges weights are log clique potentials over nodes
- Learn weights from training data; variety of weight learning approaches, here use voted perceptron
- Graph partitioning performed using correlation clustering

Markov Logic

- A logical KB is a set of **hard constraints** on the set of possible worlds
- Make them **soft constraints;** when a world violates a formula, it becomes less probable but not impossible
- Give each formula a **weight**
  - Higher weight ⇒ Stronger constraint

$$P(world) \propto \exp\left(\sum weights\ \ of\ \ formulas\ \ it\ \ satisfies\ \right)$$

[Richardson & Domingos, 06]

Markov Logic

- A **Markov Logic Network (MLN)** is a set of pairs **(F, w)** where
  - **F** is a formula in first-order logic
  - **w** is a real number

# true groundings of *ith* clause

$$P(X) = \frac{1}{Z} \exp\left(\sum_{i \in F} w_i n_i(x) \right)$$

Normalization Constant

Iterate over all first-order MLN formulas

[Richardson & Domingos, 06]

Problem Formulation

- **Given**
  - A database of records representing entities in the real world e.g. citations
  - A set of fields e.g. author, title, venue
  - Each record represented as a set of typed predicates e.g. *HasAuthor(citation,author), HasVenue(citation,venue)*
- **Goal**
  - To determine which of the records/fields refer to the same underlying entity

Slides from [Singla & Domingos, ICDM 06]

Problem Formulation

- **Given**
  - DB of mentions of entities in the real world, e.g. citations
  - A set of fields, e.g. author, title, venue
  - Each record represented as a set of typed predicates e.g. *HasAuthor(citation,author), HasVenue(citation,venue)*
- Entities in the real world represented by one or more strings appearing in the DB, e.g. ***"J. Cox"***, ***"Cox J."***
- String constant for each record, e.g. ***"C1"***, ***"C2"***
- **Goal**: for each pair of string constants $<x_1, x_2>$ of the same type, is $x_1 = x_2$?

Slides based on [Singla & Domingos, ICDM 06]

Handling Equality

- Introduce ***Equals(x,y)*** or $x = y$
- Introduce the axioms of equality
  - Reflexivity: $x = x$
  - Symmetry: $x = y \Rightarrow y = x$
  - Transitivity: $x = y \wedge y = z \Rightarrow z = x$
  - Predicate Equivalence:

$$x_1 = x_2 \wedge y_1 \wedge y_2 \Rightarrow (R(x_1, y_1) \Leftrightarrow R(x_2,y_2))$$

Handling Equality

- Introduce **reverse predicate equivalence**
- Same relation with the same entity gives evidence about two entities being same

$$R(x_1,y_1) \wedge R(x_2,y_2) \wedge x_1 = x_2 \Rightarrow y_2 = y_2$$

- Not true logically, but gives useful information
- Example

$$HasAuthor(C1, J.\ Cox) \wedge HasAuthor(C2, Cox\ J.) \wedge C1 = C2 \Rightarrow (J.\ Cox = Cox\ J.)$$

Model for Entity Resolution

- Model is in the form of an MLN
- Query predicate is ***Equality***
- Evidence predicates are relations which hold according to the DB
- Introduce axioms of equality
- First-order rules for field comparison, Fellegi-Sunter model, relational models

Field Comparison

- Each field is a string composed of tokens
- Introduce ***HasWord(field, word)***
- Use reverse predicate equivalence

$$HasWord(f_1,w_1) \wedge HasWord(f_2,w_2) \wedge w_1 = w_2 \Rightarrow f_1 = f_2$$

- Example

$$HasWord(J.\ Cox,\ Cox) \wedge HasWord(Cox\ J.,\ Cox) \wedge (Cox = Cox) \Rightarrow (J.\ Cox = Cox\ J.)$$

- Different weight for each word : learnable similarity measure of Bilenko & Mooney [2003]

Two-level Similarity

- Individual words as units: Can’t deal with spelling mistakes
- Break each word into ngrams: Introduce ***HasNgram(word, ngram)***
- Use reverse predicate equivalence for word comparisons
- Gives a two level similarity measure as proposed by Cohen et al. [2003]

Fellegi-Sunter Model

- Uses Naïve Bayes for match decisions with field comparisons used as predictors
- Simplest Version: Field similarities measured by presence/absence of words in common

$$HasWord(f_1, w_1) \wedge HasWord(f_2, w_2) \wedge HasField(r_1, f_1) \wedge HasField(r_2, f_2) \wedge w_1 = w_2 \Rightarrow r_1 = r_2$$

- Example

HasWord(J. Cox, Cox) ∧ HasWord(Cox J., Cox) ∧ HasAuthor(C1, J. Cox) ∧ HasAuthor(C2, Cox J.) ∧ (Cox = Cox) ⇒ (C1 = C2)

Relational Models

- Fellegi-Sunter + transitivity [McCallum & Wellner 2005]

  $(f_1 = f_2) \wedge (f_2 = f_3) \Rightarrow (f_3 = f_1)$

- Fellegi-Sunter + reverse predicate equivalence for records/fields [Singla & Domingos 2005]

  $HasField(r_1, f_1) \wedge HasField(r_2, f_2) \wedge f_1 = f_2 \Rightarrow r_1 = r_2$

  $HasAuthor(C1, J.\ Cox) \wedge HasAuthor(C2, Cox\ J.) \wedge (J.\ Cox = Cox\ J.) \Rightarrow C1 = C2$

Relational Models

- Co-authorship relation for entity resolution [Bhattacharya & Getoor, DMKD04]

$$HasAuthor(c,a_1) \wedge HasAuthor(c,a_2) \Rightarrow Coauthor(a_1,a_2)$$

$$Coauthor(a_1, a_2) \wedge Coauthor(a_3, a_4) \wedge a_1 = a_3 \Rightarrow a_2 = a_4$$

Scalability

- $O(n^2)$ number of match decisions - too big even for small databases
- Use cheap heuristics (e.g. TFIDF based similarity) to identify plausible pairs
- Used the canopy approach [McCallum et al., KDD00]
- Inference/learning over plausible pairs

Probabilistic Soft Logic

- Declarative language for defining **constrained continuous Markov random field** (CCMRF) using first-order logic (FOL)
- Soft logic: truth values in [0,1]
- Logical operators relaxed using Lukasiewicz t-norms
- Mechanisms for incorporating similarity functions, and reasoning about sets
- MAP inference is a **convex optimization**
- Efficient sampling method for marginal inference

[Broecheler & Getoor, UAI10]

Predicates and Atoms

- Predicates
  - Describe relations
  - Combined with arguments to make atoms
- Atoms
  - Lifted: contains variables, e.g., Friends(X, Y)
  - Ground: no variables, e.g., AuthorOf(author1, paper1)
- Each ground atom can have a truth value in [0,1]
- PSL programs define distributions over the truth values of ground atoms

Weighted Rules

- A PSL program is a set of weighted, logical rules
- For example,

```
authorName(A1,N1) ^ authorName(A2,N2) ^ similarString(N1,N2)
=> sameAuthor(A1,A2) : 1.0
```

- Variable substitution produces a set of weighted ground rules for a particular data set

Soft Logic Relaxation

- PSL uses the Lukasiewicz t-norm to relax hard logic operators to work on soft truth values

$$\begin{aligned} a \tilde{\wedge} b &= \max\{0, a + b - 1\}, \\ a \tilde{\vee} b &= \min\{a + b, 1\}, \\ \tilde{\neg} a &= 1 - a, \end{aligned}$$

- PSL converts rules to logical statements using above operators

$$X \tilde{\Rightarrow} Y \equiv \tilde{\neg} X \tilde{\vee} Y.$$

FOL to CCMRF

- PSL converts a weighted rule into potential functions by penalizing its **distance to satisfaction**, $d(g,x) = (1 - t_g(x))$,
- $t_g(x)$ is the truth value of ground rule g under interpretation *x*
- The distribution over truth values is

$$\Pr(x) = \frac{1}{Z} \exp\left(\sum_{r \in P} \sum_{g \in G(r)} w_r d(g,x) \right)$$

w_r: weight of rule r

$G(r)$: all groundings of rule r

P : PSL program

PSL Inference

- PSL finds the most likely state by solving

$$\operatorname*{argmax}_{x} P(x) = \operatorname*{argmax}_{x} \sum_{r \in P} \sum_{g \in G(r)} w_r d(g, x)$$

- The t-norms defining $t_g(x)$ form linear constraints on *x*, making inference a linear program
- PSL uses lazy activation to ground rules, thus reducing the number of active variables and increasing efficiency
- Other distance metrics (e.g., Euclidean) for distance to satisfaction produce other types of convex objectives (e.g., quadratic programs)

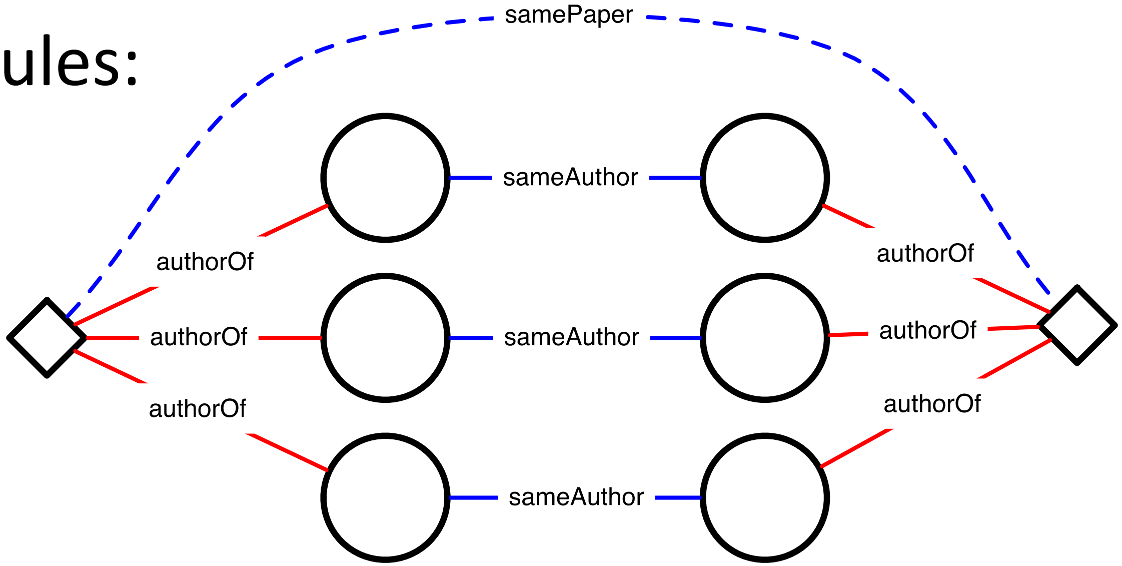
CiteSeer Example

- Citation listings collected from CiteSeer:
  - Pearl J. Probabilistic reasoning in intelligent systems.
    Pearl, Judea. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.
- Duplicate authors and papers
- Base model: Levenstein string similarity
  - authorName(A1,N1) ^ authorName(A2,N2) ^ similarString(N1,N2)
    => sameAuthor(A1,A2)

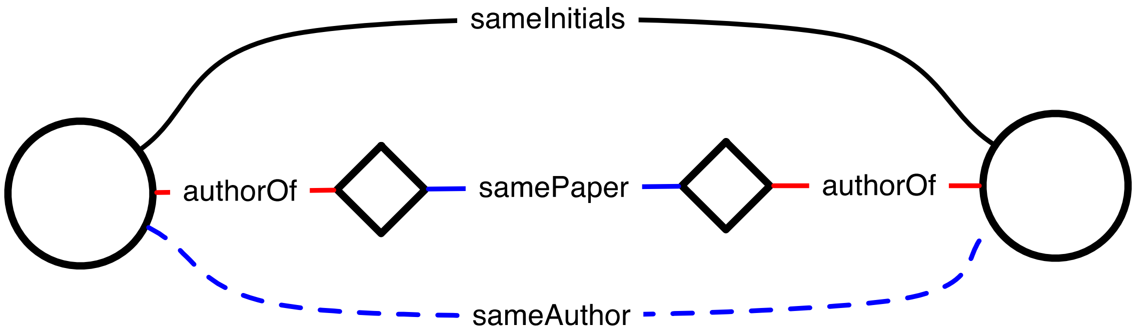
  - paperTitle(P1, T1) ^ paperTitle(P2,T2) ^ similarString(T1,T2)
    => samePaper(P1,P2)
- Only activate rule on pairs with similarity > 0.5

Reasoning about Sets

- Multi-Relational rules:
  - sameAuthorSet(P1,P2) => samePaper(P1,P2)



  - samePaper(P1,P2) ^ authorOf(A1,P1) ^ authorOf(A2,P2) ^ authorName(A1,N1) ^ authorName(A2,N2) ^ sameInitials(N1,N2) => sameAuthor(A1,A2)



Undirected Approaches

	Method	Learning/Inference Method	Evaluation
[McCallum & Wellner, NIPS04]	Conditional Random Fields (CRFs) capturing transitivity constraints	Graph partitioning (Boykov et al. 1999), performed via correlation clustering	F1 on DARPA MUC & ACE datasets
[Singla & Domingos, ICDM06]	Markov Logic Networks (MLNs)	Supervised learning and inference using MaxWalkSAT & MCMC	Conditional Log-likelihood and AUC on Cora and BibServ data
[Broecheler & Getoor, UAI10]	Probabilistic Similarity Logic (PSL)	Supervised learning and inference using continuous optimization	Precision/Recall /F1 Ontology Alignment

Summary: Collective Approaches

- Decisions for cluster-membership depends on other clusters
  - Non-probabilistic approaches
    - Similarity propagation approaches
    - Constraint-based approaches
  - Probabilistic Models
    - Generative Models
    - Undirected Models

- Advantages of non-probabilistic approaches is they often scale better than generative probabilistic approaches
- Undirected Models are often easier to specify
- Scaling undirected models active area of research

PART 3

BLOCKING/CANOPY GENERATION

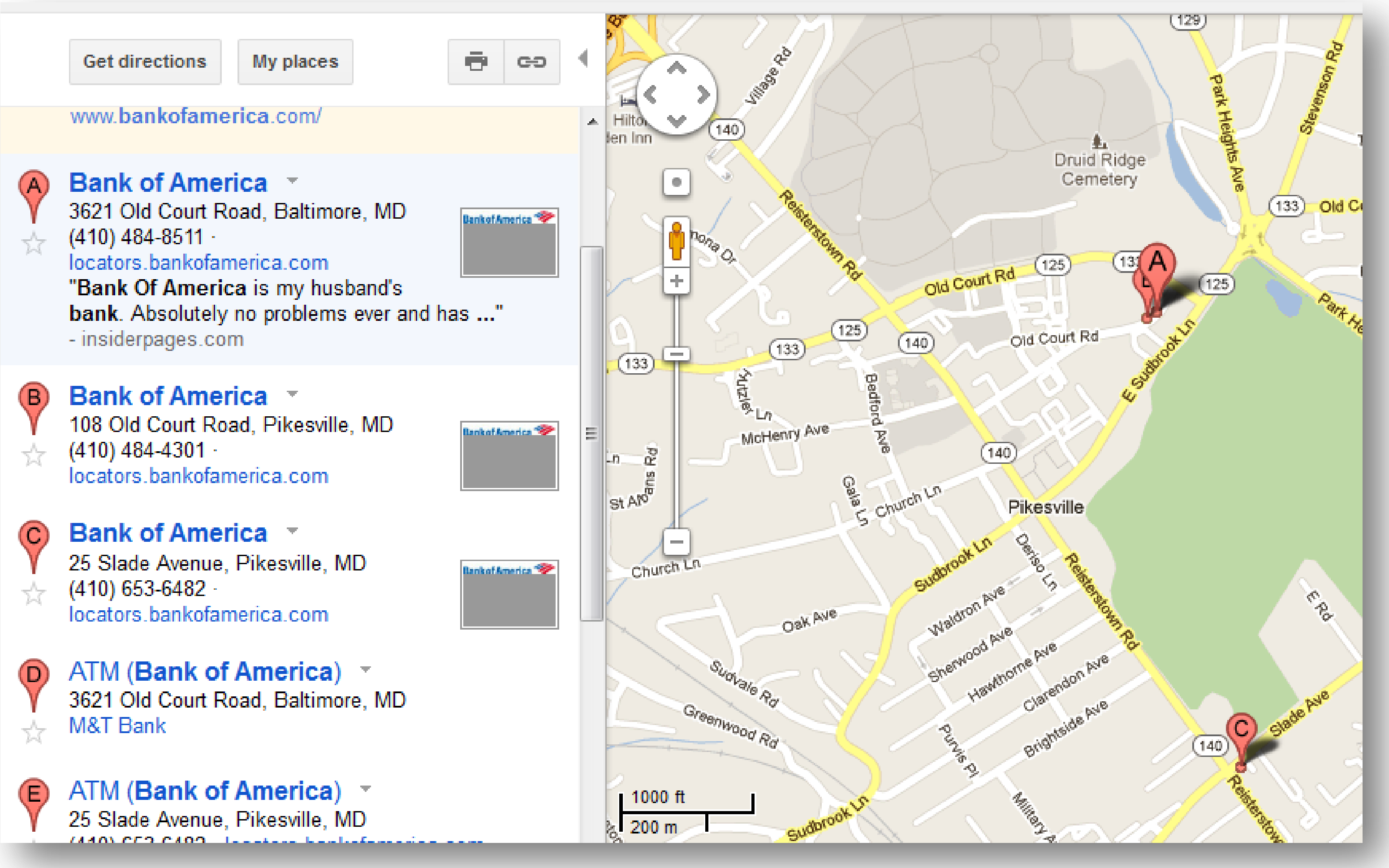
Blocking: Motivation

- Naïve pairwise: $|R|^2$ pairwise comparisons
  - 1000 business listings each from 1,000 different cities across the world
  - 1 trillion comparisons
  - 11.6 days (if each comparison is 1 μs)

- Mentions from different cities are unlikely to be matches
  - **Blocking Criterion: City**
  - 10 million comparisons
  - 10 seconds (if each comparison is 1 μs)

Blocking: Motivation

- Mentions from different cities are unlikely to be matches
  - May miss potential matches



www.bankofamerica.com/

A **Bank of America**
3621 Old Court Road, Baltimore, MD
(410) 484-8511 ·
locators.bankofamerica.com
"**Bank Of America** is my husband's **bank**. Absolutely no problems ever and has ..."
- insiderpages.com

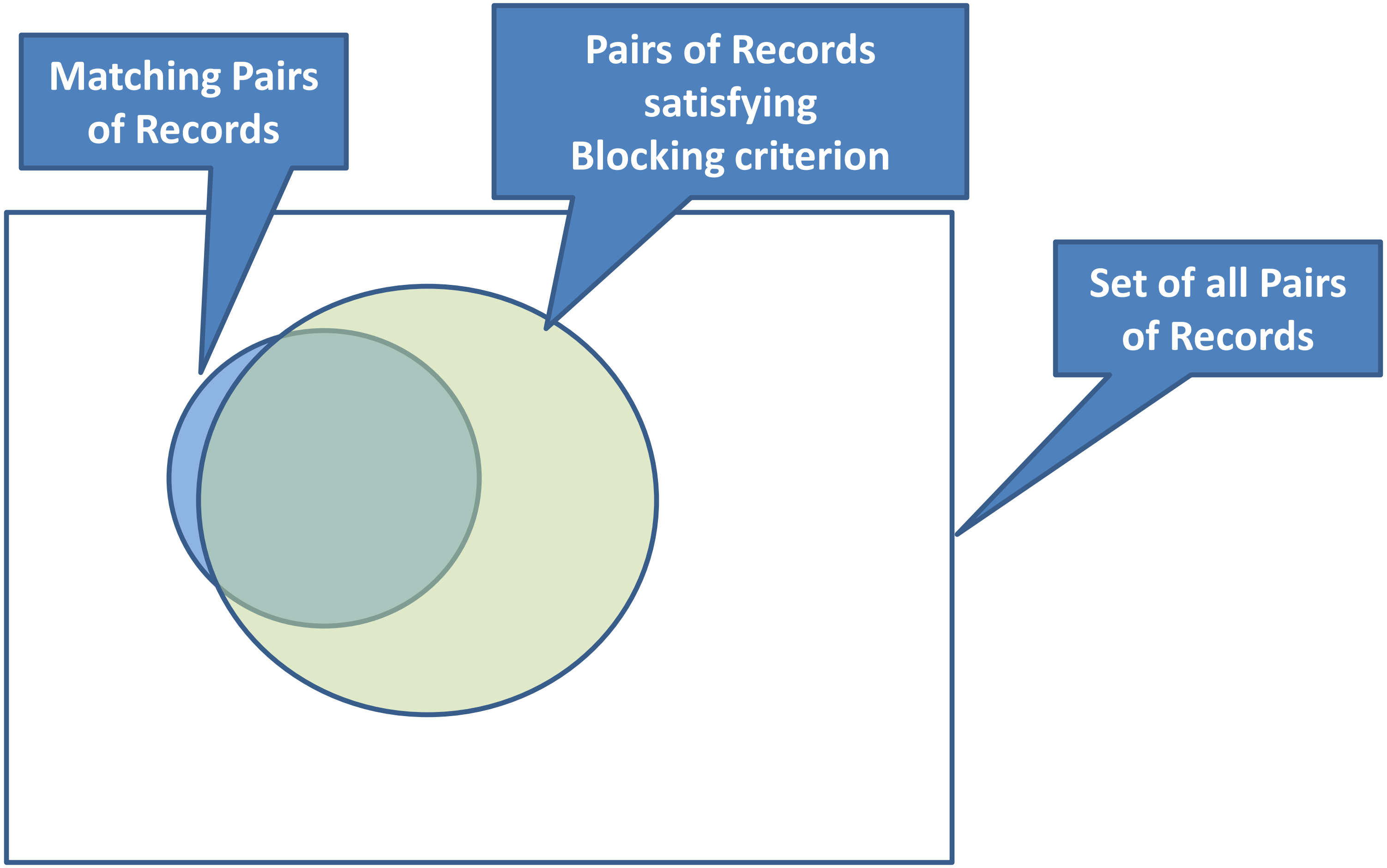
B **Bank of America**
108 Old Court Road, Pikesville, MD
(410) 484-4301 ·
locators.bankofamerica.com

C **Bank of America**
25 Slade Avenue, Pikesville, MD
(410) 653-6482 ·
locators.bankofamerica.com

D ATM (**Bank of America**)
3621 Old Court Road, Baltimore, MD
M&T Bank

E ATM (**Bank of America**)
25 Slade Avenue, Pikesville, MD
(410) 653-6482 · locators.bankofamerica.com

Blocking: Motivation


Matching Pairs of Records
Pairs of Records satisfying Blocking criterion
Set of all Pairs of Records

Blocking: Problem Statement

Input: Set of records *R*

Output: Set of *blocks/canopies*

$$\{C_1, C_2, \ldots, C_k\}, where\ \forall_i C_i \subset R\ and \bigcup_i C_i = R$$

Variants:

- *Disjoint Blocking*: Each mention appears in one block.

$$\forall_{i,j} C_i \cap C_j = \emptyset$$

- *Non-disjoint Blocking*: Mentions can appear in more than one block.

Blocking: Problem Statement

$$\{C_1, C_2, \ldots, C_k\}, where\ \forall_i C_i \subset R\ and \bigcup_i C_i = R$$

Metrics:

- Efficiency (or reduction ratio) : $\dfrac{number\ of\ pairs\ compared}{total\ number\ of\ pairs\ in\ R \times R}$

$$= \frac{|\{(x,y) \mid \exists i\ C_i, s.t.\ \ x,y \in C_i\}|}{r(r-1)/2}$$

- Recall* (or pairs completeness) : $\dfrac{number\ of\ true\ matches\ compared}{number\ of\ true\ matches\ in\ R \times R}$

**Need to know ground truth in order to compute this metric*

Blocking: Problem Statement

Metrics:

- Efficiency (or reduction ratio) : $\frac{number\ of\ pairs\ compared}{total\ number\ of\ pairs\ in\ R \times R}$

- Recall* (or pairs completeness) : $\frac{number\ of\ true\ matches\ compared}{number\ of\ true\ matches\ in\ R \times R}$

- Precision* (or pairs quality) : $\frac{number\ of\ true\ matches\ compared}{number\ of\ matches\ compared}$

- Max Canopy Size: $max_i\ |C_i|$

**Need to know ground truth in order to compute this metric*

Blocking Algorithms 1

- Hash based blocking
  - Each block C_i is associated with a hash key h_i.
  - Mention x is hashed to C_i if $hash(x) = h_i$.
  - Within a block, all pairs are compared.
  - Each hash function results in disjoint blocks.

- What *hash* function?
  - Deterministic function of attribute values
  - Boolean Functions over attribute values [Bilenko et al ICDM'06, Michelson et al AAAI'06, Das Sarma et al CIKM '12]
  - **minHash** (min-wise independent permutations) [Broder et al STOC'98]

Blocking Algorithms 2

- Pairwise Similarity/Neighborhood based blocking
  - Nearby nodes according to a similarity metric are clustered together
  - Results in non-disjoint canopies.

- Techniques
  - Sorted Neighborhood Approach [Hernandez et al SIGMOD'95]
  - Canopy Clustering [McCallum et al KDD'00]

Simple Blocking: Inverted Index on a Key

Examples of blocking keys:

- First three characters of last name
- City + State + Zip
- Character or Token n-grams
- Minimum infrequent n-grams

Learning Optimal Blocking Functions

- Using one or more blocking keys may be insufficient
  - 2,376,206 American's shared the surname Smith in the 2000 US
  - NULL values may create large blocks.

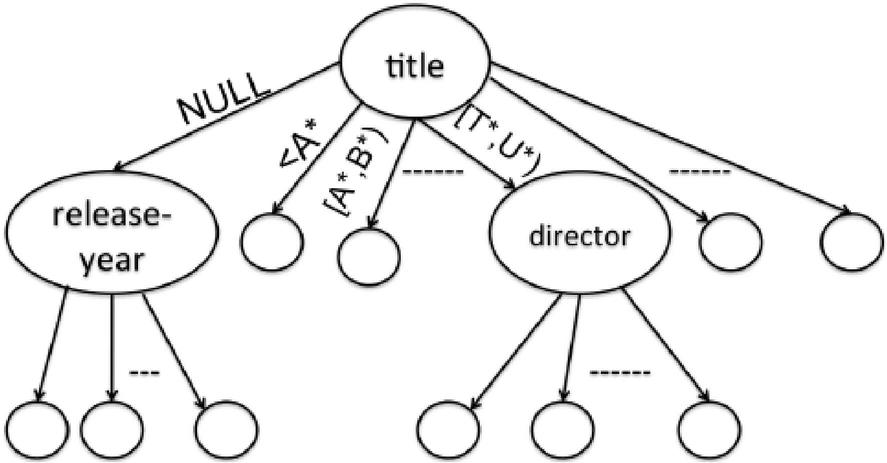
- Solution: Construct blocking functions by combining simple functions

Complex Blocking Functions

- Conjunction of functions [Michelson et al AAAI’06, Bilenko et al ICDM’06]
  - {City} AND {last four digits of phone}

- Chain-trees [Das Sarma et al CIKM‘12]
  - **If** ({City} = NULL or LA) **then** {last four digits of phone} AND {area code}
    else {last four digits of phone} AND {City}

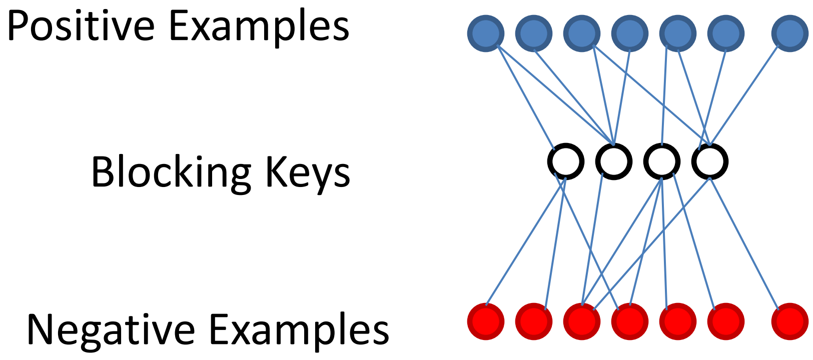
- BlkTrees [Das Sarma et al CIKM‘12]



Learning an Optimal function [Bilenko et al ICDM '06]

- Find k blocking functions that eliminate the most non-matches, while retaining almost all matches.
  - Need a training set of positive and negative pairs

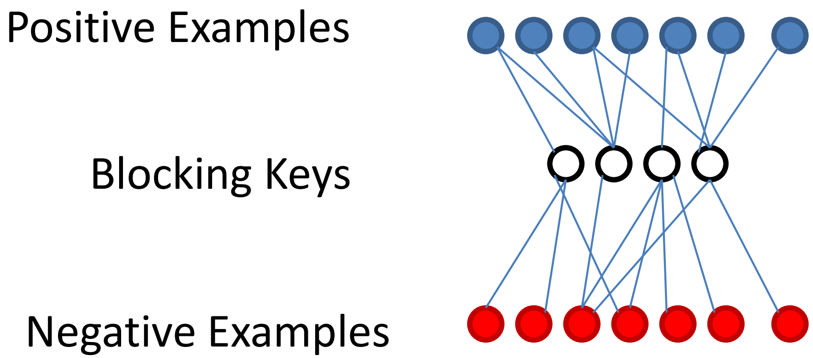
- Algorithm Idea: Red-Blue Set Cover



Pick k Blocking keys such that
(a) At most ε blue nodes are not covered
(b) Number of red nodes covered is minimized

Learning an Optimal function [Bilenko et al ICDM '06]

- Algorithm Idea: Red-Blue Set Cover



Pick k Blocking keys such that
(a) At most ε blue nodes are not covered
(b) Number of red nodes covered is minimized

- Greedy Algorithm:
  - Construct "good" conjunctions of blocking keys *{p$_1$, p$_2$, …}*.
  - Pick k conjunctions *{p$_{i1}$, p$_{i2}$, …, p$_{ik}$}*, such that the following is minimized

$$\frac{number\ of\ new\ blue\ nodes\ covered\ by\ p_{i_j}}{number\ of\ red\ nodes\ covered\ by\ p_{i_j}}$$

minHash (Minwise Independent Permutations)

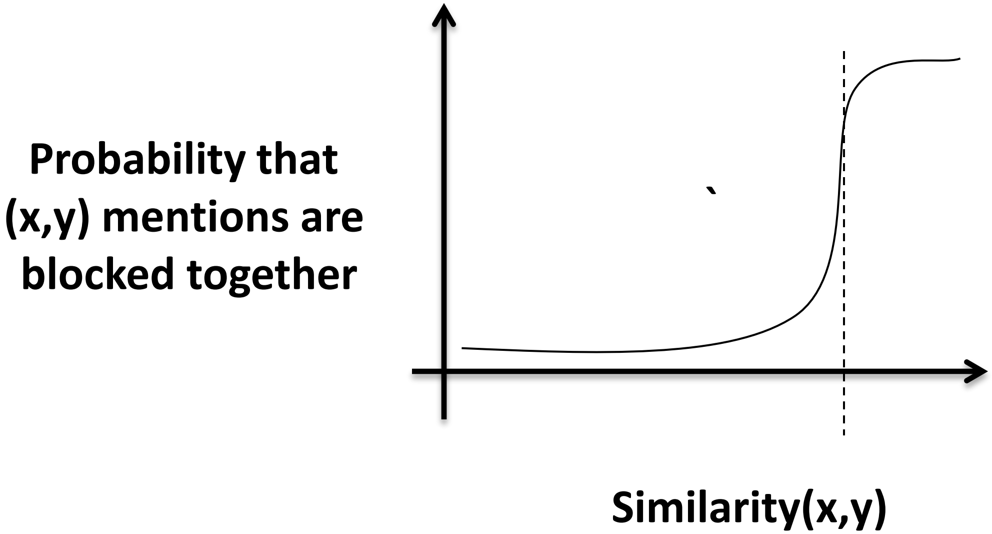
- Let F_x be a set of features for mention x
  - (functions of) attribute values
  - character ngrams
  - optimal blocking functions …
- Let π be a random permutation of features in F_x
  - E.g., order imposed by a random hash function
- *minHash(x)* = minimum element in F_x according to π

Why minHash works?

Surprising property: For a random permutation π,

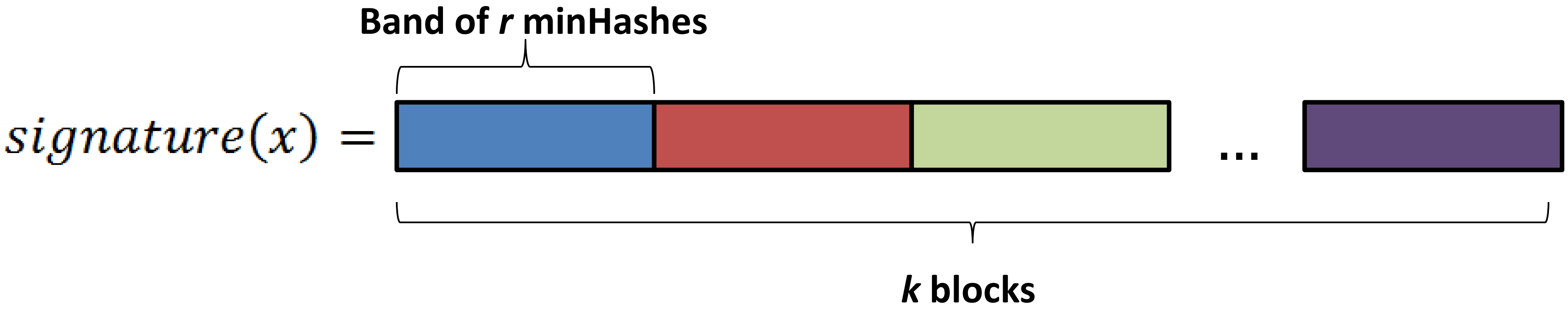
$$P(minHash(x) = minhash(y)) = \frac{F_x \cap F_y}{F_x \cup F_y}$$

How to build a blocking scheme such that only pairs with Jacquard similarity > s fall in the same block (with high prob)?



Blocking using minHashes

- Compute minHashes using *r* * *k* permutations (hash functions)



Band of *r* minHashes

$signature(x) =$

…

***k* blocks**

- Signature's that match on ***1 out of k*** bands, go to the same block.

minHash Analysis

False Negatives: (missing matches)

P(pair x,y not in the same block with Jacquard sim = s) $= (1 - s^r)^k$

should be very low for high similarity pairs

False Positives: (blocking non-matches)

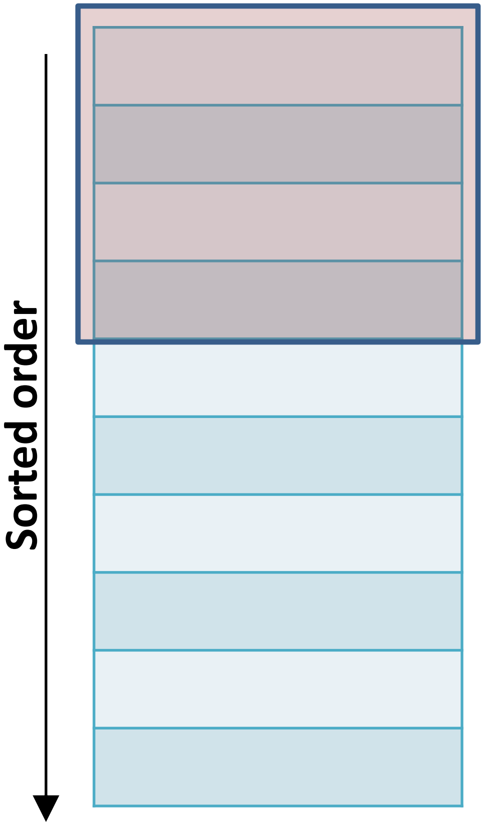
P(pair x,y in the same block with Jacquard sim = s) $= k \times s^r$

$r = 5, k = 20$

Sim(s)	P(not same block)
0.9	10^{-8}
0.8	0.00035
0.7	0.025
0.6	0.2
0.5	0.52
0.4	0.81
0.3	0.95
0.2	0.994
0.1	0.9998

Sorted Neighborhood [Hernandez et al SIGMOD'95]

- Compute a **Key** for each mention.
- **Sort** the mentions based on the key.
- **Merge**: Check whether a record matches with *(w-1)* previous records.
  - Efficient implementation using *Sort Merge Band Join* [DeWitt et al VLDB'91]
- Perform multiple passes with different keys

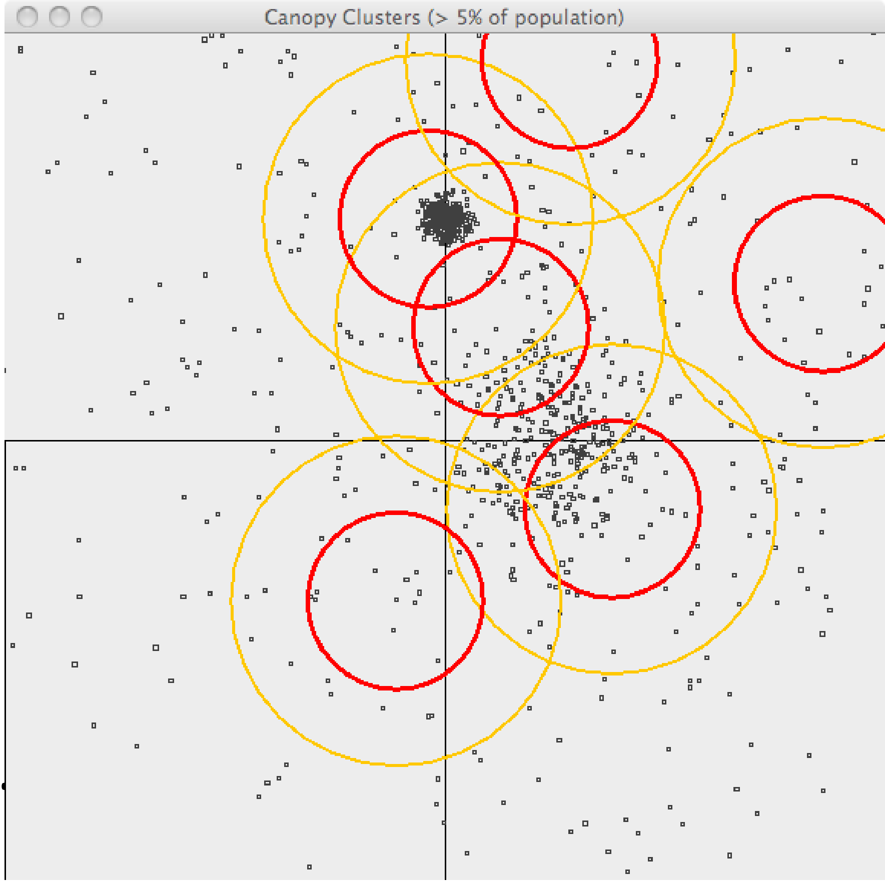


Canopy Clustering [McCallum et al KDD'00]

Input: Mentions *M*,
d(x,y), a distance metric,
thresholds $T_1 > T_2$

Algorithm:

1. Pick a random element *x* from *M*
2. Create new canopy C_x using mentions y s.t. $d(x,y) < T_1$
3. Delete all mentions *y* from *M* s.t. $d(x,y) < T_2$
4. Return to Step 1 if *M* is not empty.



SCALING COLLECTIVE ER

Scaling Collective ER [Rastogi et al VLDB11]

Current state-of-the-art: **Collective Entity Matching**

(+) High ***accuracy***

(-) Often scale only to a few 1000 entities [SD06], since runtime is quadratic in the number of pairs.

Example: Dedup papers and authors

Id	**Author-1**	**Author-2**	**Paper**
A_1	John Smith	Richard Johnson	Indices and Views
A_2	J Smith	R Johnson	SQL Queries
A_3	Dr. Smyth	R Johnson	Indices and Views

Slides adapted from [Rastogi et al VLDB11] talk

Algorithm

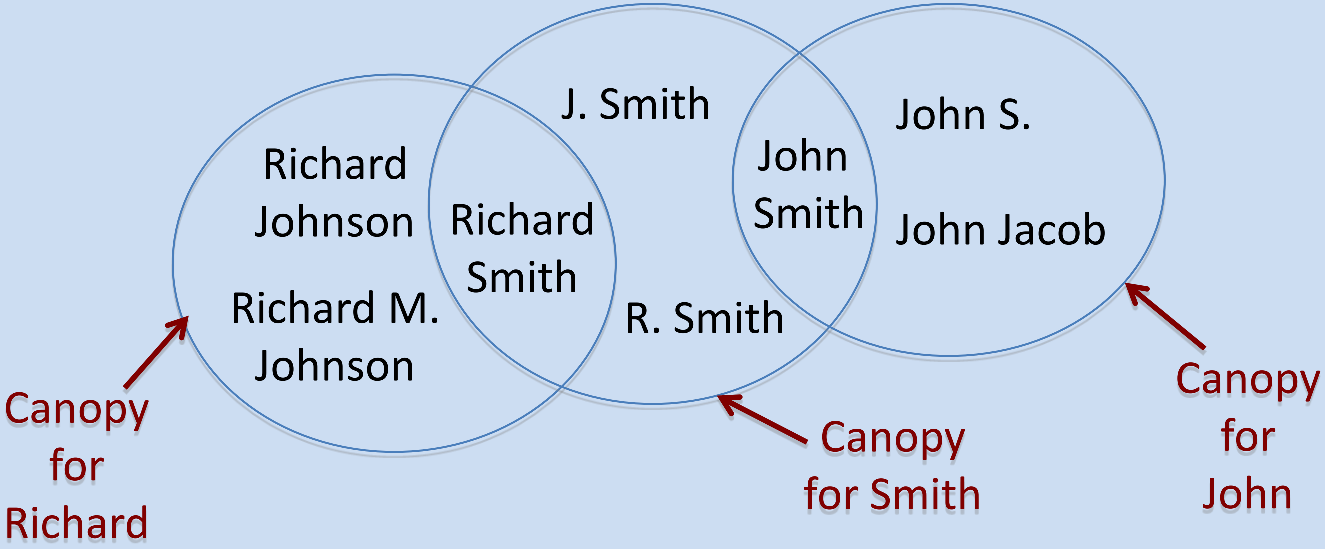
- Generates overlapping canopies (e.g., Canopy clustering)

- Run collective matcher on each canopy

Efficiency: Use Canopies[McCallum et al KDD 00]

Reduces # of candidate pairs from:

O(|Mentions|^2) to |Candidate pairs|



Pair-wise approach becomes efficient: O(|Candidate pairs|)

Efficiency of Collective approach

Collective methods still not efficient: $\Omega(|\text{Candidate pairs}|^2)$

Example for Collective methods[SD06]

- |References|= 1000, |Candidate pairs| = 15,000,
  - Time ~ 5 minutes
- |References| = 50,000, |Candidate pairs| = 10 million
  - Time required = 2,500 hours ~ 3 months

Distribute

Run collective entity-matching in each canopy separately

Example for Collective methods[SD06]

- |References|= 1000, |Candidates| = 15,000,
  - Time = 5 minutes
- One canopy: |References| = 100, |Candidates| ~ 1000,
  - Time ~ 10 Seconds
- |References| = 50,000, # of canopies ~ 13k
  - Time ~ 20 hours << 3 months!

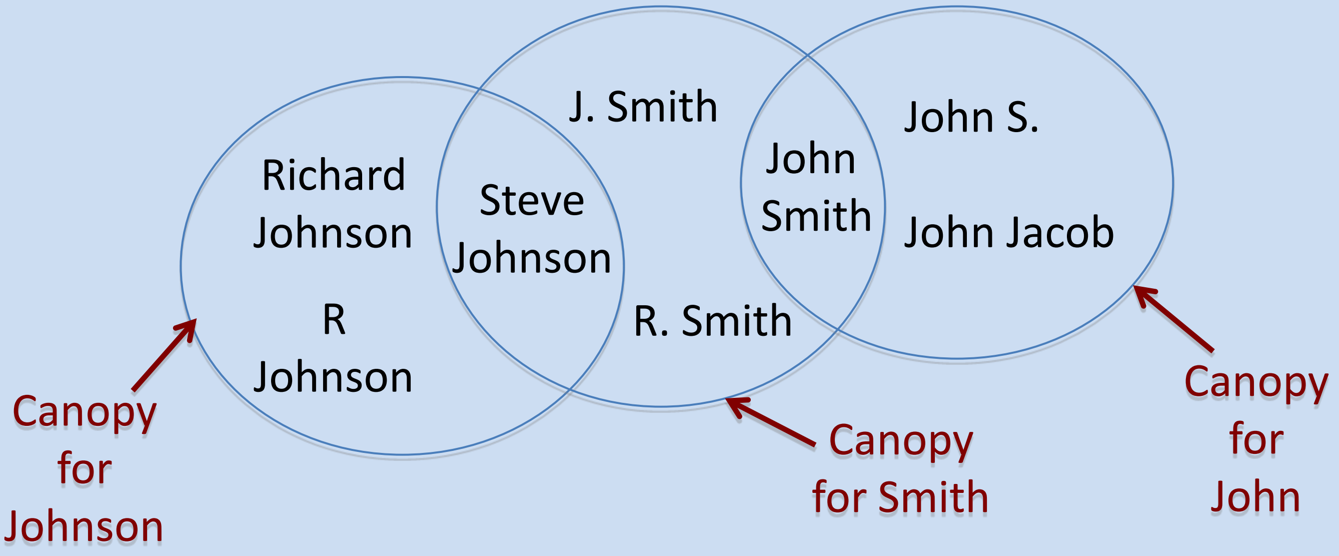
Partitioning into smaller chunks helps!

Problem: Correlations across canopies will be lost

$\text{CoAuthor}(A_1, B_1) \wedge \text{CoAuthor}(A_2, B_2) \wedge \text{match}(B_1, B_2) \rightarrow \text{match}(A_1, A_2)$

Example: CoAuthor rule grounds to the correlation

match(Richard Johnson, R Johnson) => match(J. Smith, John Smith)



Message Passing

Simple Message Passing (SMP)

1. Run entity matcher M locally in each canopy
2. If M finds a match(r_1,r_2) in some canopy, pass it as evidence to all canopies
3. Rerun M within each canopy using new evidence
4. Repeat until no new matches found in each canopy

Runtime: $O(k^2 f(k) c)$

- k : maximum size of a canopy
- f(k): Time taken by ER on canopy of size k
- c : number of canopies

Formal Properties

for a well behaved ER method …

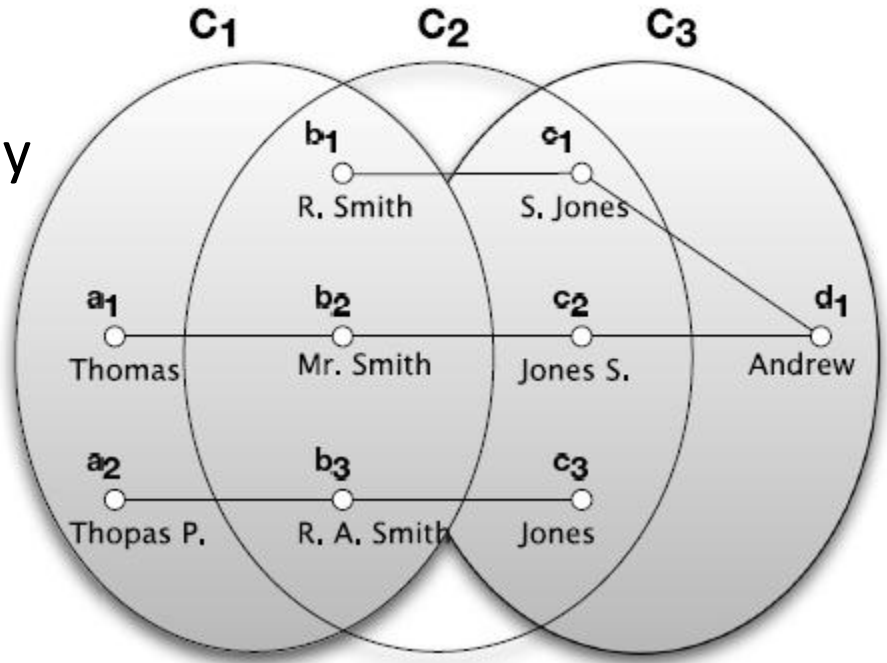
Convergence: No. of steps ≤ no. of matches

Consistency: Output independent of the canopy order

Soundness: Each output match is actually a true match

~~***Completeness***: Each true match is also a output match~~

Completeness



Papers 2 and 3 match only if a canopy knows that

- match(a1,a2)
- match(b2,b3)
- match(c2,c3)

Simple message passing will not find any matches

- thus, no messages are passed, no progress

Solution: Maximal message passing

- **Send a message if there is a potential for match**

Summary of Blocking

- $O(|R|^2)$ pairwise computations can be prohibitive.
  - Blocking eliminates comparisons on a large fraction of non-matches.
- Equality-based Blocking:
  - Construct (one or more) blocking keys from features
  - Records not matching on any key are not compared.
- Similarity based Blocking:
  - Form overlapping canopies of records based on similarity.
  - Only compare records within a cluster.
- Message Passing + blocking can help scale collective ER.

Part 4

CHALLENGES AND FUTURE DIRECTIONS

Challenges

- So far, we have viewed ER as a one-time process applied to entire database; none of these hold in real world.
- Temporal ER
  - ER algorithms need to account for change in real world
  - Reasoning about multiple sources [Pal & **M** et al. WWW 12]
  - Model transitions [Li et al VLDB11]
- Reasoning about source quality
  - Sources are not independent
  - Copying Problem [Dong et al VLDB09]
- Query Time ER
  - How do we selectively determine the smallest number of records to resolve, so we get accurate results for a particular query?
  - Collective resolution for queries [Bhattacharya & Getoor JAIR07]

Open Issues

- ER & User-generated data
  - Deduplicated entities interact with users in the real world
    - Users tag/associate photos/reviews with businesses on Google / Yahoo
  - What should be done to support interactions?
- ER is often part of bigger inference problem
  - Pipelined approaches and joint approaches to information extraction and graph identification
  - How can we characterize how ER errors affect overall quality of results?
- ER Theory
  - Need better support for theory which can give relational learning bounds
- ER & Privacy
  - ER enables record re-identification
  - How do we develop a theory of privacy-preserving ER?

Summary

- Growing omnipresence of massive linked data, and the need for creating knowledge bases from text and unstructured data motivate a number of challenges in ER

- Especially interesting challenges and opportunities for ER and social media data

- As data, noise, and knowledge grows, greater needs & opportunities for intelligent reasoning about entity resolution

- Many other challenges
  - Large scale identity management
  - Understanding theoretical potentials & limits of ER

THANK YOU!

References – Intro

W. Willinger et al, “Mathematics and the Internet: A Source of Enormous Confusion and Great Potential”, Notices of the AMS 56(5), 2009

L. Gill and M. Goldcare, “English National Record Linkage of Hospital Episode Statistics and Death Registration Records”, Report to the Department of Health, 2003

References – Single Entity ER

D. Menestrina et al, “Evaluation Entity Resolution Results”, PVLDB 3(1-2), 2010

M. Cochinwala et al, “Efficient data reconciliation”, Information Sciences 137(1-4), 2001

M. Bilenko & R. Mooney, “Adaptive Duplicate Detection Using Learnable String Similarity Measures”, KDD 2003

P. Christen, “Automatic record linkage using seeded nearest neighbour and support vector machine classification.”, KDD 2008

Z. Chen et al, “Exploiting context analysis for combining multiple entity resolution systems”, SIGMOD 2009

A. McCallum & B. Wellner, “Conditional Models of Identity Uncertainty with Application to Noun Coreference”, NIPS 2004

H. Newcombe et al, “Automatic linkage of vital records”, Science 1959

I. Fellegi & A. Sunter, “A Theory for Record Linkage”, JASA 1969

W. Winkler, “Overview of Record Linkage and Current Research Directions”, Research Report Series, US Census, 2006

T. Herzog et al, “Data Quality and Record Linkage Techniques”, Springer, 2007

P. Ravikumar & W. Cohen, “A Hierarchical Graphical Model for Record Linkage”, UAI 2004

S. Sarawagi et al, “Interactive Deduplication using Active Learning”, KDD 2000

S. Tejada et al, “Learning Object Identification Rules for Information Integration”, IS 2001

A. Arasu et al, “On active learning of record matching packages”, SIGMOD 2010

K. Bellare et al, “Active sampling for entity matching”, KDD 2012

A. Beygelzimer et al, “Agnostic Active Learning without Constraints”, NIPS 2010

References – Single Entity ER (contd.)

R. Gupta & S. Sarawagi, “Answering Table Augmentation Queries from Unstructured Lists on the Web”, PVLDB 2(1), 2009

A. Das Sarma et al, “An Automatic Blocking Mechanism for Large-Scale De-duplication Tasks”, CIKM 2012

M. Bilenko et al, “Adaptive Product Normalization: Using Online Learning for Record Linkage in Comparison Shopping”, ICDM 2005

S. Chaudhuri et al, “Robust Identification of Fuzzy Duplicates”, ICDE 2005

W. Soon et al, “A machine learning approach to coreference resolution of noun phrases”, Computational Linguistics 27(4) 2001

N. Bansal et al, “Correlation Clustering”, Machine Learning 56(1-3), 2004

V. Ng & C. Cardie, “Improving machine learning approaches to coreference resolution”, ACL 2002

M. Elsner & E. Charnaik, “You talking to me? a corpus and algorithm for conversation disentanglement”, ACL-HLT 2008

M. Elsner & W. Schudy, “Bounding and Comparing Methods for Correlation Clustering Beyond ILP”, ILP-NLP 2009

N. Ailon et al, “Aggregating inconsistent information: Ranking and clustering”, JACM 55(5), 2008

X. Dong et al, “Integrating Conflicting Data: The Role of Source Dependence”, PVLDB 2(1), 2009

A. Pal et al, “Information Integration over Time in Unreliable and Uncertain Environments”, WWW 2012

A. Culotta et al, “Canonicalization of Database Records using Adaptive Similarity Measures”, KDD 2007

O. Benjelloun et al, “Swoosh: A generic approach to Entity Resolution”, VLDBJ 18(1), 2009

References – Multi-Relational ER

A. Arasu et al, “Large-Scale Deduplication with Constraints using Dedupalog”, ICDE 2009

X. Dong et al, “Reference Recounciliation in Complex Information Spaces”, SIGMOD 2005

I. Bhattacharya & L. Getoor, “Collective Entity Resolution in Relational Data”, TKDD 2007

I. Bhattacharya & L. Getoor, “A Latent Dirichlet Model for Unsupervised Entity Resolution “, SDM 2007

M. Broecheler & L. Getoor , “Probabilistic Similarity Logic”, UAI 2010

H. Pasula et al , “Identity Uncertainty and Citation Matching”, NIPS 2002

D. Kalashnikov et al, “Domain-Independent Data Cleaning via Analysis of Entity-Relationship Graph”, TODS06

J. Lafferty et al, “Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data.”, ICML 2001

X. Li et al, “Identification and Tracing of Ambiguous Names: Discriminative and Generative Approaches”, AAAI 2004

A. McCallum & B. Wellner, “Conditional Models of Identity Uncertainty with Application to Noun Coreference”, NIPS 2004

M. Richardson & P. Domingos, “Markov Logic”, Machine Learning 62, 2006

W. Shen et al, “Constraint-based Entity Matching”, AAAI 2005

P. Singla & P. Domingos, “Entity Resolution with Markov Logic”, ICDM 2006

References – Blocking

M. Bilenko et al, “Adaptive Blocking: Learning to Scale Up Record Linkage and Clustering”, ICDM 2006

M. Michelson & C. Knoblock, “Learning Blocking Schemes for Record Linkage”, AAAI 2006

A. Das Sarma et al, “An Automatic Blocking Mechanism for Large-Scale De-duplication Tasks”, CIKM 2012

A. Broder et al, “Min-Wise Independent Permutations”, STOC 1998

M. Hernandez & S. Stolfo, “The merge/purge problem for large databases”, SIGMOD 1995

A. McCallum et al, “Efficient clustering of high-dimensional data sets with application to reference matching”, KDD 2000

V. Rastogi et al, “Large-Scale Collective Entity Matching”, PVLDB 4(4), 2011

References – Challenges & Future Directions

I. Bhattacharya and L. Getoor, "Query-time Entity Resolution", JAIR 2007

X. Dong, L. Berti-Equille, D. Srivastava, ”Truth discovery and copying detection in a dynamic world”, VLDB 2009

P. Li, X. Dong, A. Maurino, D. Srivastava, “Linking Temporal Records”, VLDB 2011

A. Pal, V. Rastogi, A. Machanavajjhala, P. Bohannon, “Information integration over time in unreliable and uncertain environments”, WWW 2012