

# 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\\_VLDB2012.pdf](http://www.cs.umd.edu/~getoor/Tutorials/ER_VLDB2012.pdf)

<http://goo.gl/f5eym>

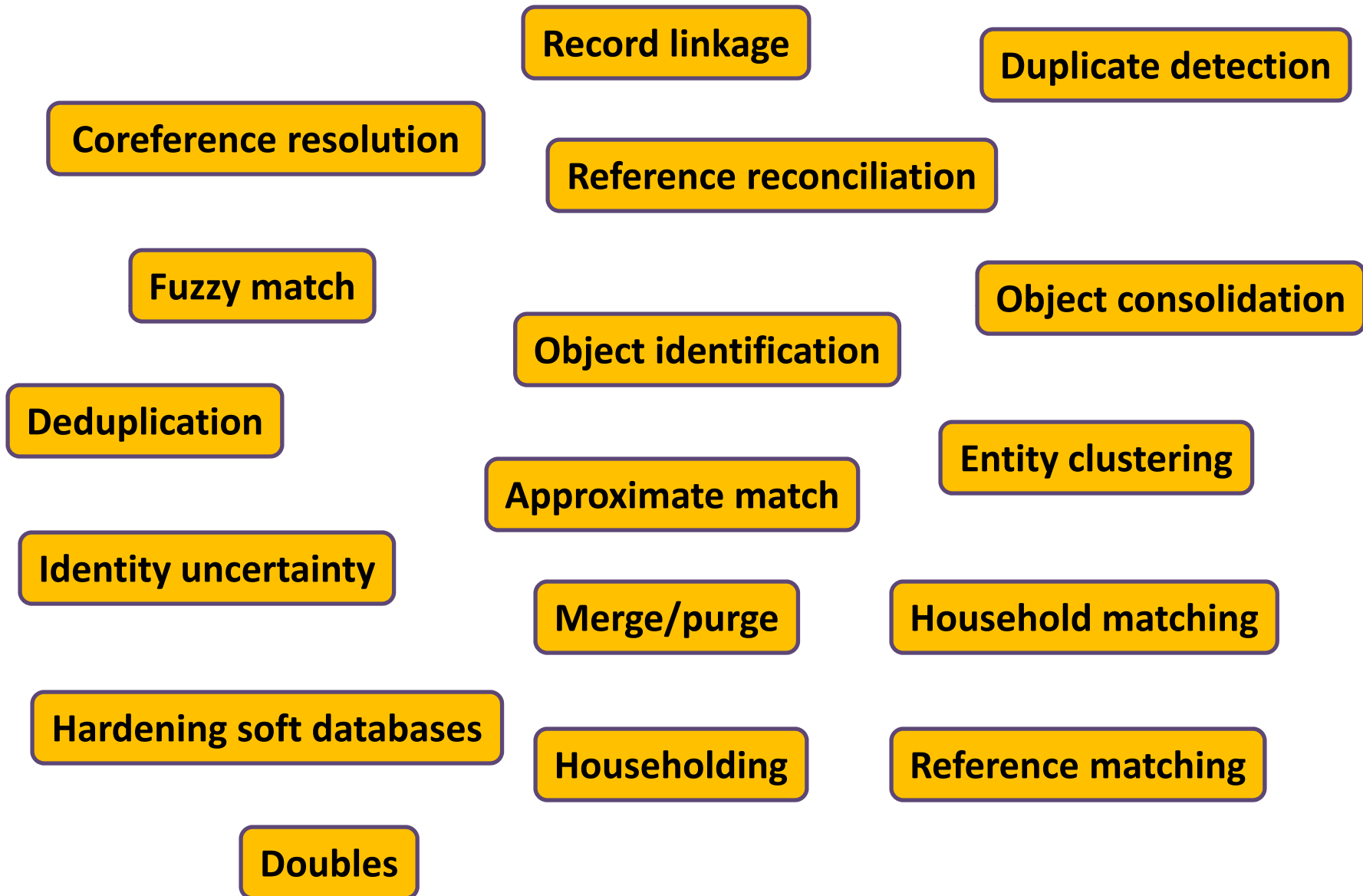
# What is Entity Resolution?

*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.
- ...

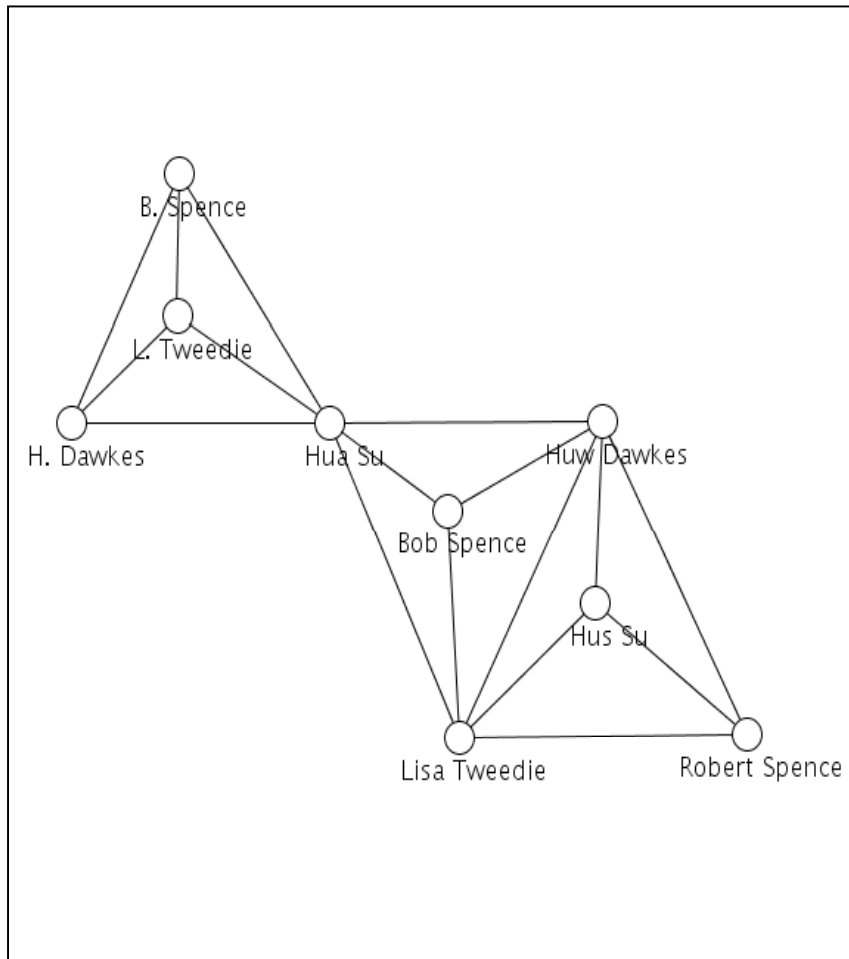
# Ironically, Entity Resolution has many duplicate names



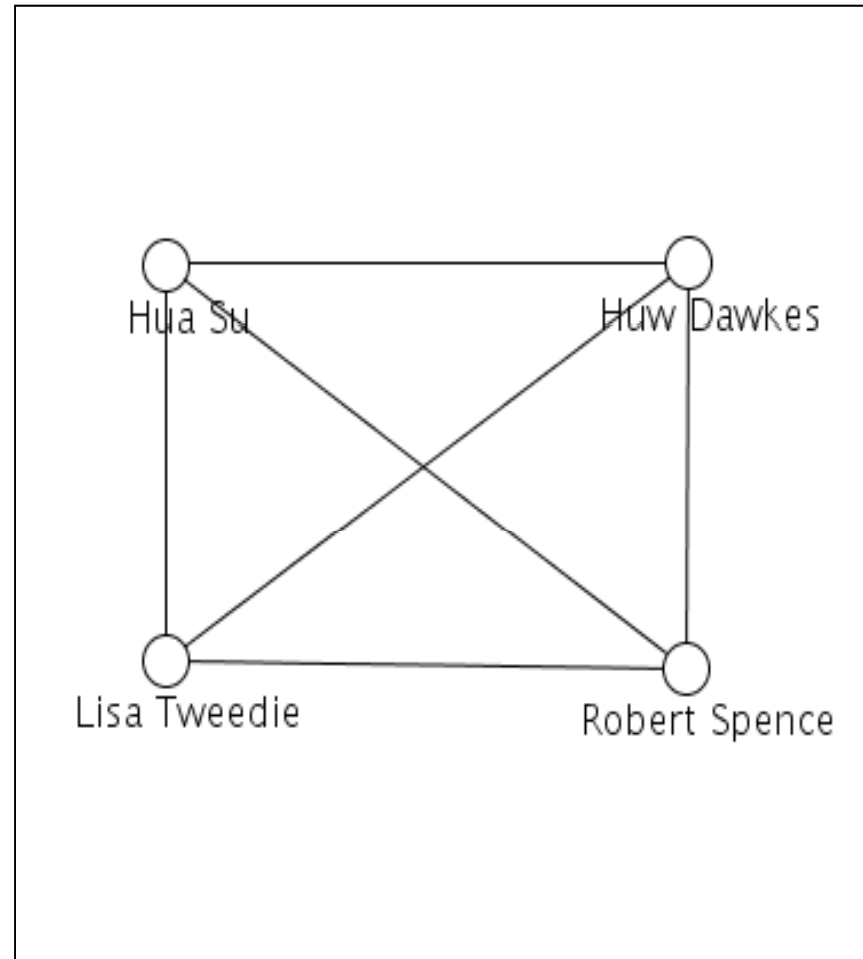
# ER Motivating Examples

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

# ER and Network Analysis



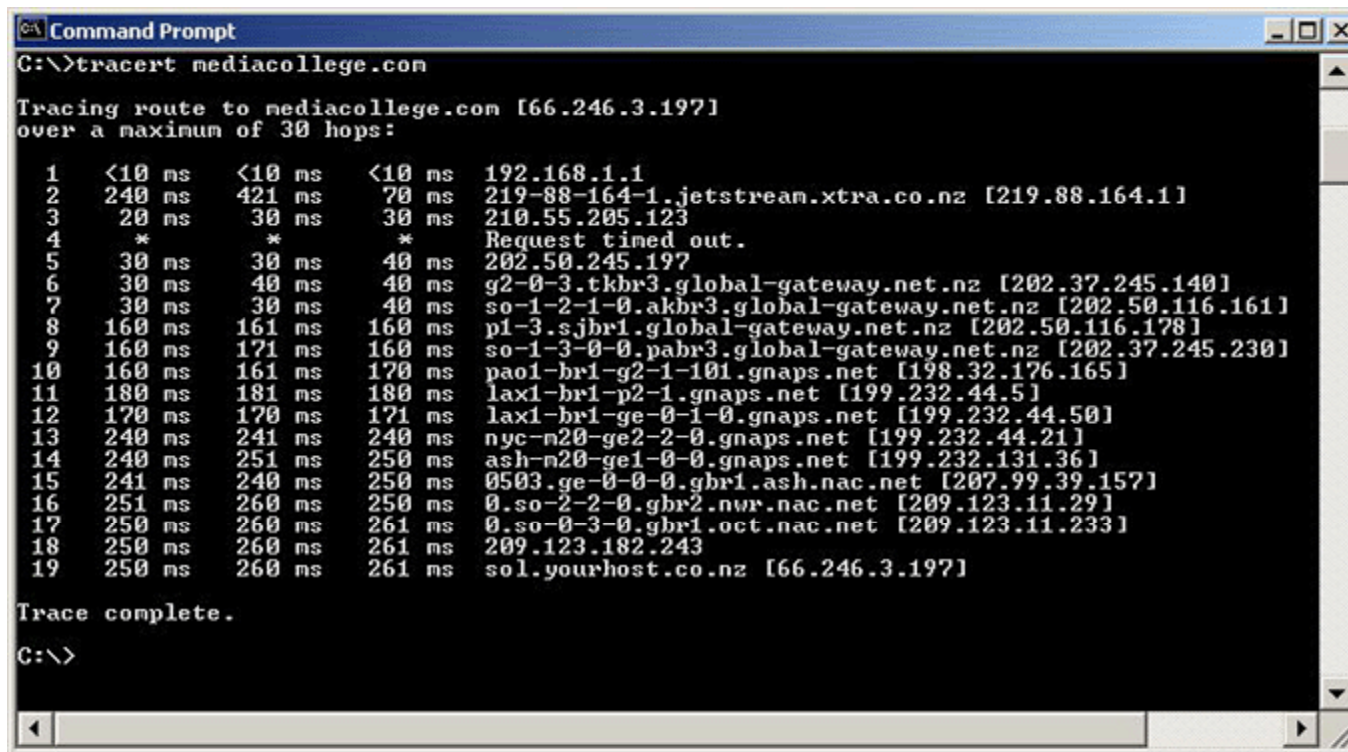
before



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:

  0  <10 ms  <10 ms  <10 ms  192.168.1.1
  1  240 ms  421 ms  70 ms  219-88-164-1.jetstream.xtra.co.nz [219.88.164.1]
  2  20 ms  30 ms  30 ms  210.55.205.123
  3  *      *      *      Request timed out.
  4  30 ms  30 ms  40 ms  202.50.245.197
  5  30 ms  40 ms  40 ms  g2-0-3.tkbr3.global-gateway.net.nz [202.37.245.140]
  6  30 ms  30 ms  40 ms  so-1-2-1-0.akbr3.global-gateway.net.nz [202.50.116.161]
  7  160 ms  161 ms  160 ms  p1-3.sjbr1.global-gateway.net.nz [202.50.116.178]
  8  160 ms  171 ms  160 ms  so-1-3-0-0.pabr3.global-gateway.net.nz [202.37.245.230]
  9  160 ms  161 ms  170 ms  pao1-br1-g2-1-101.gnaps.net [198.32.176.165]
 10  180 ms  181 ms  180 ms  lax1-br1-p2-1.gnaps.net [199.232.44.5]
 11  170 ms  170 ms  171 ms  lax1-br1-ge-0-1-0.gnaps.net [199.232.44.50]
 12  240 ms  241 ms  240 ms  nyc-n20-ge2-2-0.gnaps.net [199.232.44.21]
 13  240 ms  251 ms  250 ms  ash-n20-ge1-0-0.gnaps.net [199.232.131.36]
 14  241 ms  240 ms  250 ms  0503.ge-0-0-0.gbr1.ash.nac.net [207.99.39.157]
 15  251 ms  260 ms  250 ms  0.so-2-2-0.gbr2.nwr.nac.net [209.123.11.29]
 16  250 ms  260 ms  261 ms  0.so-0-3-0.gbr1.oct.nac.net [209.123.11.233]
 17  250 ms  260 ms  261 ms  209.123.182.243
 18  250 ms  260 ms  261 ms  sol.yourhost.co.nz [66.246.3.197]
 19

Trace complete.

C:\>
```

# IP Aliasing Problem [Willinger et al. 2009]

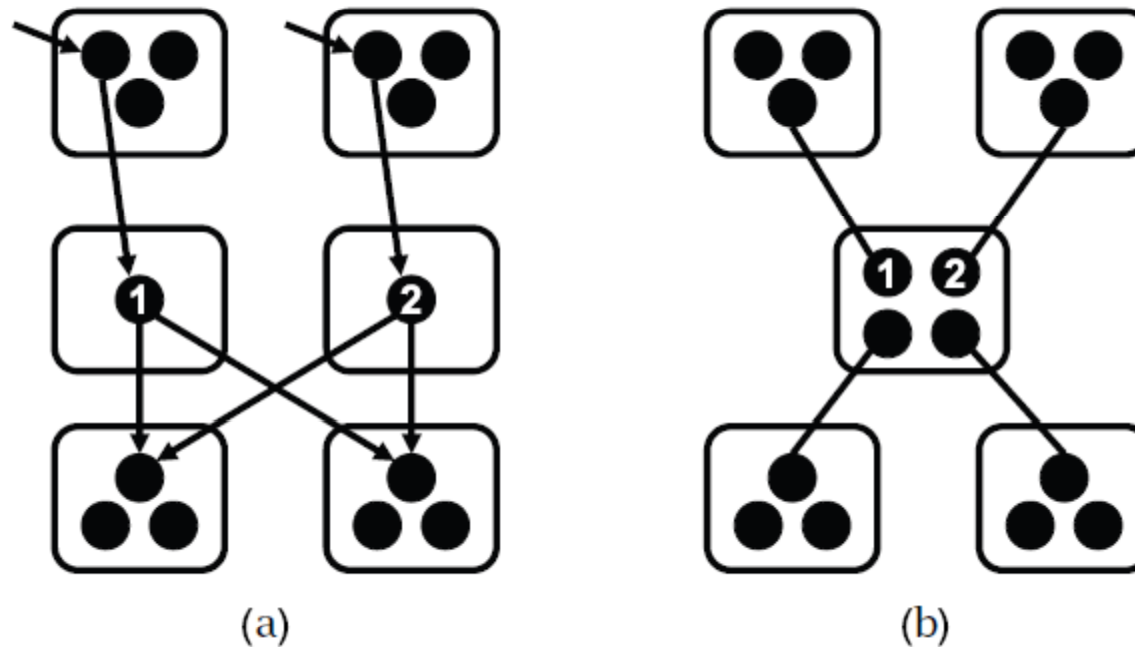


Figure 2. The IP alias resolution problem. Paraphrasing Fig. 4 of [50], traceroute does not list routers (boxes) along paths but IP addresses of input interfaces (circles), and alias resolution refers to the correct mapping of interfaces to routers to reveal the actual topology. In the case where interfaces 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]

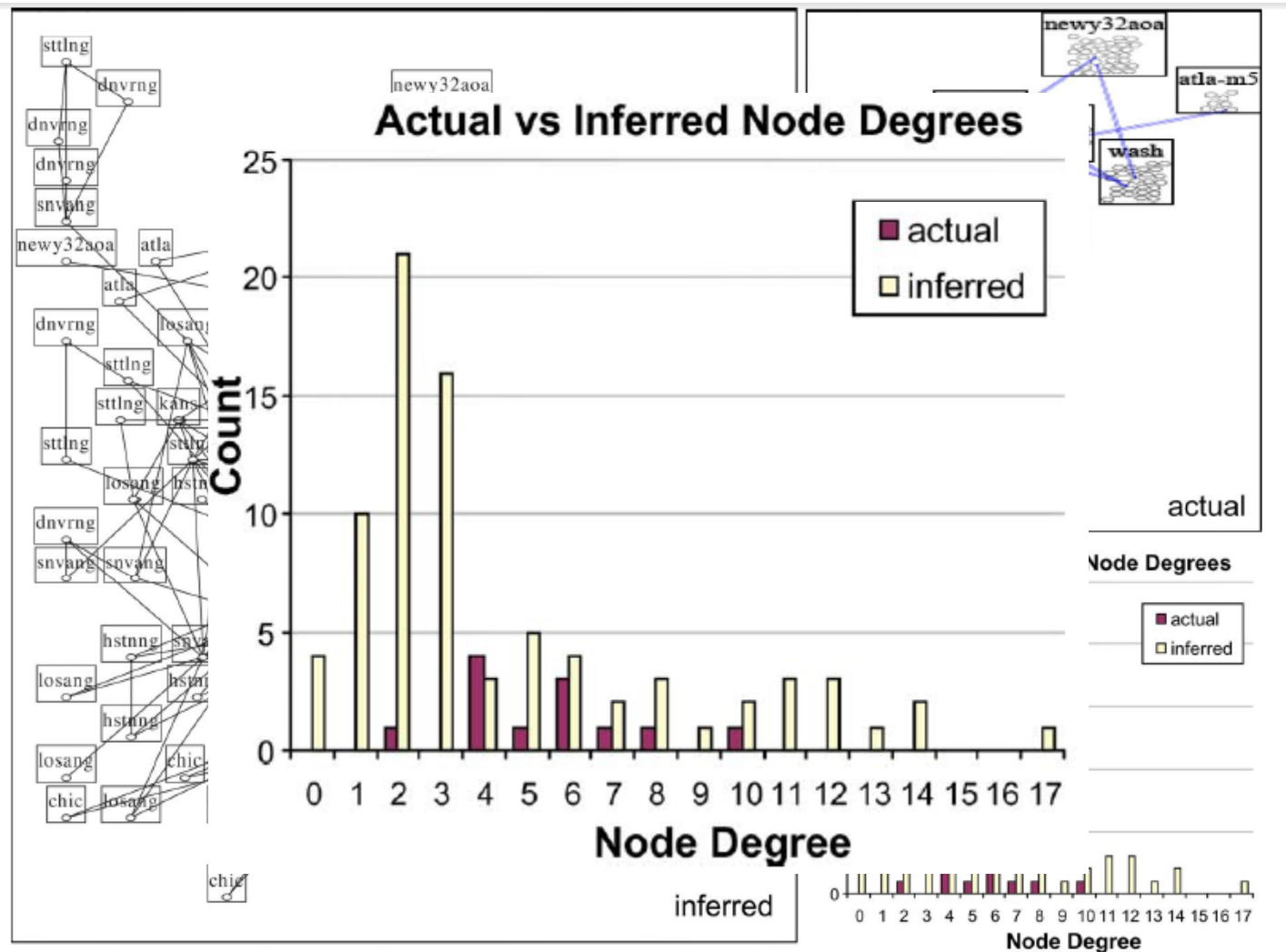


Figure 3. The IP alias resolution problem in practice. This is re-produced from [48] and shows a comparison between the Abilene/Internet2 topology inferred by Rocketfuel (left) and the actual topology (top right). Rectangles represent routers with interior ovals denoting interfaces. The histograms of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



↓	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



- Abbreviations / Data Truncation





# 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. Abstract Problem Statement
2. Algorithmic Foundations of ER
3. Scaling ER to Big-Data
4. Challenges & Future Directions



# Outline

1. Abstract Problem Statement
2. Algorithmic Foundations of ER
  - a) Data Preparation and Match Features
  - b) Pairwise ER
  - c) Constraints in ER
  - d) Algorithms
    - Record Linkage
    - Deduplication
    - Collective ER
3. Scaling ER to Big-Data
4. Challenges & Future Directions



5 minute break

# Outline

1. Abstract Problem Statement
2. Algorithmic Foundations of ER
- 3. Scaling ER to Big-Data**
  - a) Blocking/Canopy Generation
  - b) Distributed ER
4. Challenges & Future Directions

# Outline

1. Abstract Problem Statement
2. Algorithmic Foundations of ER
3. Scaling ER to Big-Data
4. Challenges & Future Directions

# Scope of the Tutorial

- What we cover:
  - Fundamental algorithmic concepts in ER
  - Scaling ER to big datasets
  - Taxonomy of current ER algorithms
- What we do not cover:
  - Schema/ontology resolution
  - Data fusion/integration/exchange/cleaning
  - Entity/Information Extraction
  - Privacy aspects of Entity Resolution
  - Details on similarity measures
  - Technical details and proofs

# ER References

- Book / Survey Articles
  - Data Quality and Record Linkage Techniques [T. Herzog, F. Scheuren, W. Winkler, Springer, '07]
  - Duplicate Record Detection [A. Elmagrid, P. Ipeirotis, V. Verykios, TKDE '07]
  - An Introduction to Duplicate Detection [F. Naumann, M. Herschel, M&P synthesis lectures 2010]
  - Evaluation of Entity Resolution Approached on Real-world Match Problems [H. Kopke, A. Thor, E. Rahm, PVLDB 2010]
  - Data Matching [P. Christen, Springer 2012]
- Tutorials
  - Record Linkage: Similarity measures and Algorithms [N. Koudas, S. Sarawagi, D. Srivatsava SIGMOD '06]
  - Data fusion--Resolving data conflicts for integration [X. Dong, F. Naumann VLDB '09]
  - Entity Resolution: Theory, Practice and Open Challenges <http://goo.gl/Ui38o> [L. Getoor, A. Machanavajjhala AAAI '12]

PART 1

# **ABSTRACT PROBLEM STATEMENT**

# Abstract Problem Statement

Real World



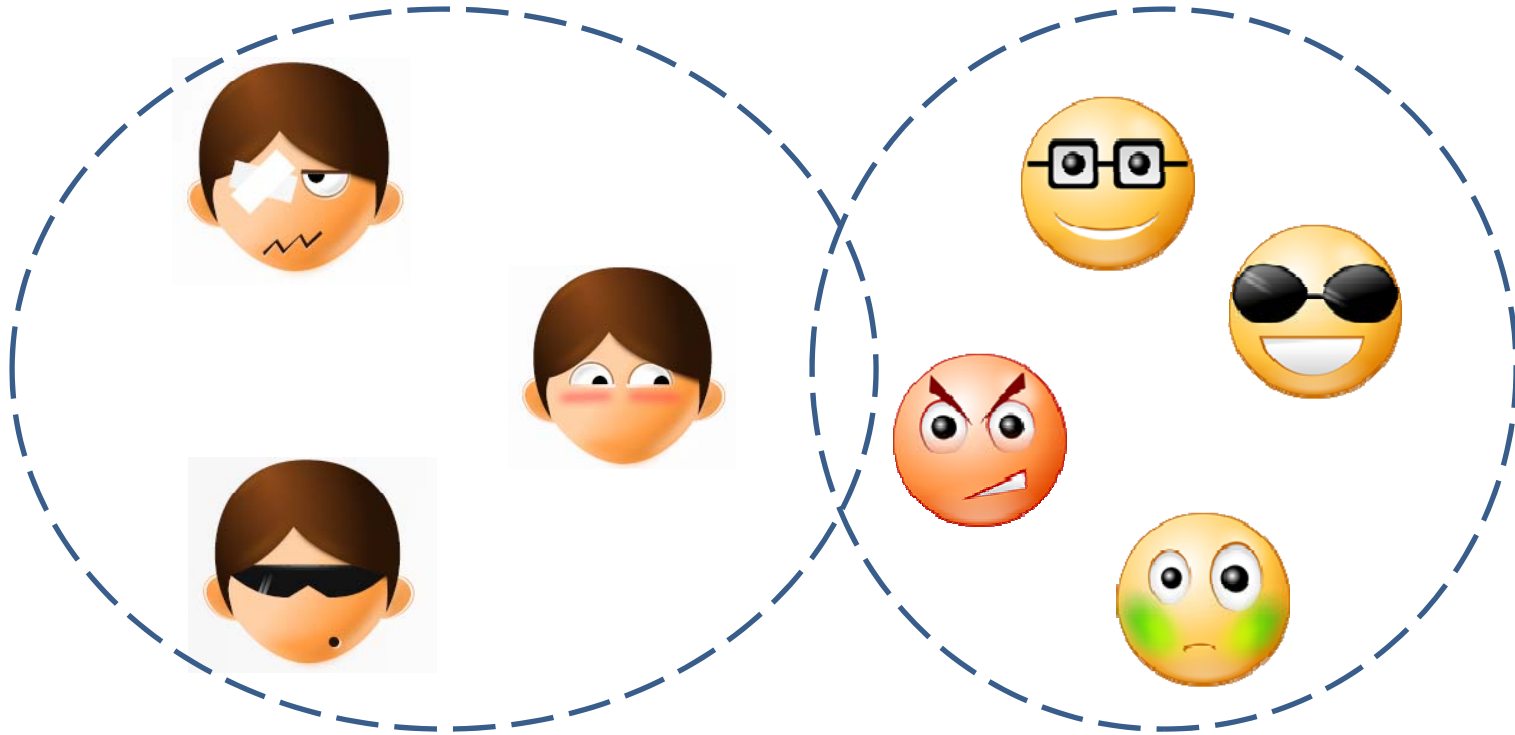
Digital World



Records /  
Mentions

# Deduplication Problem Statement

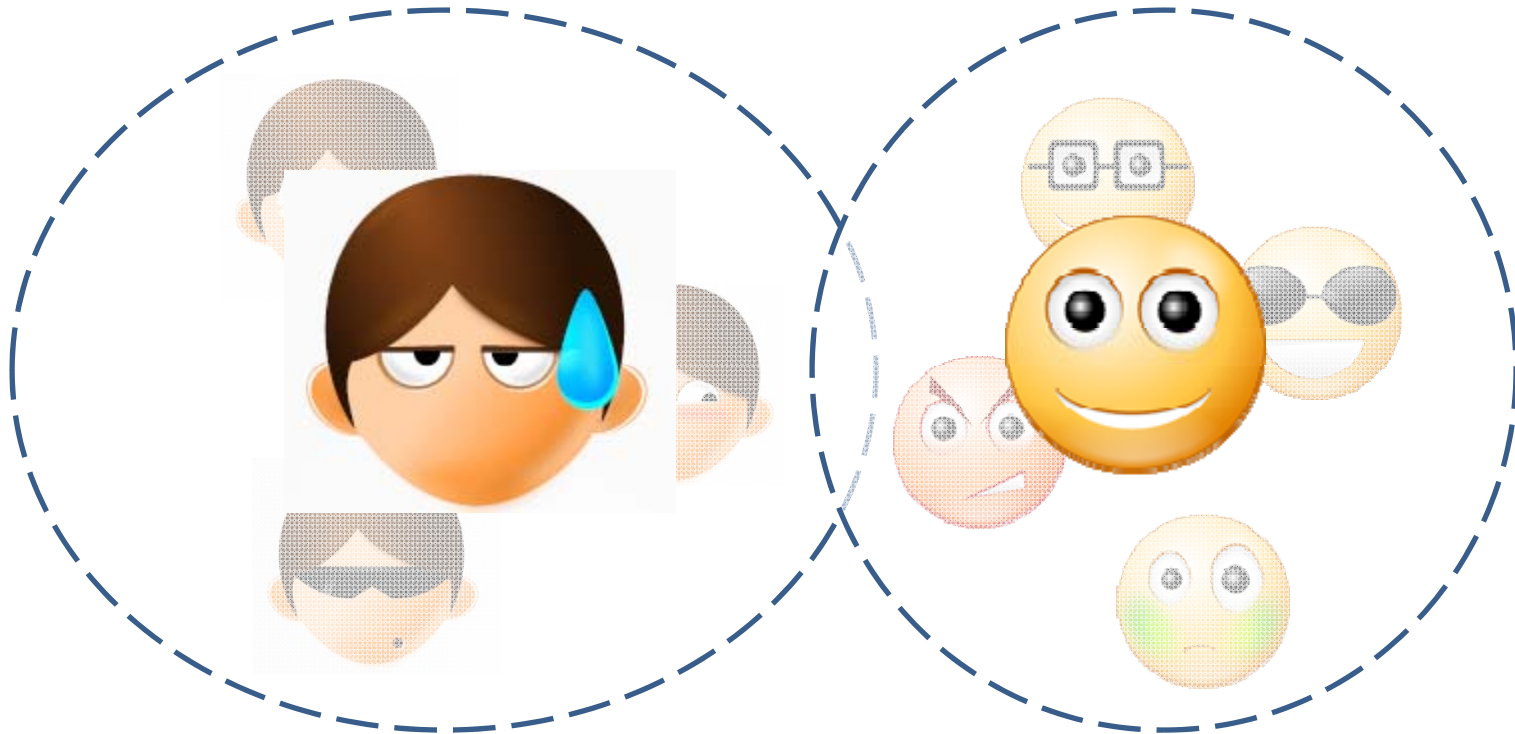
- Cluster the records/mentions that correspond to same entity





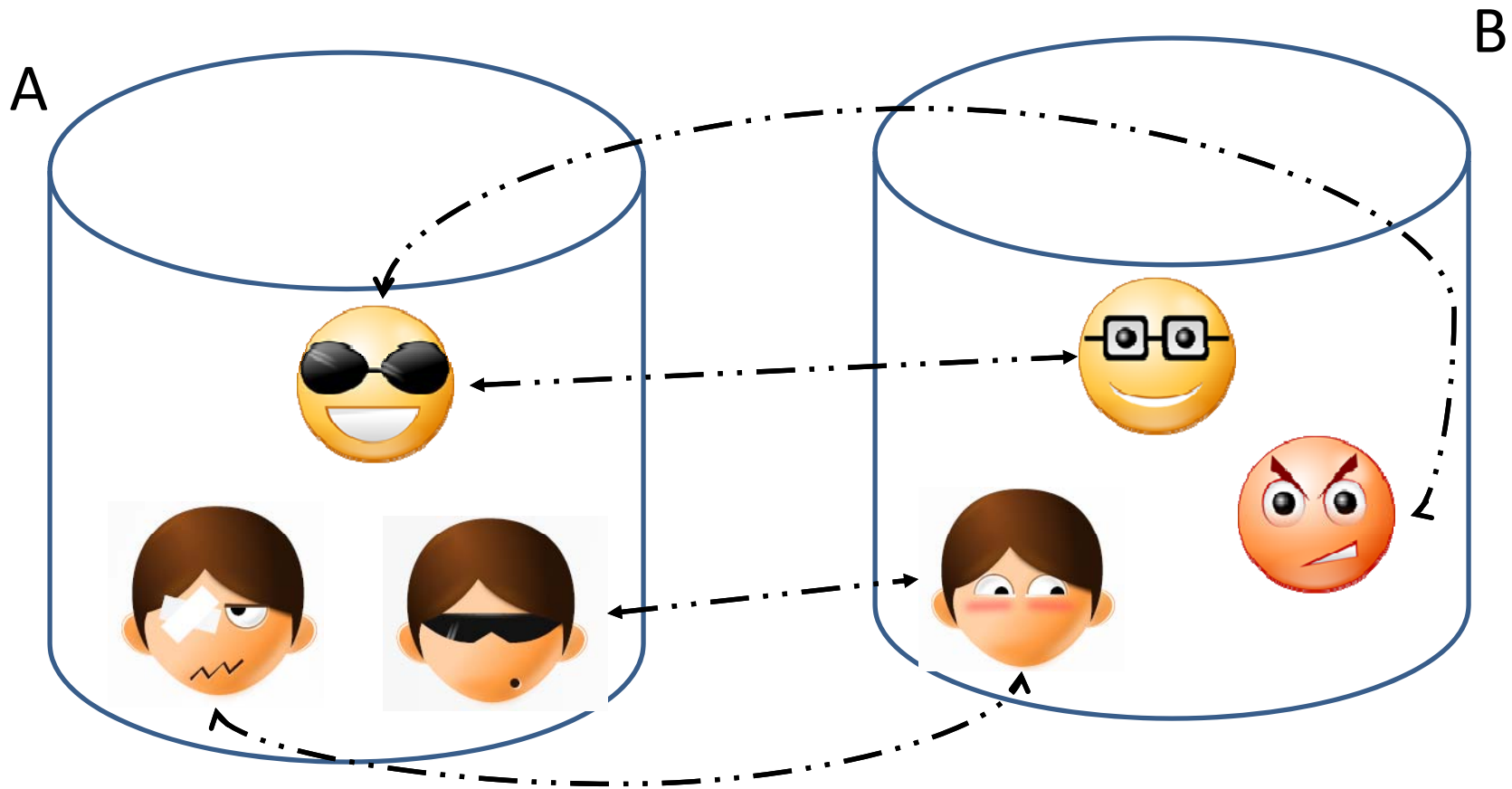
# Deduplication Problem Statement

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



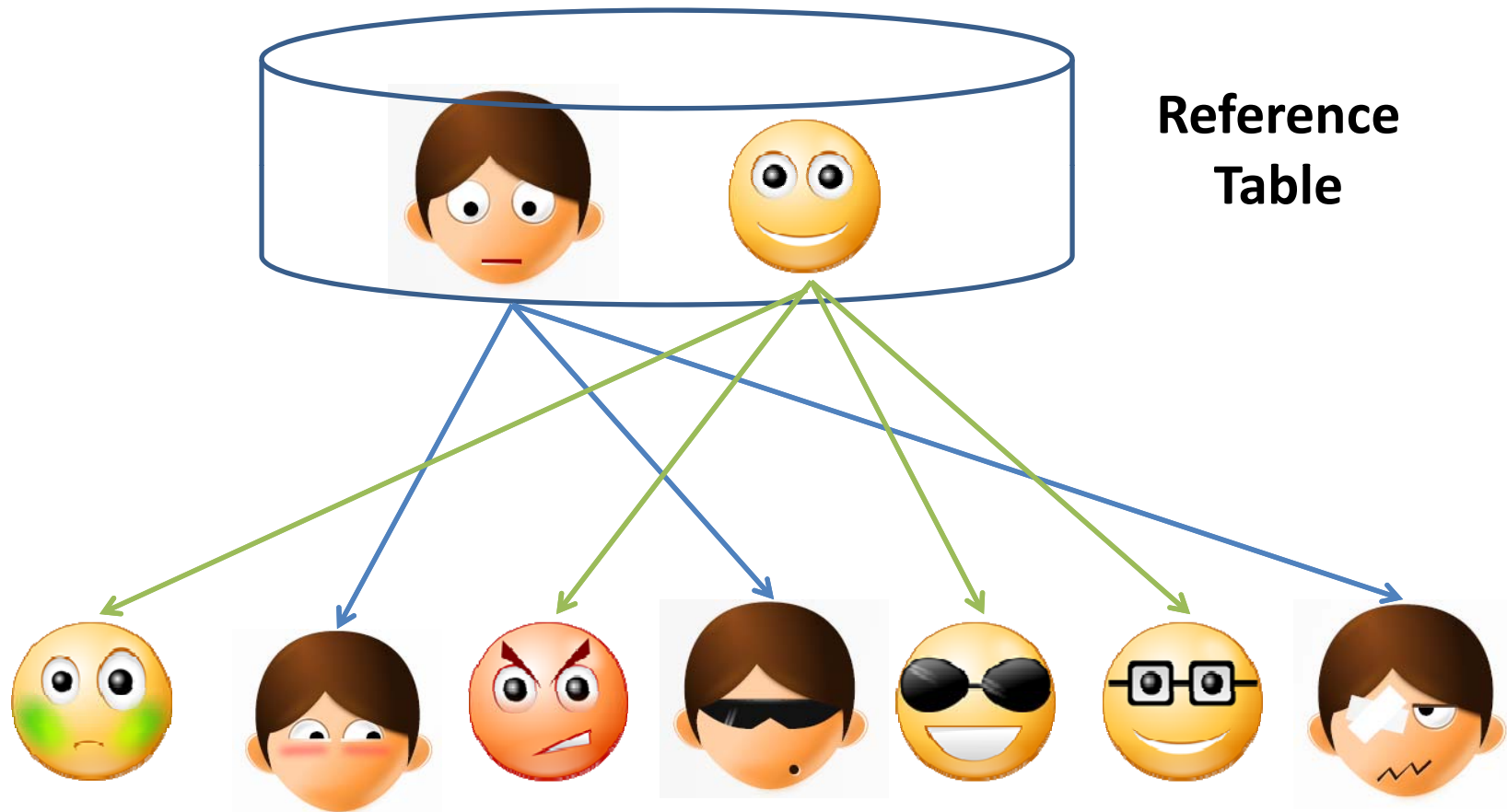
# Record Linkage Problem Statement

- Link records that match across databases



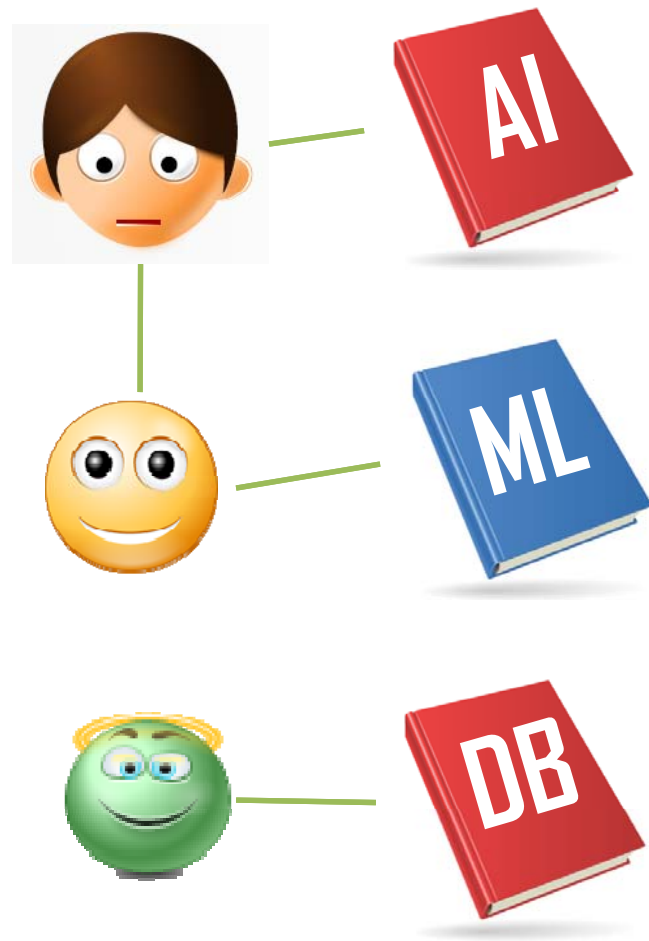
# Reference Matching Problem

- Match noisy records to clean records in a reference table

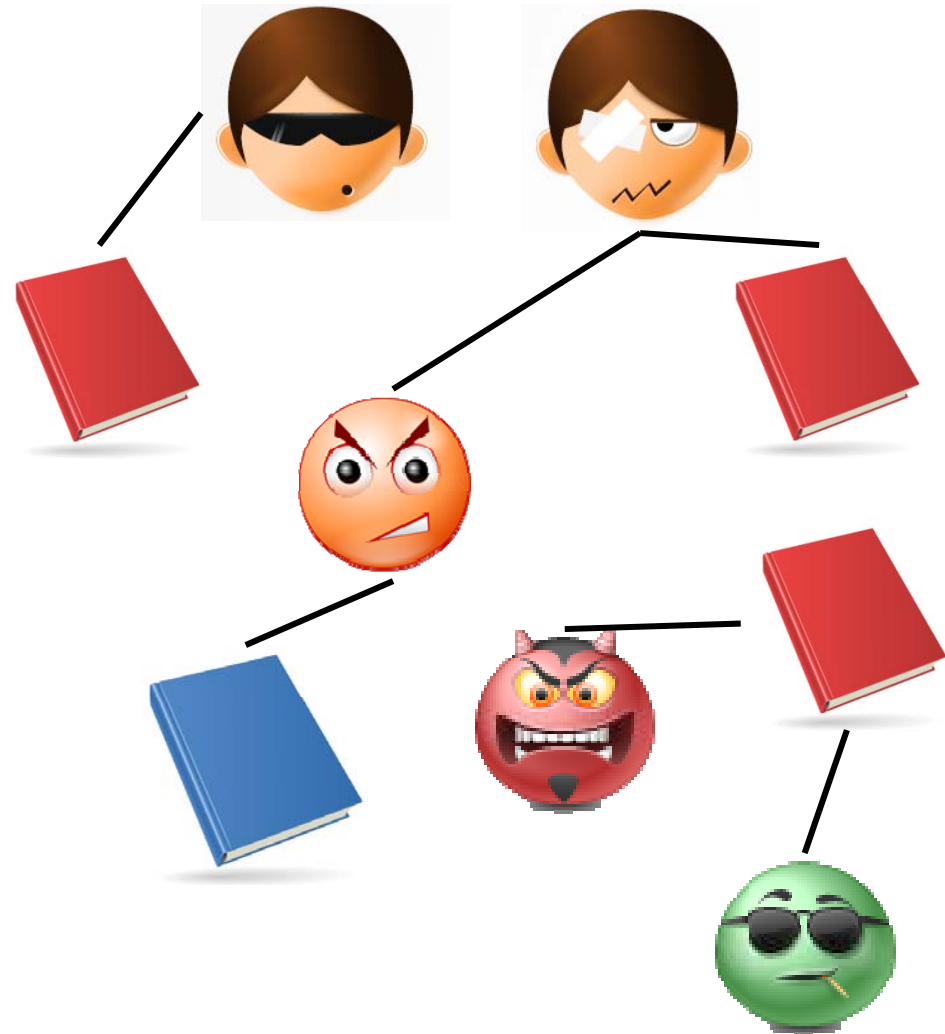


# Abstract Problem Statement

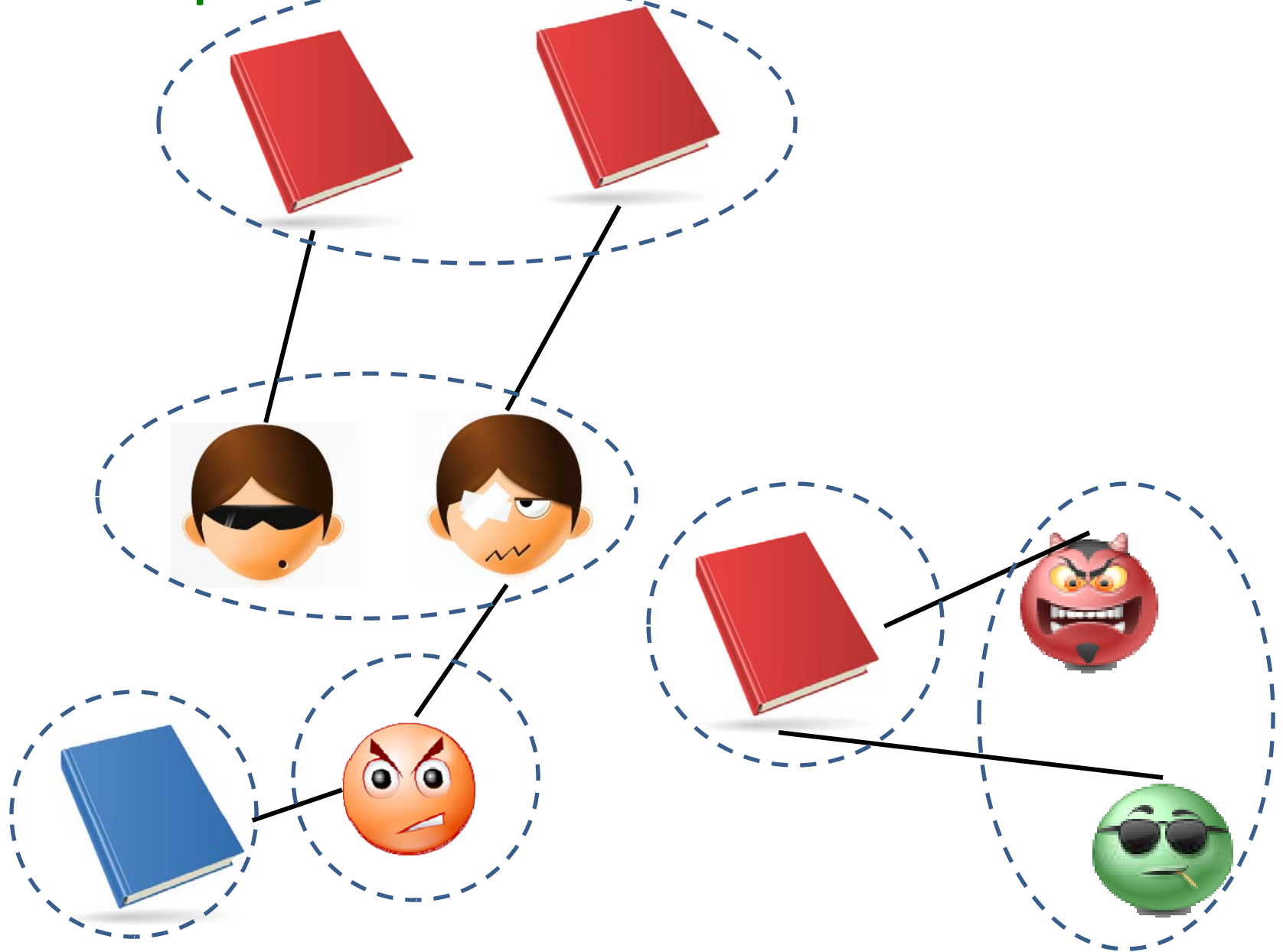
Real World



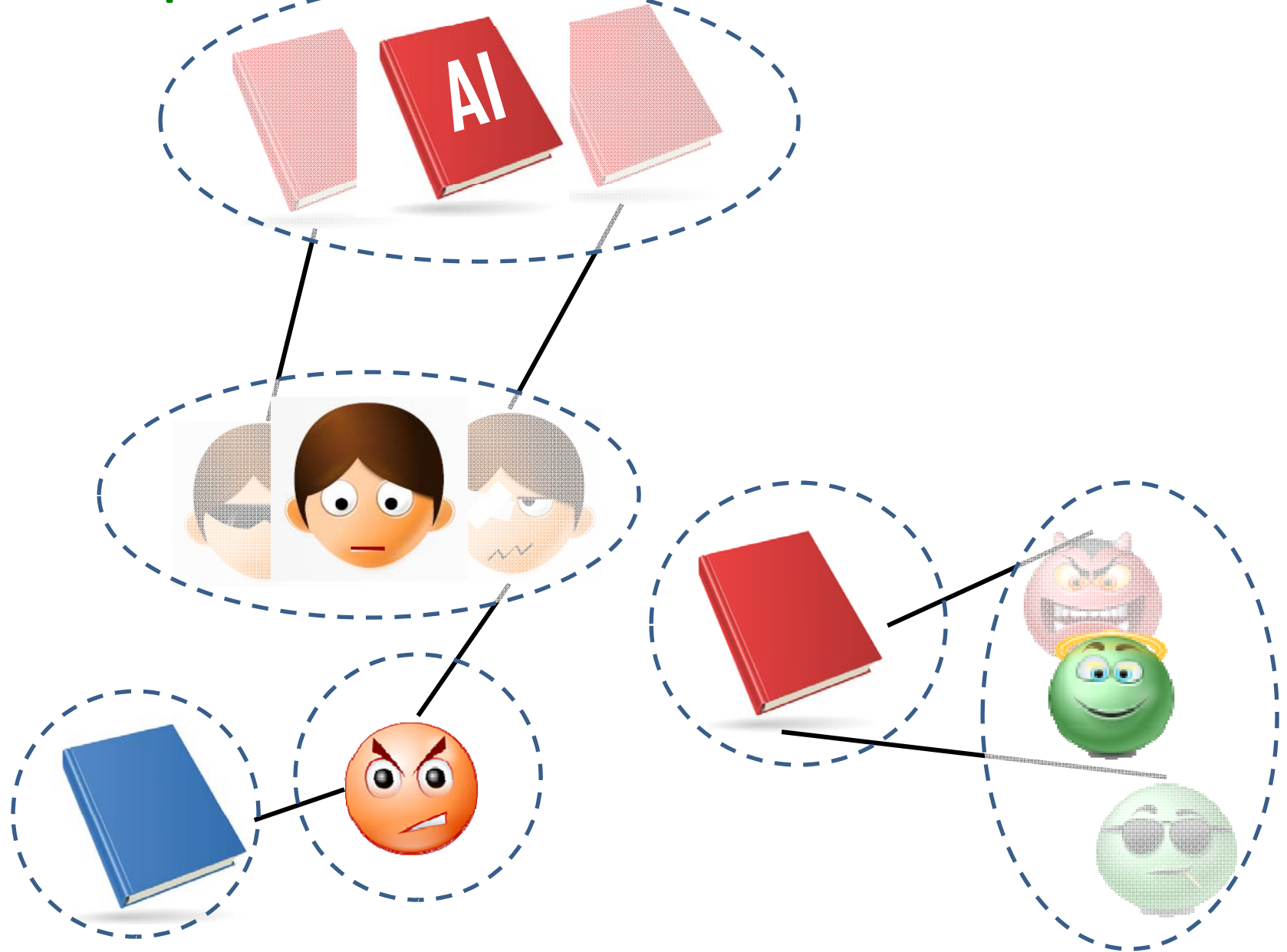
Digital World



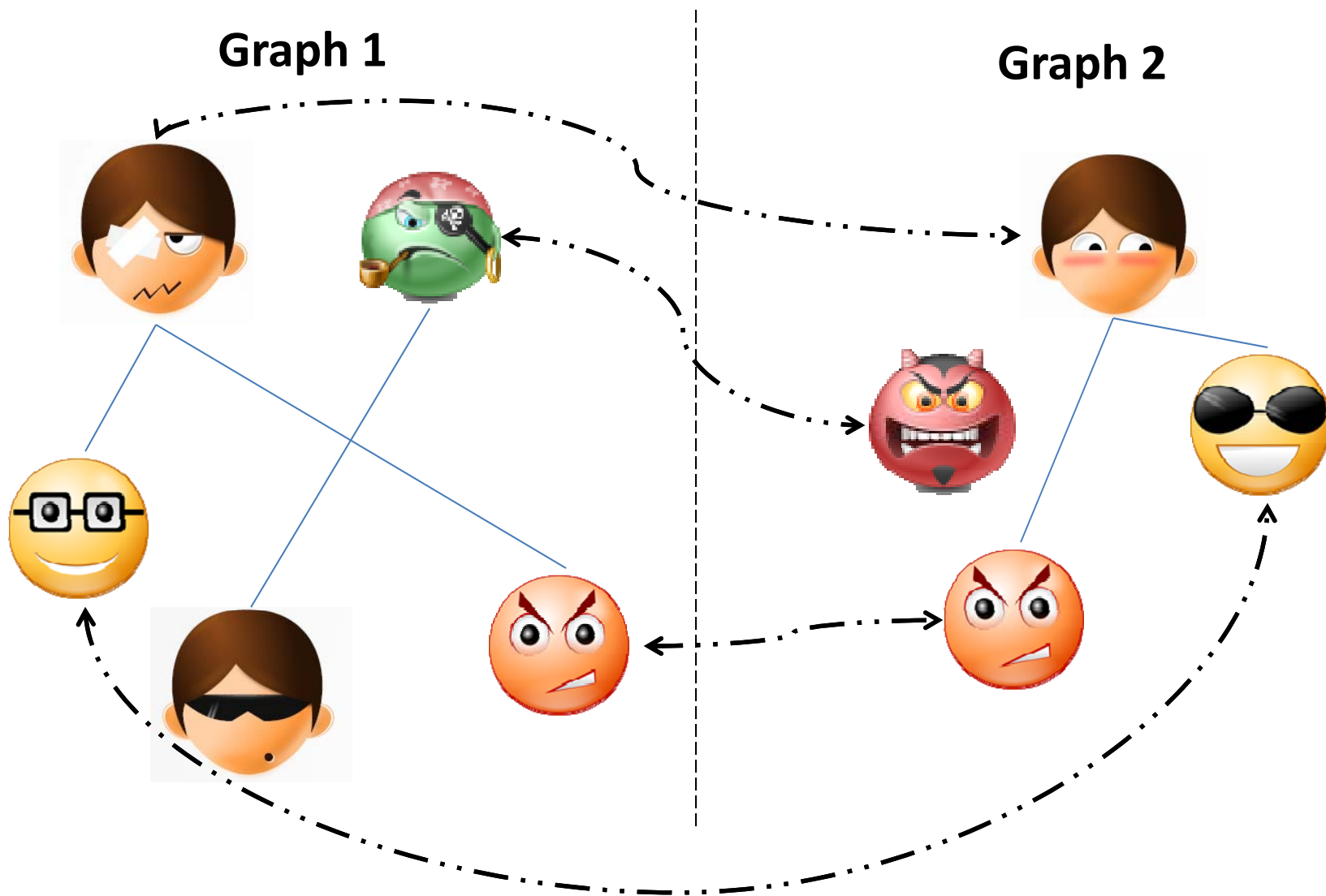
# Deduplication Problem Statement



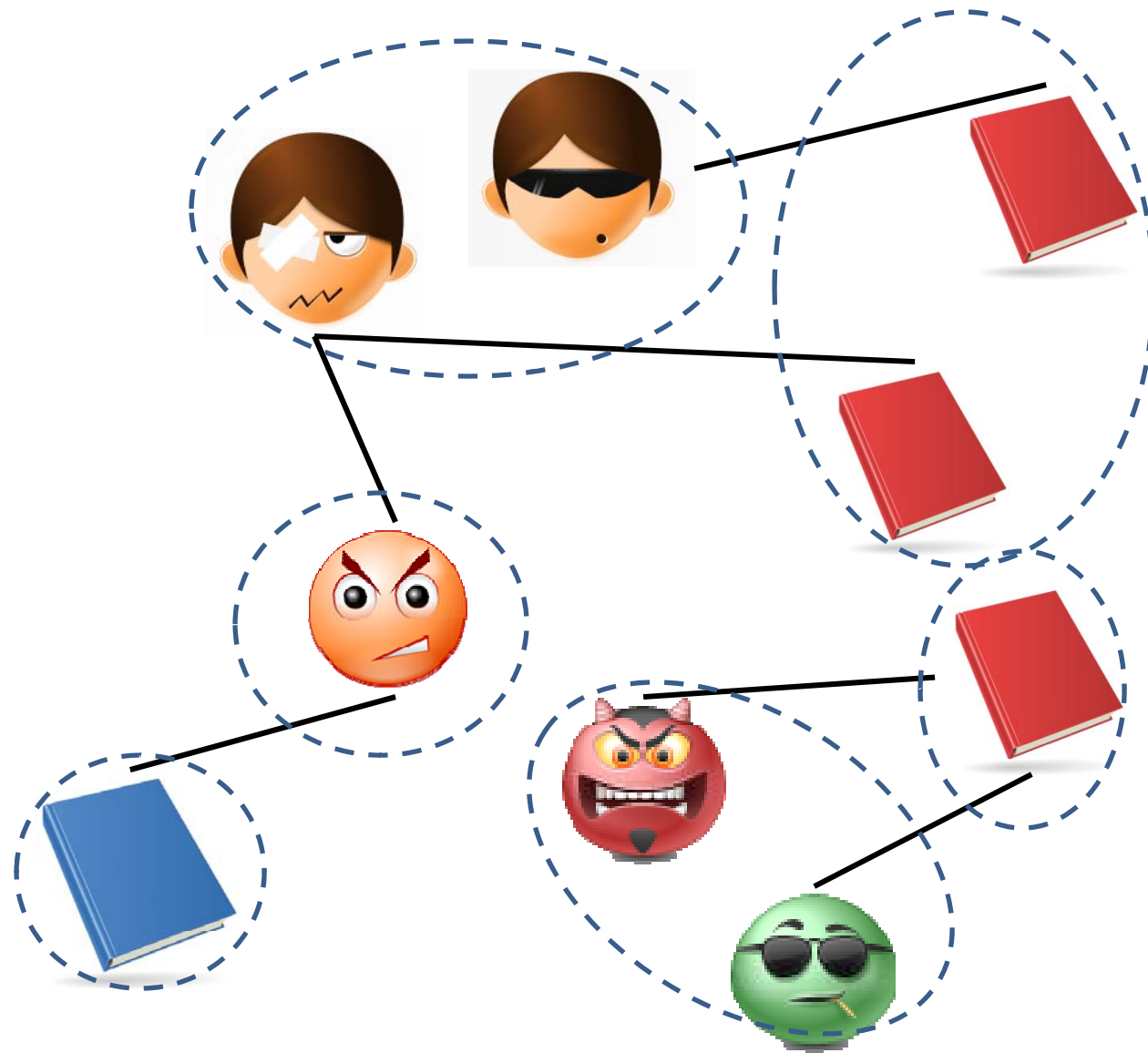
# Deduplication with Canonicalization



# Graph Alignment (& motif search)

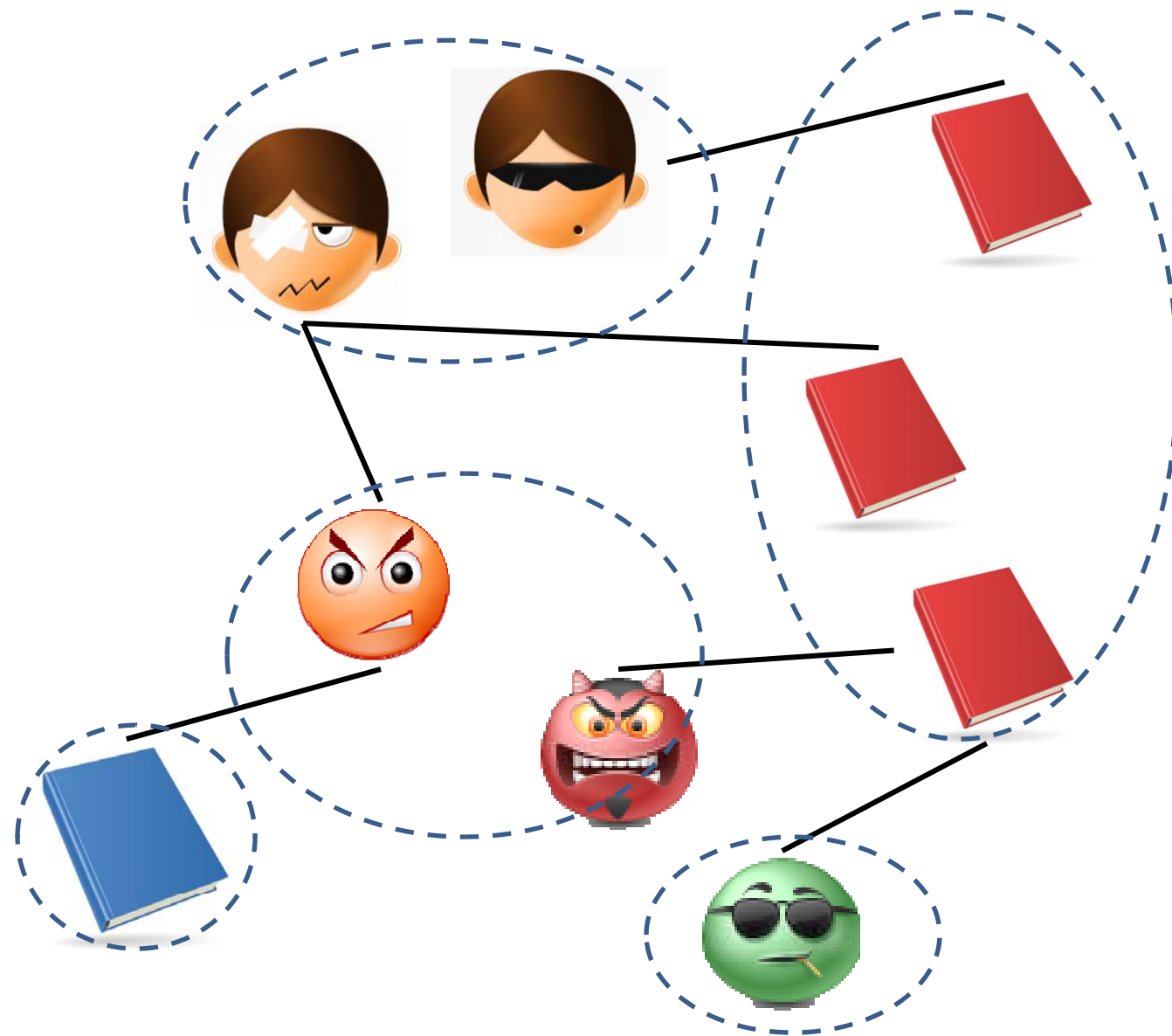


# Relationships are crucial





# Relationships are crucial

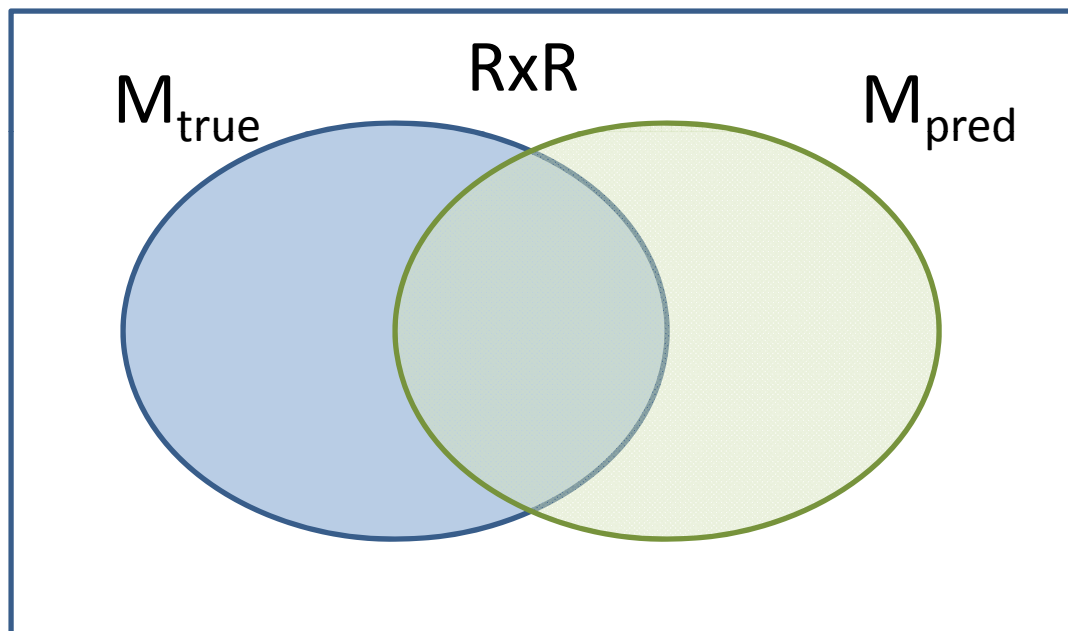


# Notation

- $R$ : 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

# Relationship between $M_{\text{true}}$ and $M_{\text{pred}}$

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

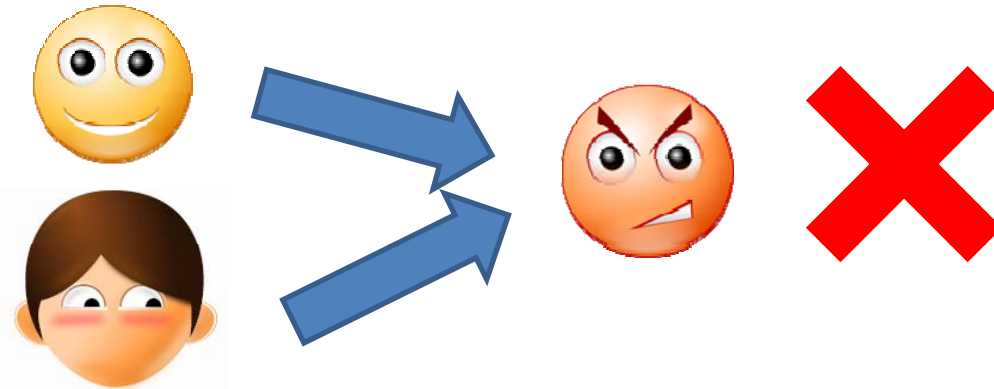


# Metrics

- Pairwise metrics
  - Precision/Recall, F1
  - # of predicted matching pairs
- Cluster level metrics
  - purity, completeness, complexity
  - Precision/Recall/F1: Cluster-level, closest cluster, MUC, B<sup>3</sup>, Rand Index
  - Generalized merge distance [Menestrina et al, PVLDB10]
- Little work that evaluations correct prediction of links

# 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*

$$\left( \text{👤}, \text{👤} \right) \in M_{\text{true}}$$

# 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

(, )  $\in M_{\text{true}}$

AND



(, )  $\in M_{\text{true}}$

(, )  $\in M_{\text{true}}$

# ER vs (Multi-relational) 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 2

**ALGORITHMIC FOUNDATIONS OF ER**



# Outline of Part 2

a) Data Preparation and Match Features

b) Pairwise ER

- Determining whether or not a pair of records match

c) Constraints in ER



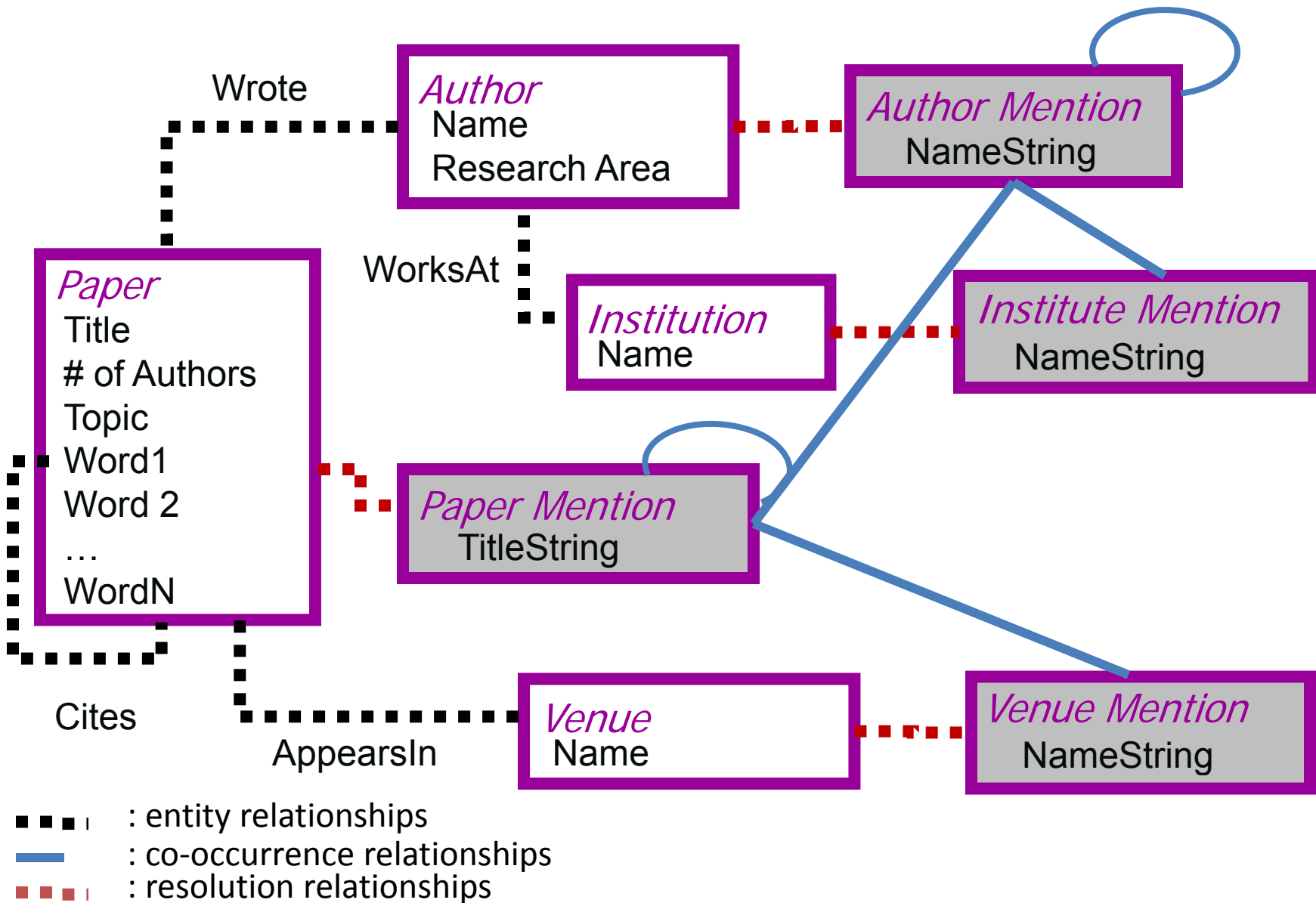
**5 minute break**

d) Algorithms

- Record linkage (Propagation through exclusivity negative constraint),
- Deduplication (Propagation through transitivity positive constraint),
- Collective (Propagation through General Constraints)

**MOTIVATING EXAMPLE:  
BIBLIOGRAPHIC DOMAIN**

# Entities & Relations in Bibliographic Domain



PART 2-a

**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 conditioning) each feature a boolean attribute
  - Set valued attributes
    - Set of phones vs phone
  - Record segmentation
- Data normalization
  - Often convert to 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 Features

- For two references  $x$  and  $y$ , compute a “comparison” vector of *similarity scores* of component attribute.
  - [ 1<sup>st</sup>-author-match-score,  
paper-match-score,  
venue-match-score,  
year-match-score, .... ]
- Similarity scores
  - Boolean (match or not-match)
  - Real values based on distance functions

# Summary of Matching Features

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

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

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.

# Relational Matching 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



PART 2-b

**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.
  - $0.5 * 1^{\text{st}}\text{-author-match-score} + 0.2 * \text{venue-match-score} + 0.3 * \text{paper-match-score}$ .
  - Hard to pick weights.
    - Match on last name match *more predictive* than login name.
    - Match on “Smith” *less predictive* than match on “Getoor” or “Machanavajjhala”.
  - Hard to tune a threshold.

# Pairwise Match Score

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

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.
  - $(1^{\text{st}}\text{-author-match-score} > 0.7 \text{ AND venue-match-score} > 0.8)$   
OR  $(\text{paper-match-score} > 0.9 \text{ AND venue-match-score} > 0.9)$
  - Manually formulating the right set of rules is hard.

# Basic ML Approach

- $r = (x, y)$  is record pair,  $\gamma$  is comparison vector,  $M$  matches,  $U$  non-matches
- Decision rule 
$$R = \frac{P(\gamma | r \in M)}{P(\gamma | r \in U)}$$

$$R > t \Rightarrow r \rightarrow \text{Match}$$

$$R \leq t \Rightarrow r \rightarrow \text{Non - Match}$$

# Fellegi & Sunter Model [FS, Science '69]

- $r = (x, y)$  is record pair,  $\gamma$  is comparison vector,  $M$  matches,  $U$  non-matches

- Decision rule 
$$R = \frac{P(\gamma | r \in M)}{P(\gamma | r \in U)}$$

$$R \geq t_l \Rightarrow r \rightarrow \text{Match}$$

$$t_l < R < t_u \Rightarrow r \rightarrow \text{Potential Match}$$

$$R \leq t_u \Rightarrow r \rightarrow \text{Non - Match}$$

- Naïve Bayes Assumption: 
$$P(\gamma | r \in M) = \prod_i P(\gamma_i | r \in M)$$

# 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)
    - [Gupta & Sarawagi, VLDB09]
- 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

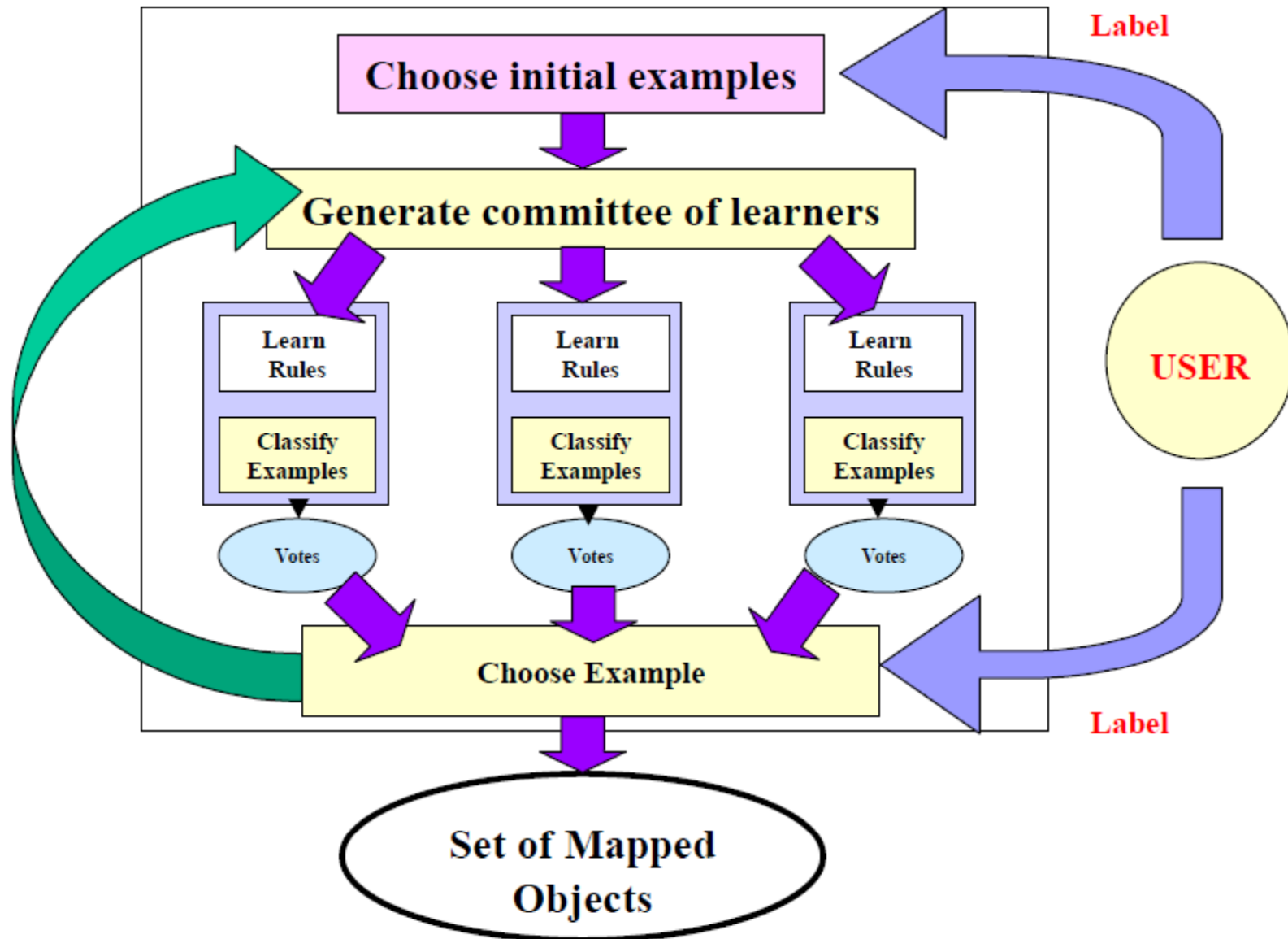
- Constructing a training set is hard – since most pairs of records are “easy non-matches”.
  - 100 records from 100 cities.
  - Only  $10^6$  pairs out of total  $10^8$  (1%) come from the same city
- Some pairs are hard to judge even by humans
  - Inherently ambiguous
    - E.g., Paris Hilton (person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

# Avoiding Training Set Generation

- Unsupervised / Semi-supervised Techniques
  - EM based techniques to learn parameters
    - [Winkler '06, 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]
  - Crowdsourcing
    - [Wang et al VLDB '12, Marcus et al VLDB '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].

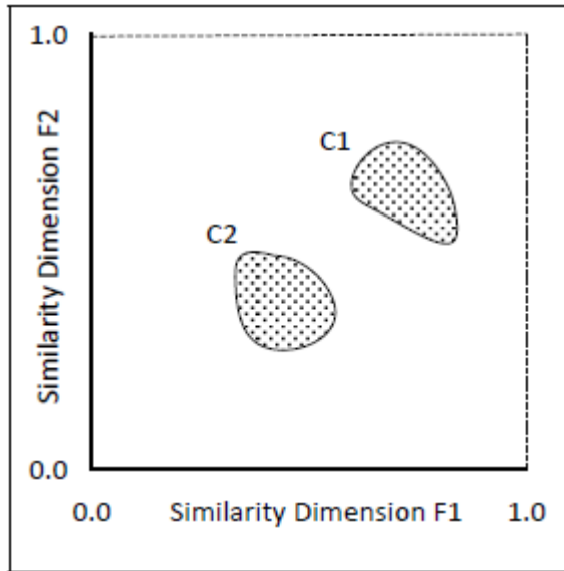
$$\text{minimize } \frac{f_n(h) + f_p(h)}{n}$$

- However, ER is very imbalanced:
  - Number of non-matches > 100 \* number of matches.
  - Classifying all pairs as “non-matches” has low 0-1 loss (< 1%).
- Hence, need active learning techniques that minimize precision/recall.

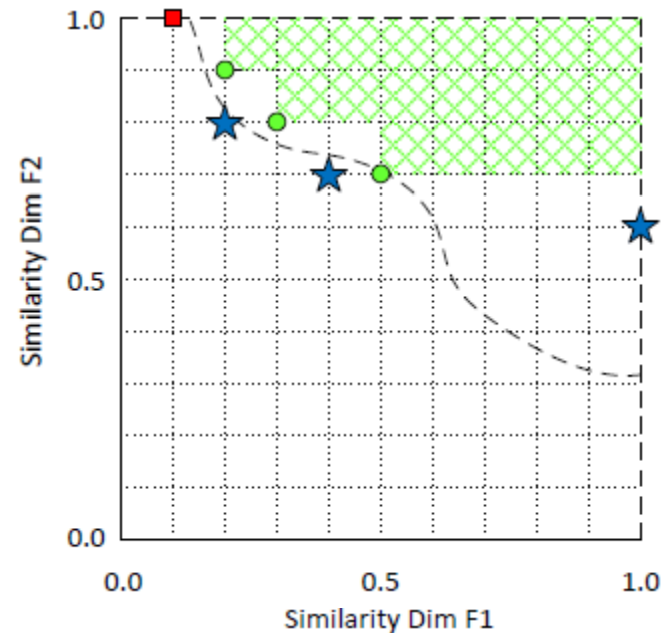
$$\begin{array}{ll} \text{maximize} & \text{recall}(h) \\ \text{subject to} & \text{precision}(h) \geq \tau \end{array}$$

# 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  $\rightarrow$  Weighted 0-1 Loss Problem (using a Lagrange Multiplier  $\lambda$ ).
2. Given a fixed value for  $\lambda$ , weighted 0-1 Loss can be optimized by (one call to) a blackbox active learning classifier.
3. Right value of  $\lambda$  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

# Crowdsourcing

- Growing interest in integrating human computation in declarative workflow engines.
  - ER is an important problem (e.g., for evaluating fuzzy joins)
  - [Wang et al VLDB '12, Marcus et al VLDB '12, ...]
- Opportunity: utilize crowdsourcing for creating training sets, or for active learning.
- Key open issue: Handling errors in human judgments
  - In an experiment on Amazon Mechanical Turk:
    - Pairwise matching judgment, each given to 5 different people
  - Majority of workers agreed on truth on only 90% of pairwise judgments.

# 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 high fidelity training sets is a hard problem
- Active Learning & Crowdsourcing for ER are active areas of research.

PART 2-c

**CONSTRAINTS**

# Constraints

- 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
  - **Functional Dependency:** If M1 and M2 match, then M3 and M4 must match
- Transitivity is key to deduplication
- Exclusivity is key to record linkage
- Functional dependencies for data cleaning, e.g.,  
[Ananthakrishna et al., VLDB02][Fan, PODS08][Bohannon et al, ICDE07]



# Positive & Negative Evidence

- Positive
  - **Transitivity:** If M1 and M2 match, M2 and M3 match, then M1 and M3 match
  - **Functional Dependency:** If M1 and M2 match, then M3 and M4 must match
- Negative
  - **Exclusivity:** If M1 matches with M2, then M3 cannot match with M2

# Positive & Negative Evidence

- Positive
  - **Transitivity:** If M1 and M2 match, M2 and M3 match, then M1 and M3 match
  - **Exclusivity:** If M1 doesn't match with M2, then M3 can match with M2
  - **Functional Dependency:** If M1 and M2 match, then M3 and M4 must match
- Negative
  - **Transitivity:** If M1 and M2 match, M2 and M3 do not match, then M1 and M3 do not match
  - **Exclusivity:** If M1 matches with M2, then M3 cannot match with M2
  - **Functional Dependency:** If M1 and M2 do not match, then M3 and M4 cannot match

# Constraint Types

	Hard Constraint	Soft Constraint
Positive Evidence	<p>If M1, M2 match then M3, M4 must match</p> <p><i>If two papers match, their venues match</i></p>	<p>If M1, M2 match then M3, M4 more likely to match</p> <p><i>If two venues match, then their papers are more likely to match</i></p>
Negative Evidence	<p>Mention M1 and M2 must refer to distinct entities (Uniqueness)</p> <p><i>Coauthors are distinct</i></p> <p>If M1, M2 don't match then M3, M4 cannot match</p> <p><i>If two venues don't match, then their papers don't match</i></p>	<p>If M1, M2 don't match then M3, M4 less likely to match</p> <p><i>If institutions don't match, then authors less likely to match</i></p>

# Constraint Types

	Hard Constraint	Soft Constraint
Positive Evidence	<p>If M1, M2 match then M3, M4 must match</p> <p><i>If two papers match, their venues match</i></p>	<p>If M1, M2 match then M3, M4 more likely to match</p> <p><i>If two venues match, then their papers are more likely to match</i></p>
Negative Evidence	<p>Mention M1 and M2 must refer to distinct entities (<b>Uniqueness</b>)</p> <p><i>Coauthors are distinct</i></p> <p>If M1, M2 don't match then M3, M4 cannot match</p> <p><i>If two venues don't match, then their papers don't match</i></p>	<p>If M1, M2 don't match then M3, M4 less likely to match</p> <p><i>If institutions don't match, then authors less likely to match</i></p>

# Constraint Types

	Hard Constraint	Soft Constraint
Positive Evidence	<p>If M1, M2 match then M3, M4 must match</p> <p><i>If two papers match, their venues match</i></p>	<p>If M1, M2 match then M3, M4 more likely to match</p> <p><i>If two venues match, then their papers are more likely to match</i></p>
Negative Evidence	<p>Mention M1 and M2 must refer to distinct entities (<b>Uniqueness</b>)</p> <p><i>Coauthors are distinct</i></p> <p>If M1, M2 don't match then M3, M4 cannot match</p> <p><i>If two venues don't match, then their papers don't match</i></p>	<p>If M1, M2 don't match then M3, M4 less likely to match</p> <p><i>If institutions don't match, then authors less likely to match</i></p>

# Constraint Types

	Hard Constraint	Soft Constraint
Positive Evidence	<p>If M1, M2 match then M3, M4 must match</p> <p><i>If two papers match, their venues match</i></p>	<p>If M1, M2 match then M3, M4 more likely to match</p> <p><i>If two venues match, then their papers are more likely to match</i></p>
Negative Evidence	<p>Mention M1 and M2 must refer to distinct entities (<b>Uniqueness</b>)</p> <p><i>Coauthors are distinct</i></p> <p>If M1, M2 don't match then M3, M4 cannot match</p> <p><i>If two venues don't match, then their papers don't match</i></p>	<p>If M1, M2 don't match then M3, M4 less likely to match</p> <p><i>If institutions don't match, then authors less likely to match</i></p>

# Constraint Types

	Hard Constraint	Soft Constraint
Positive Evidence	<p>If M1, M2 match then M3, M4 match</p> <p><i>If two papers match, their coauthors match</i></p>	<p><i>to match</i></p>
Negative Evidence	<p>M1 and M2 must refer to distinct entities (<b>Uniqueness</b>)</p> <p><i>Coauthors are distinct</i></p> <p>If M1, M2 don't match then M3, M4 cannot match</p> <p><i>If two venues don't match, their papers don't match</i></p>	<p>If M1, M2 don't match then M3, M4 less likely to match</p> <p><i>to match, to match</i></p>

Note that some of the constraints may be **relational** and require joins

May be **directional** or **bidirectional**

Constraints can be **recursive**, e.g., if two authors have matching co-authors, then they match

# Additional Constraints

- Aggregate Constraints [Chaudhuri et al. SIGMOD07]
  - count constraints
    - Entity A can link to at most N Bs
      - Authors have at most 5 papers at any conference
  - Other aggregates like sum, average more complex
- Again, these can be either hard or soft constraints, provide positive or negative evidence



# Match Dependencies

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 Extent

- **Global:** If 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

# 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]

# Algorithms for Handling Constraints

- Record linkage - propagation through exclusivity
  - Weighted k-partite matching
- Deduplication - propagation through transitivity
  - Correlation clustering
- Collective - propagation through general constraints
  - Similarity propagation
    - Dependency graphs, Collective Relational Clustering
  - Probabilistic approaches
    - LDA, CRFs, Markov Logic Networks, Probabilistic Relational Models,
  - Hybrid approaches
    - Dedupalog

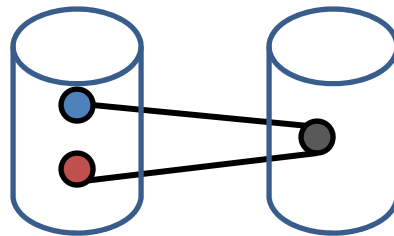
PART 2-d

**ALGORITHMS**

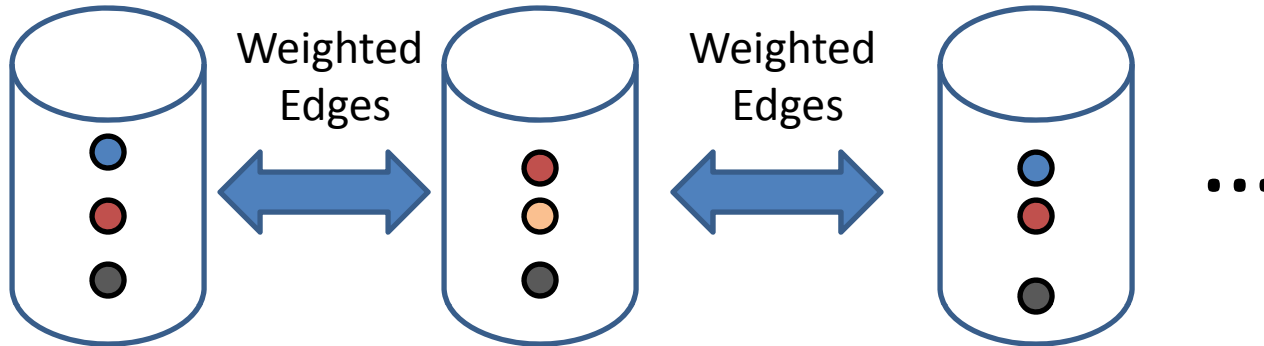
# RECORD LINKAGE

# 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



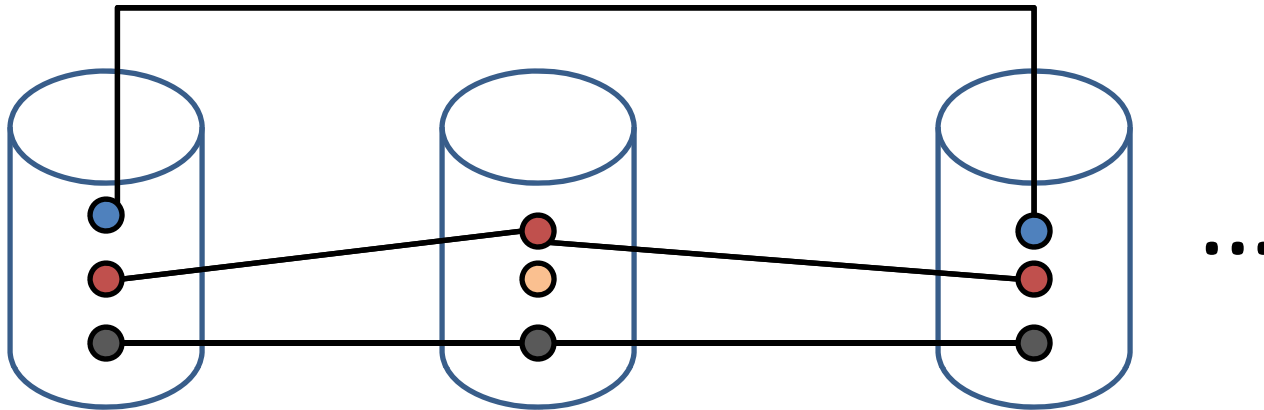
# Weighted K-Partite Matching



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



# Weighted K-Partite Matching



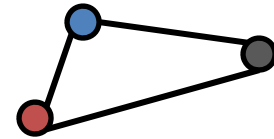
- Find a matching (each record matches at most one other record from other database) that maximize the sum of weights.
- General problem is NP-hard (3D matching)
- Successive bipartite matching is typically used. [Gupta & Sarawagi, VLDB '09]

**DEDUPLICATION**

# Deduplication => Transitivity

- Often pairwise ER algorithm output “inconsistent” results

–  $(x, y) \in M_{\text{pred}}, (y, z) \in M_{\text{pred}}, \text{ but } (x, z) \notin M_{\text{pred}}$

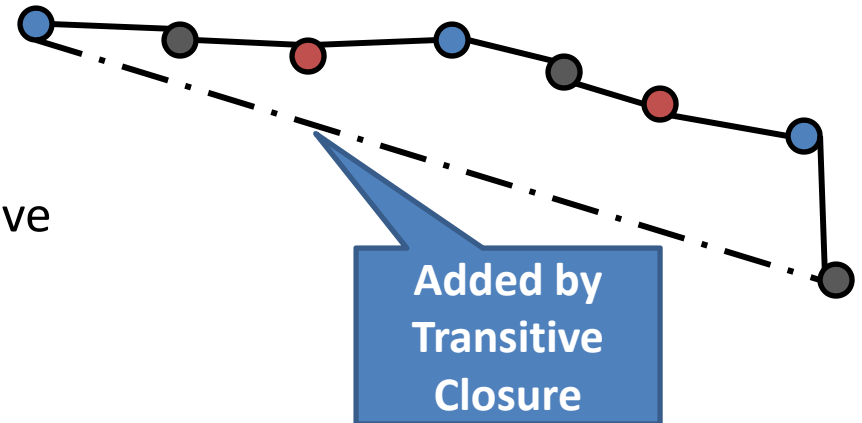


- Idea: Correct this by adding additional matches using transitive closure

- 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 apriori
  - 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

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

$$\text{minimize } \sum r_{xy} w_{xy}^+ + (1 - r_{xy}) w_{xy}^-$$

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

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

Transitive  
closure

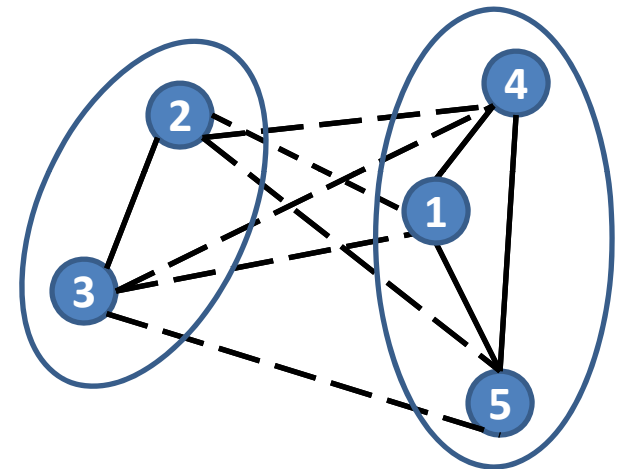
# Correlation Clustering

$$\begin{aligned} & \text{minimize } \sum r_{xy} w_{xy}^+ + (1 - r_{xy}) w_{xy}^- \\ & \text{s.t. } \forall x, y, z \in R, \\ & \quad r_{xy} + r_{xz} + r_{yz} \neq 2 \end{aligned}$$

- 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  $w_{xy}^- = (1 - p_{xy})$

– Logarithmic:  $w_{xy}^+ = \log(p_{xy})$  and  $w_{xy}^- = \log(1 - p_{xy})$

# 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  $\pi$

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:

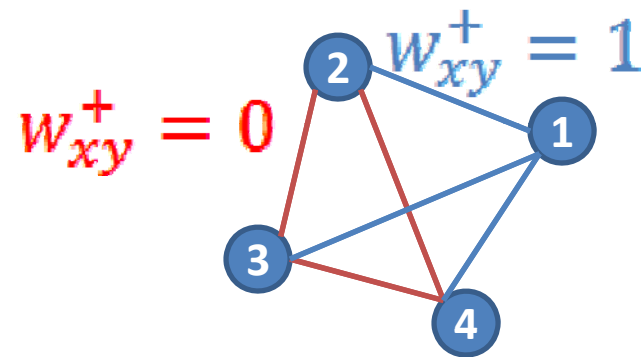
- Run the algorithm for many random permutations , and pick the clustering with best objective value (better than average run)

# Greedy with approximation guarantees

PIVOT Algorithm

[Ailon et al 2008 JACM]

- Pick a random (*pivot*) record  $p$ .
- New cluster =  $\{x \mid w_{px}^+ > 0\}$
- $\pi = \{1,2,3,4\}$   $C = \{\{1,2,3,4\}\}$
- $\pi = \{2,4,1,3\}$   $C = \{\{1,2\}, \{4\}, \{3\}\}$
- $\pi = \{3,2,4,1\}$   $C = \{\{1,3\}, \{2\}, \{4\}\}$



When weights are 0/1,

$E(\text{cost}(\text{greedy})) < 3 \text{ OPT}$

For  $w_{xy}^+ + w_{xy}^- = 1$ ,

$E(\text{cost}(\text{greedy})) < 5 \text{ OPT}$

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

PART 2-d

# **CANONICALIZATION**

# Canonicalization

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

- Starbucks,  
3457 Hillsborough Road  
Durham, NC  
Ph: *null*
- Starbucks,  
Hillsborough Rd, Durham  
Ph: (919) 333-4444

**Starbucks**

**3457 Hillsborough Road, Durham, NC**

**Ph: (919) 333-4444**

Critically important in Web portals where users must be shown a consolidated view

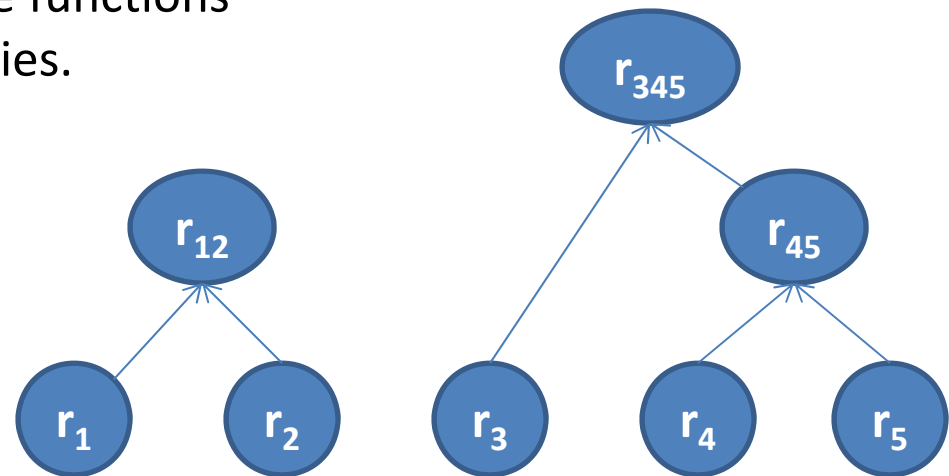
- Each mention only contains a subset of the attributes
- Mentions contain variations (of names, addresses)
- Some of the mentions have incorrect values

# 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.



# **COLLECTIVE ENTITY RESOLUTION**

# Collective Approaches

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



# **SIMILARITY PROPAGATION**

# Similarity Propagation Approaches

- Similarity propagation algorithms define a graph which encodes the similarity between entity mentions and matching decisions, and compute matching decisions by propagating similarity values.
  - Details of constructed graph 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

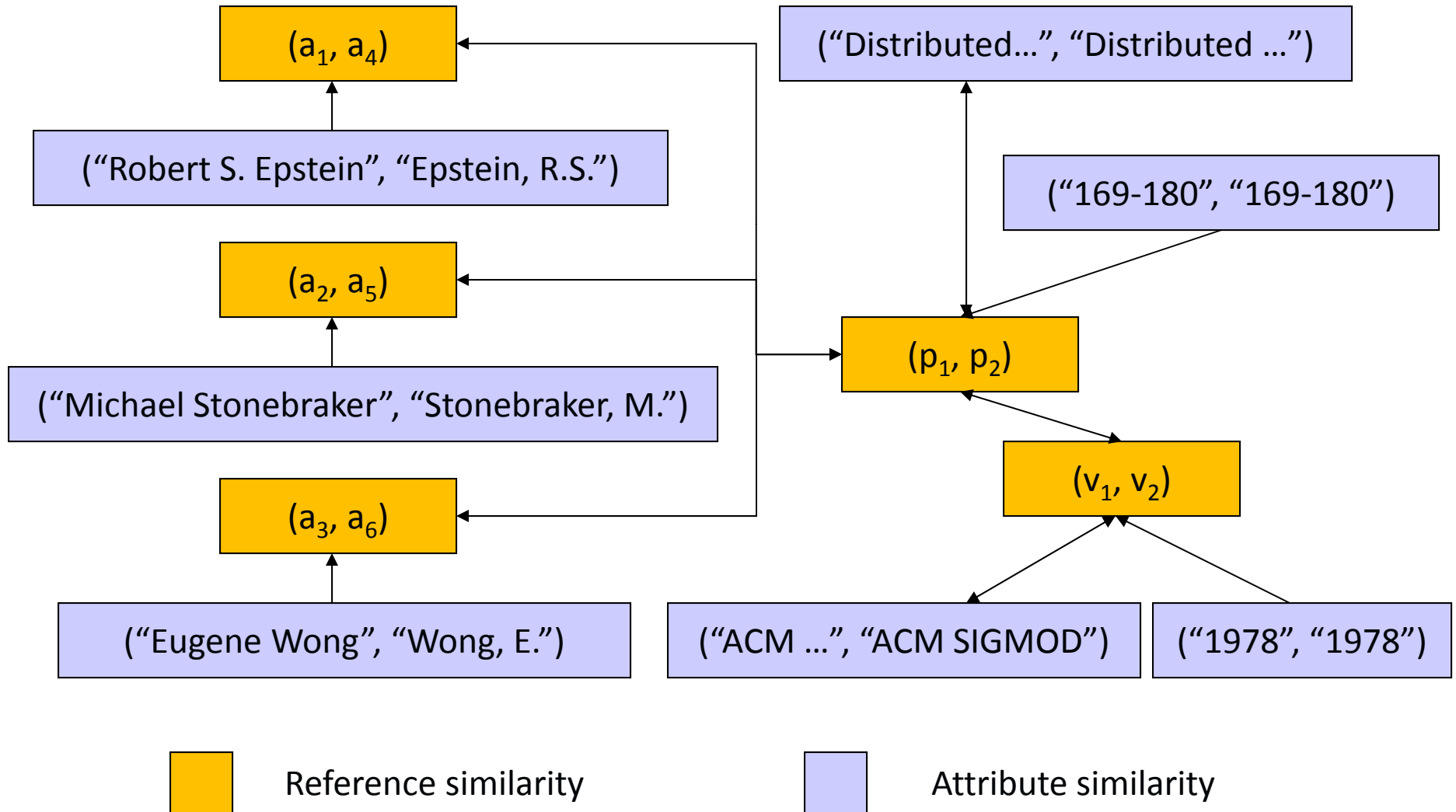
# Dependency Graph

[Dong et al., SIGMOD05 ]

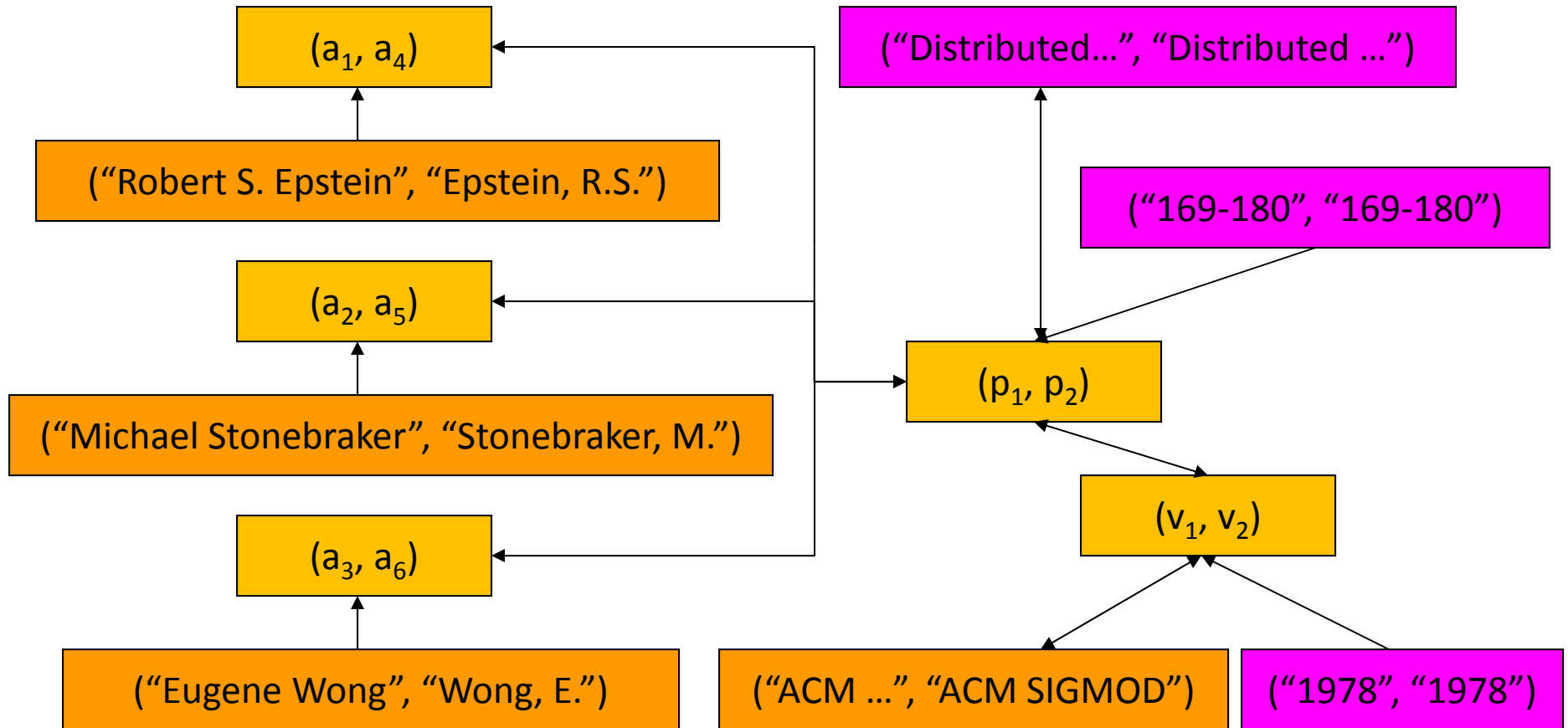
- Construct a graph where nodes represent similarity comparisons between attribute values (real-valued) and match decisions based on matching decisions of associated nodes (boolean-valued)
- As mentions are resolved, enriched to contain associated nodes of all matched mentions
- Similarity propagated until fixed point is reached
- Negative constraints (not-match nodes) are checked after similarity propagation is performed, and inconsistencies are fixed

# Exploit the Dependency Graph

Slides from [Dong et al, SIGMOD05]



# Exploit the Dependency Graph



 Reconciled

 Similar

# Collective Relational Clustering

[Bhattacharya & Getoor, TKDD07]

- Construct a graph where leaf nodes are individual mentions
- Perform hierarchical agglomerative clustering to merge clusters of mentions
- Similarity computed based on a combination of attribute and relational similarity
- When clusters are merged, update the similarities of any related clusters (clusters corresponding to mentions which co-occur with merged mentions)

# 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

where for example

$$sim_A(c_i, c_j) = \sum_{a \in Attributes} sim(c_i^*, c_j^*) \quad \text{for cluster representative } c^*$$

and

$$sim_R(c_i, c_j) = sim_{jaccard}(N(c_i), N(c_j))$$

where  $N(c)$  are the relational neighbors of  $c$

# 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. If no negative constraints violated
  8. Merge to create new cluster
  9. Construct canonical cluster representative
  10. Update similarity for 'related' clusters
- $O(n k \log n)$  algorithm w/ efficient implementation



# Similarity-propagation Approaches

	Method	Notes	Constraints	Evaluation
ReIDC [Kalashnikov et al, TODS06]	Reference disambiguation using using Relationship-based data cleaning (ReIDC)	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	Precision/Recall, F1 on three bibliographic datasets: CiteSeer, ArXiv, and BioBase, and synthetic data

**PROBABILISTIC MODELS:  
GENERATIVE APPROACHES**

# Generative Probabilistic Approaches

- Probabilistic semantics based on Directed Models
  - Model dependencies between match decisions in a generative manner
  - Disadvantage: acyclicity requirement
- Variety of approaches
  - Based on Latent Dirichlet Allocation, Bayesian Networks
- Examples
  - Latent Dirichlet Allocation [Bhattacharya & Getoor, SDM07]
  - Probabilistic Relational Models [Pasula et al, NIPS02]

# LDA for Entity Resolution: Discovering Groups from Co-Occurrence Relations



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

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

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

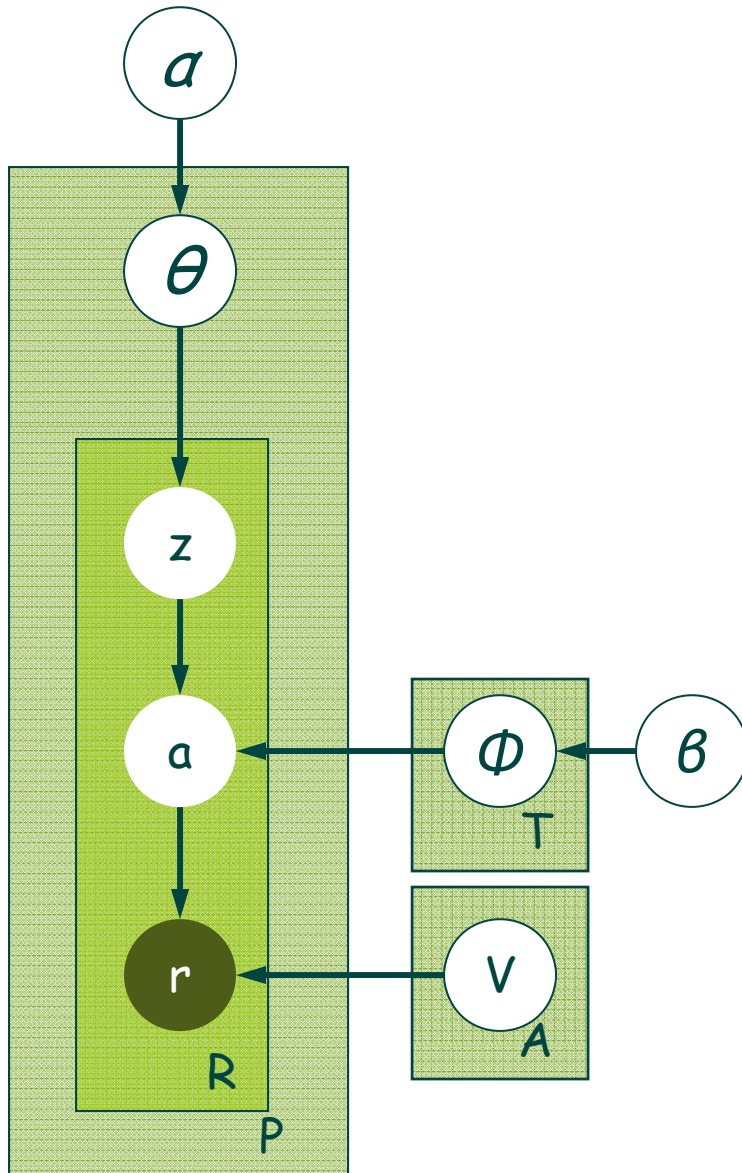


P4: Alfred V. Aho, Stephen C. Johnson,  
Jefferey D. Ullman

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

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

# LDA-ER Model



- Entity label  $a$  and group label  $z$  for each reference  $r$
- $\theta$ : 'mixture' of groups for each co-occurrence
- $\phi_z$ : multinomial for choosing entity  $a$  for each group  $z$
- $v_a$ : multinomial for choosing reference  $r$  from entity  $a$
- Dirichlet priors with  $\alpha$  and  $\beta$

Inference using blocked Gibbs sampling for efficiency (and improved accuracy)

# 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 Unsupervised approach	Precision/Recall /F1 on CiteSeer and HEP data

**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]



# 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  $\Rightarrow$  Stronger constraint

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

# 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

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

The diagram shows the probability formula  $P(X) = \frac{1}{Z} \exp\left(\sum_{i \in F} w_i n_i(x)\right)$  with several annotations:

- A red box labeled "Normalization Constant" points to the  $Z$  in the denominator.
- A purple box labeled "Iterate over all first-order MLN formulas" points to the  $i \in F$  index in the summation.
- A blue box labeled "# true groundings of *ith* clause" points to the  $n_i(x)$  term in the summation.

[Richardson & Domingos, 06]

# ER Problem Formulation in MLNs

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

# 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:

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

# Positive, Soft Evidence

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

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

- Not true logically, but gives useful information
- Example

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

# Field Comparison

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

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

- Example

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

- Can have different weight for each word

# 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

# Record Matching

- Simplest Version: Field similarities measured by presence/absence of words in common

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

- Example

$$\text{HasWord}(J. Cox, Cox) \wedge \text{HasWord}(Cox J., Cox) \wedge \text{HasAuthor}(P1, J. Cox) \wedge \text{HasAuthor}(P2, Cox J.) \wedge (Cox = Cox) \Rightarrow (P1 = P2)$$

- Transitivity

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

- Additional Constraints

$$\text{HasAuthor}(c, a_1) \wedge \text{HasAuthor}(c, a_2) \Rightarrow \text{Coauthor}(a_1, a_2) \\ \text{Coauthor}(a_1, a_2) \wedge \text{Coauthor}(a_3, a_4) \wedge a_1 = a_3 \Rightarrow a_2 = a_4$$



# Inference

- Use cheap heuristics (e.g. TFIDF based similarity) to identify plausible pairs
- Inference/learning over plausible pairs
- Inference method: lazy grounding + MaxWalkSAT
- Learning: supervised and transfer (learn/hand set on one domain and transferred)

# Probabilistic Soft Logic

[Broecheler & Getoor, UAI10]

- 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

# 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

# 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

# **HYBRID APPROACHES**

# Hybrid 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]

# Dedupalog [Arasu et al., ICDE09]

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 initial approximate 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 Names Assumption (**UNA**)**

Publishers (UNA) and Papers (NOT UNA)

**Slides based on [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\***( $a_1, a_2$ )  $\leftrightarrow$  AuthorSimilar( $a_1, a_2$ )

Author1	Author2
AA	Arvind Arasu
Arvind A	Arvind Arasu

*“Cluster authors with similar names”*

\*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<sub>1</sub>,id<sub>2</sub>)  $\leftrightarrow$  PaperRef(id<sub>1</sub>,t<sub>1</sub>,-), PaperRef(id<sub>2</sub>,t<sub>2</sub>,-), TitleSimilar(t<sub>1</sub>,t<sub>2</sub>)

**Author\***(a<sub>1</sub>,a<sub>2</sub>)  $\leftrightarrow$  AuthorSimilar(a<sub>1</sub>,a<sub>2</sub>)

( $\leftrightarrow$ ) Soft-constraints:  
*Pay a cost if violated.*

---

**Paper\***(id<sub>1</sub>,id<sub>2</sub>)  $\leq$  PaperEq(id<sub>1</sub>,id<sub>2</sub>)

$\neg$  **Paper\***(id<sub>1</sub>,id<sub>2</sub>)  $\leq$  PaperNeq(id<sub>1</sub>,id<sub>2</sub>)

( $\leq$ ) Hard-constraints: *Any clustering must satisfy these*

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

1. PaperEQ, PaperNEQ are relations (EDBS)
2.  $\neg$  denotes Negation here.

# Additional Constraints

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

**Author\***( $a_1, a_2$ )  $\leftarrow$  **Paper\***( $id_1, id_2$ ), **Wrote**( $id_1, a_1, 1$ ), **Wrote**( $id_2, a_2, 1$ )

---

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

**Publisher\***( $x, y$ )  $\leftarrow$  **Publishes**( $x, p_1$ ), **Publishes**( $x, p_2$ ), **Paper\***( $p_1, p_2$ )

---

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

$\neg$  **Author\***( $x, y$ )  $\leftarrow$   $\neg$  (**Wrote**( $x, p_1, -$ ), **Wrote**( $y, p_2, -$ ), **Wrote**( $z, p_1, -$ ),  
**Wrote**( $z, p_2, -$ ), **Author\***( $x, y$ ))

---

---

# Dedupalog via CC

Semantics: Translate a Dedupalog Program to a set of graphs

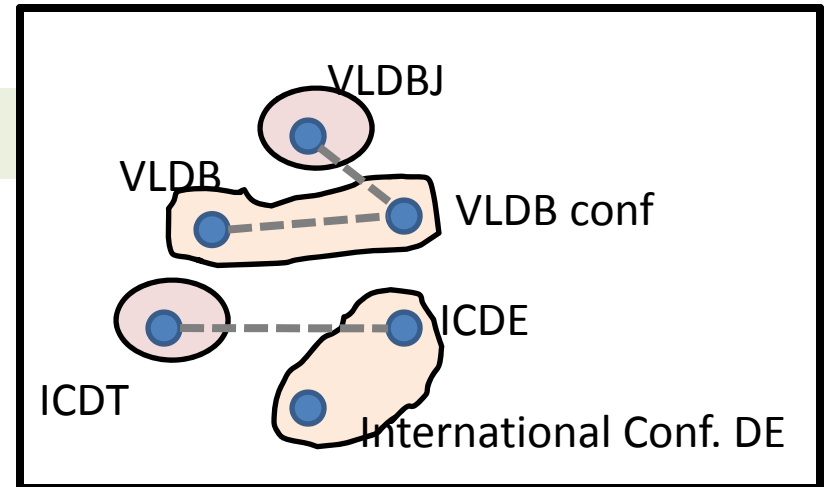
Nodes are references (in the ! Relation)

Entity References: Conference!(c)

**Conference**<sup>\*</sup>(c<sub>1</sub>,c<sub>2</sub>) <-> ConfSim(c<sub>1</sub>,c<sub>2</sub>)

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** $^*(c_1, c_2) \leftarrow \text{ConfSim}(c_1, c_2)$



**Conference** $^*(c_1, c_2) \leq \text{ConfEQ}(c_1, c_2)$

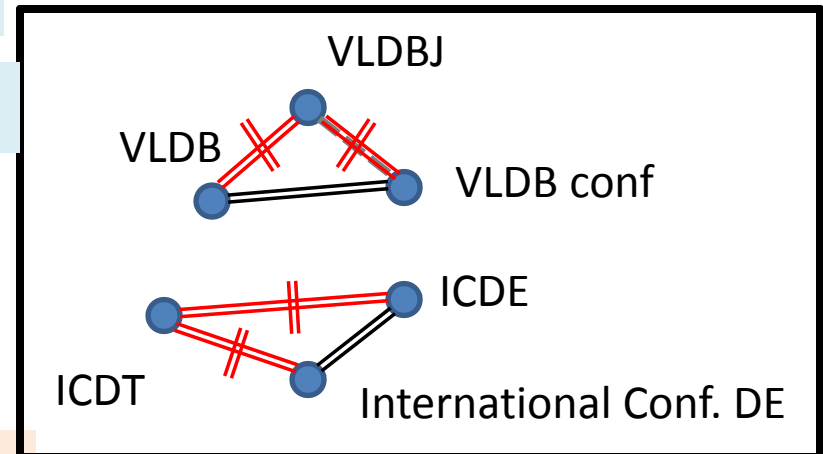
$\neg$ **Conference** $^*(c_1, c_2) \leq \text{ConfNEQ}(c_1, c_2)$





Soft

 Positive  
 Negative

Hard

 Equal  
 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

# Voting

Extend algorithm to **whole** language via *voting technique*.  
Support different entity types, 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.

# Hybrid Approaches

	Method	Evaluation
Constraint-based Entity Matching [Shen, Li & Doan, AAAI05]; builds on (Li, Morie, & Roth, AI Mag 2004)	Two layer model: Layer 1: Generative model for data sets that satisfy constraints; Layer 2: EM algorithm and the relaxation labeling algorithm to perform matching. In each iteration, use EM to estimate parameters of the generative model and a matching assignment, then employs relaxation labeling to exploit the constraints	Researchers and IMDB with noise added
Dedupalog [Arasu, Re, Suciu, ICDE09]	Declarative specification for rich collection of constraints with nice syntactic sugar added to datalog for ER. Inference: Correlation clustering+ voting	Precision/Recall on Cora, subset of ACM dataset

# Summary: Collective Approaches

- Decisions for cluster-membership depends on other clusters
  - Similarity propagation approaches
  - Probabilistic Models
    - Generative Models
    - Undirected Models
  - Hybrid Approaches
- Non-probabilistic approaches often scale better than generative probabilistic approaches
- Undirected/constraint-based models are often easier to specify
- Scaling undirected models active area of research

PART 3

**SCALING ER TO BIG-DATA**



# Scaling ER to Big-Data

- Blocking/Canopy Generation
- Distributed ER

PART 3-a

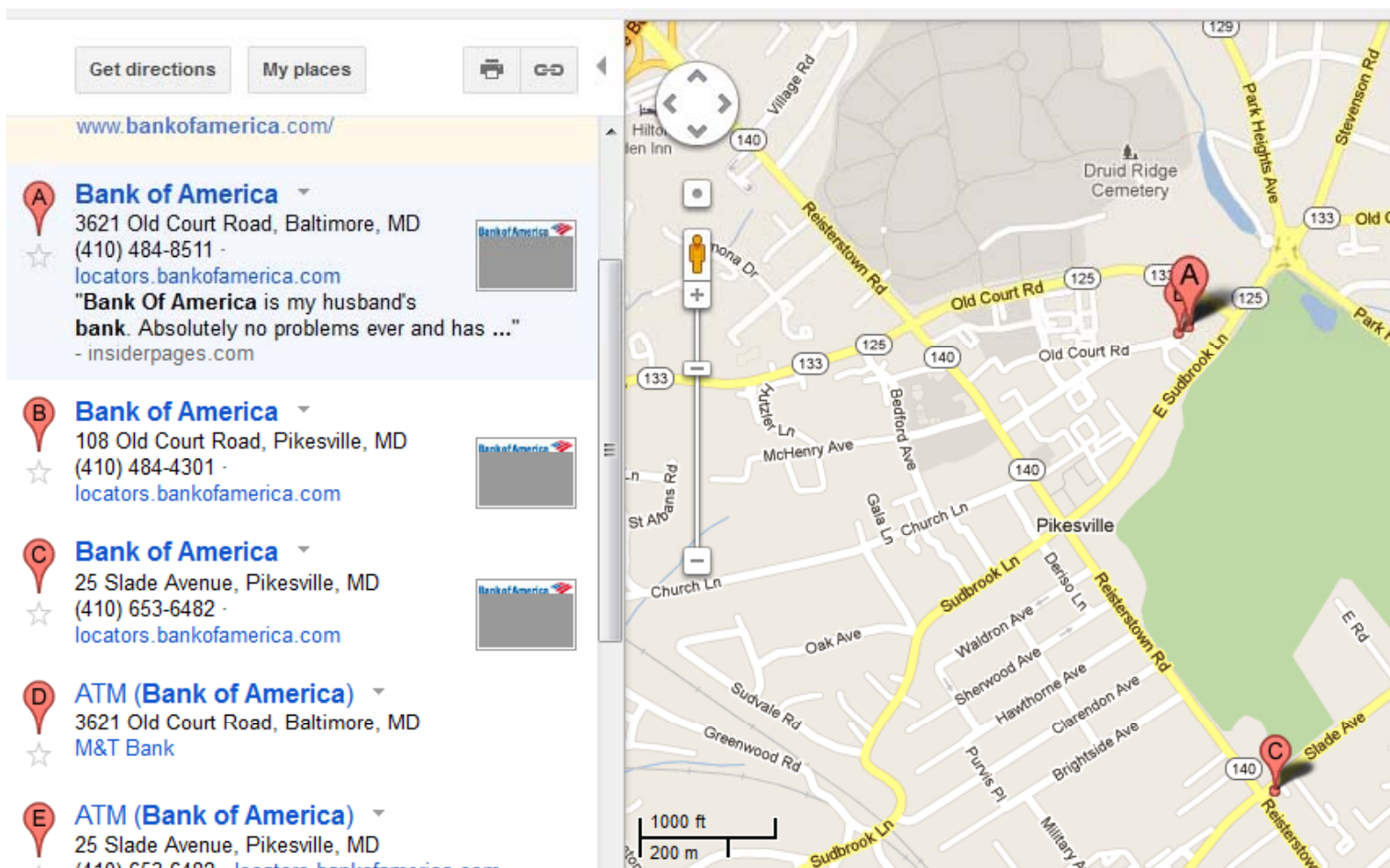
**BLOCKING/CANOPY GENERATION**

# Blocking: Motivation

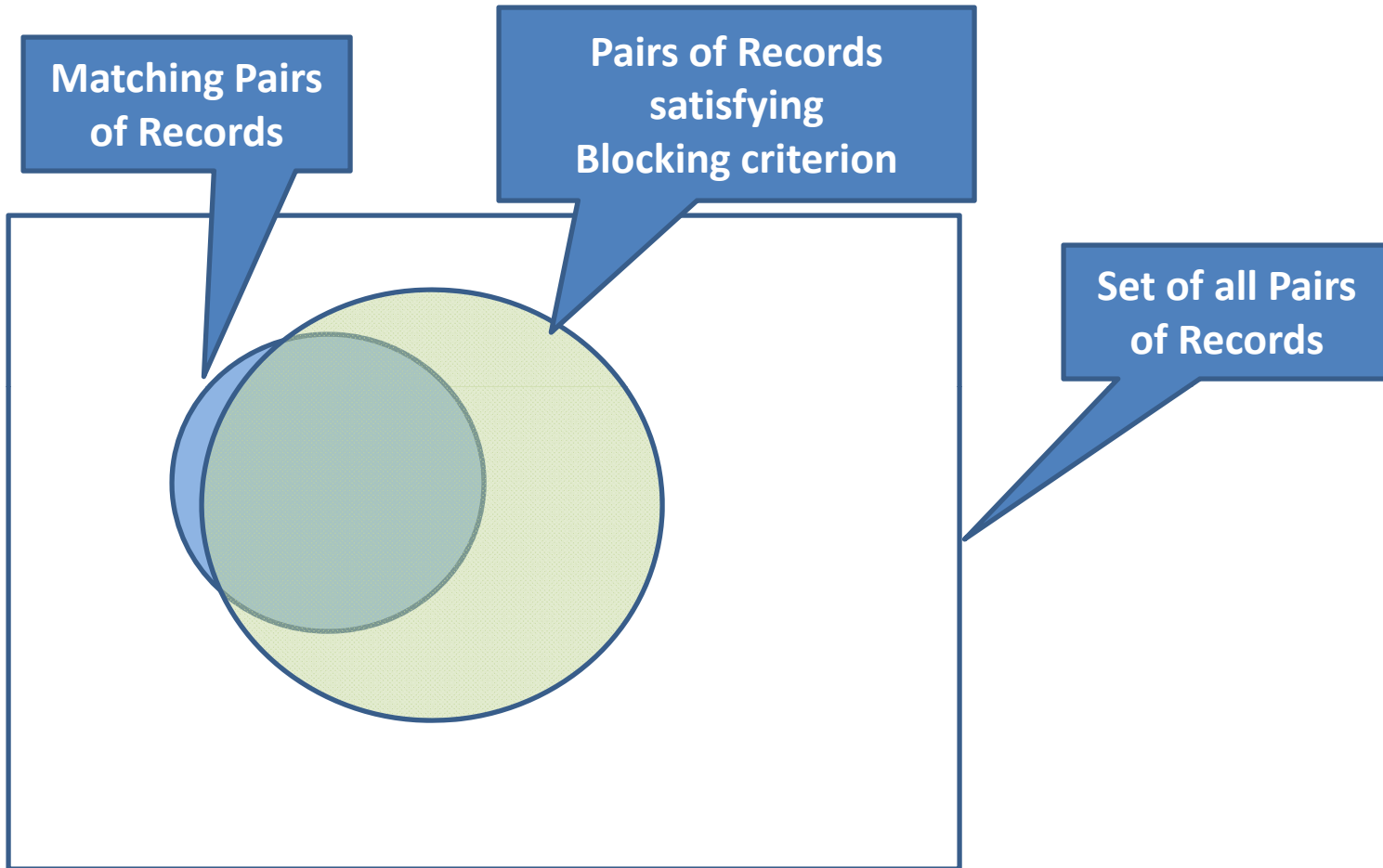
- Naïve pairwise:  $|R|^2$  pairwise comparisons
  - 1000 business listings each from 1,000 different cities across the world
  - 1 trillion comparisons
  - 11.6 days (if each comparison is 1  $\mu$ s)
- Mentions from different cities are unlikely to be matches
  - **Blocking Criterion: City**
  - 1 billion comparisons
  - 16 minutes (if each comparison is 1  $\mu$ s)

# Blocking: Motivation

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



# Blocking: Motivation



# 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 AAI'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

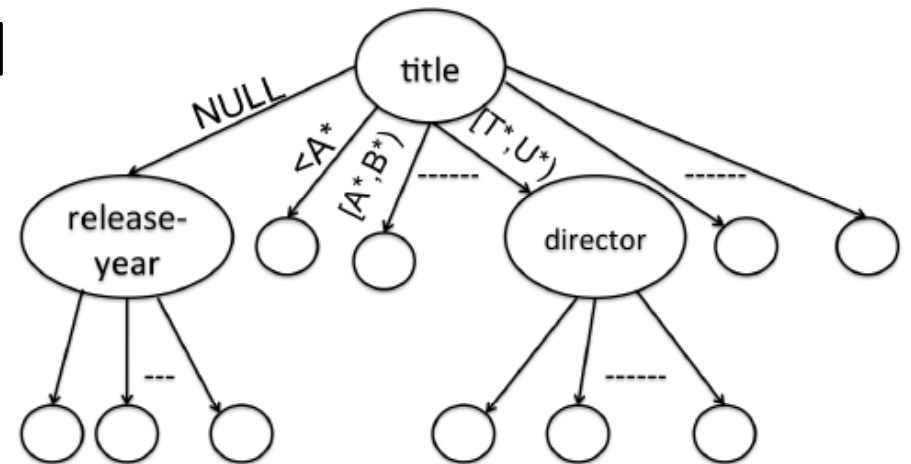


# Learning Optimal Blocking Functions

- 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]



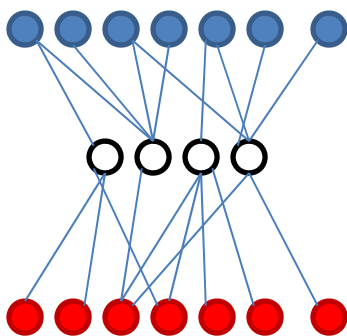
# Learning an Optimal function [Bilenko et al ICDM '06]

- 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

Positive Examples

Blocking Keys

Negative Examples

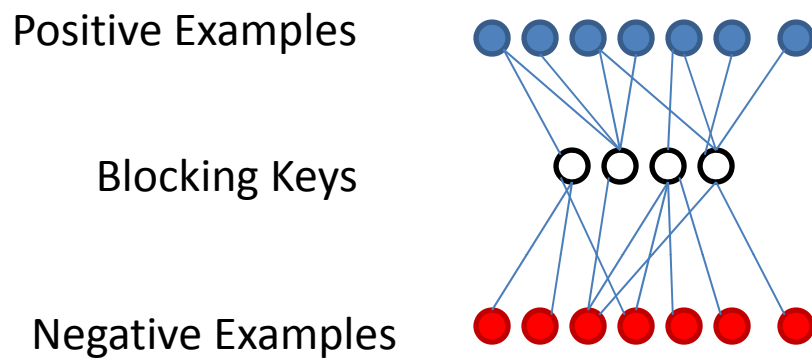


Pick  $k$  Blocking keys such that

- At most  $\epsilon$  blue nodes are not covered
- 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  $\epsilon$  blue nodes are not covered
- (b) Number of red nodes covered is minimized

- Greedy Algorithm:

- Construct “good” conjunctions of blocking keys  $\{p_1, p_2, \dots\}$ .
- Pick  $k$  conjunctions  $\{p_{i_1}, p_{i_2}, \dots, p_{i_k}\}$ , such that the following is minimized

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

# minHash (Minwise Independent Permutations)

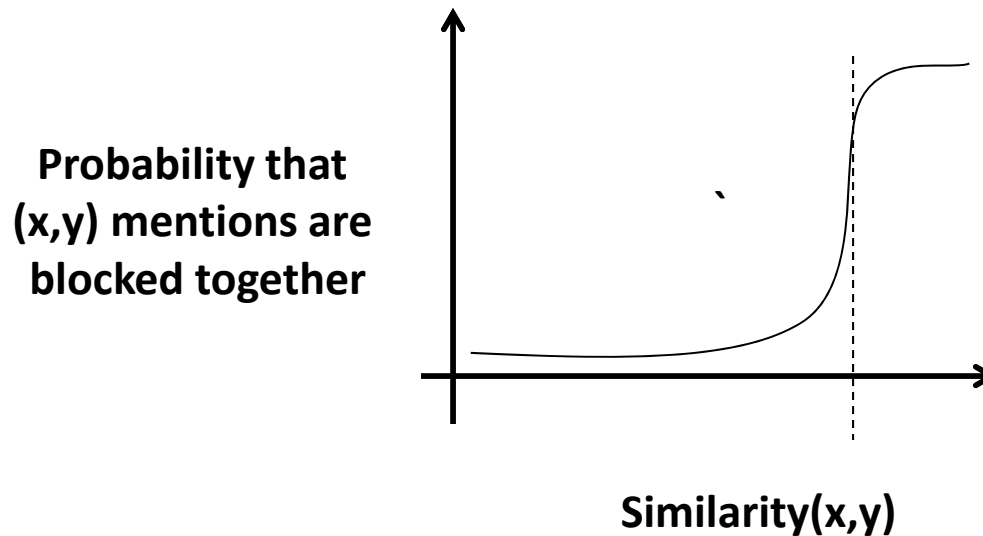
- Let  $F_x$  be a set of features for mention  $x$ 
  - (functions of) attribute values
  - character ngrams
  - optimal blocking functions ...
- Let  $\pi$  be a random permutation of features in  $F_x$ 
  - E.g., order imposed by a random hash function
- $minHash(x) =$  minimum element in  $F_x$ 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

# Why minHash works?

**Surprising property:** For a random permutation  $\pi$ ,

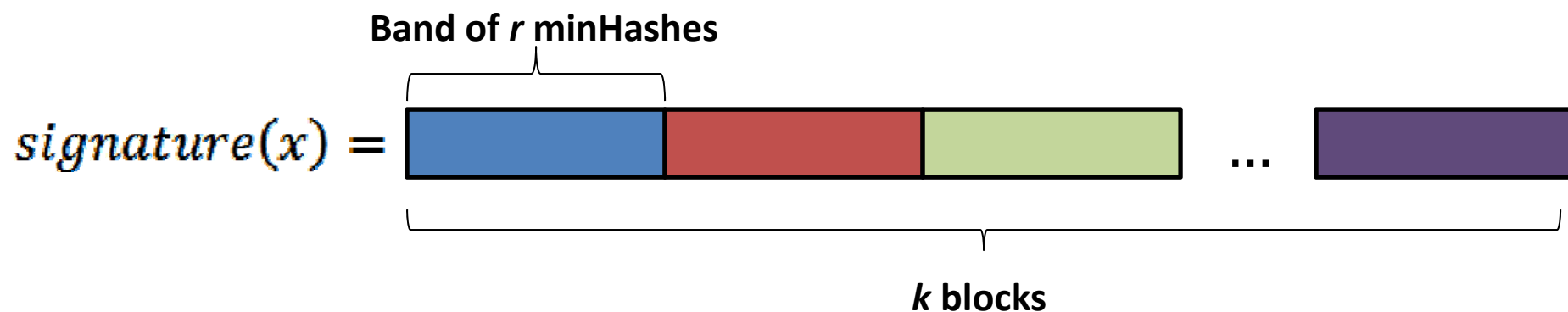
$$P(\text{minHash}(x) = \text{minhash}(y)) = \frac{F_x \cap F_y}{F_x \cup F_y}$$

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)



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

# minHash Analysis

$r = 5, k = 20$

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) =  $k \times s^r$

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

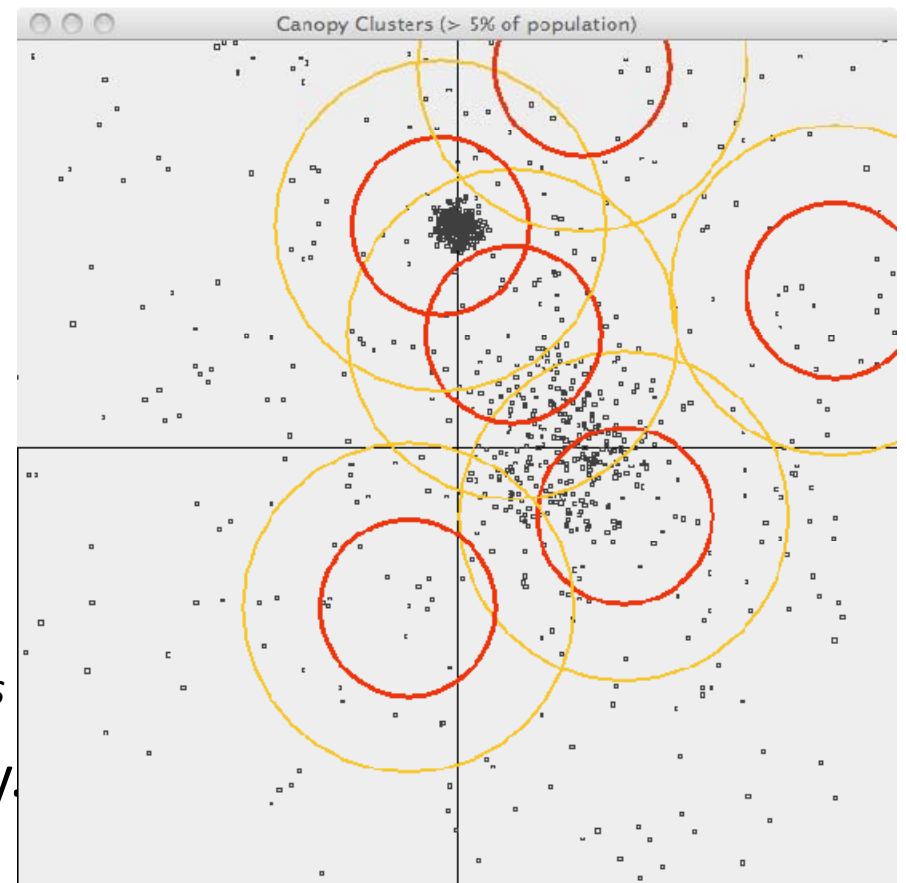


# Canopy Clustering [McCallum et al KDD'00]

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.  $d(x,y) < T_2$  (from consideration in this
4. Return to Step 1 if  $M$  is not empty.



PART 3-b

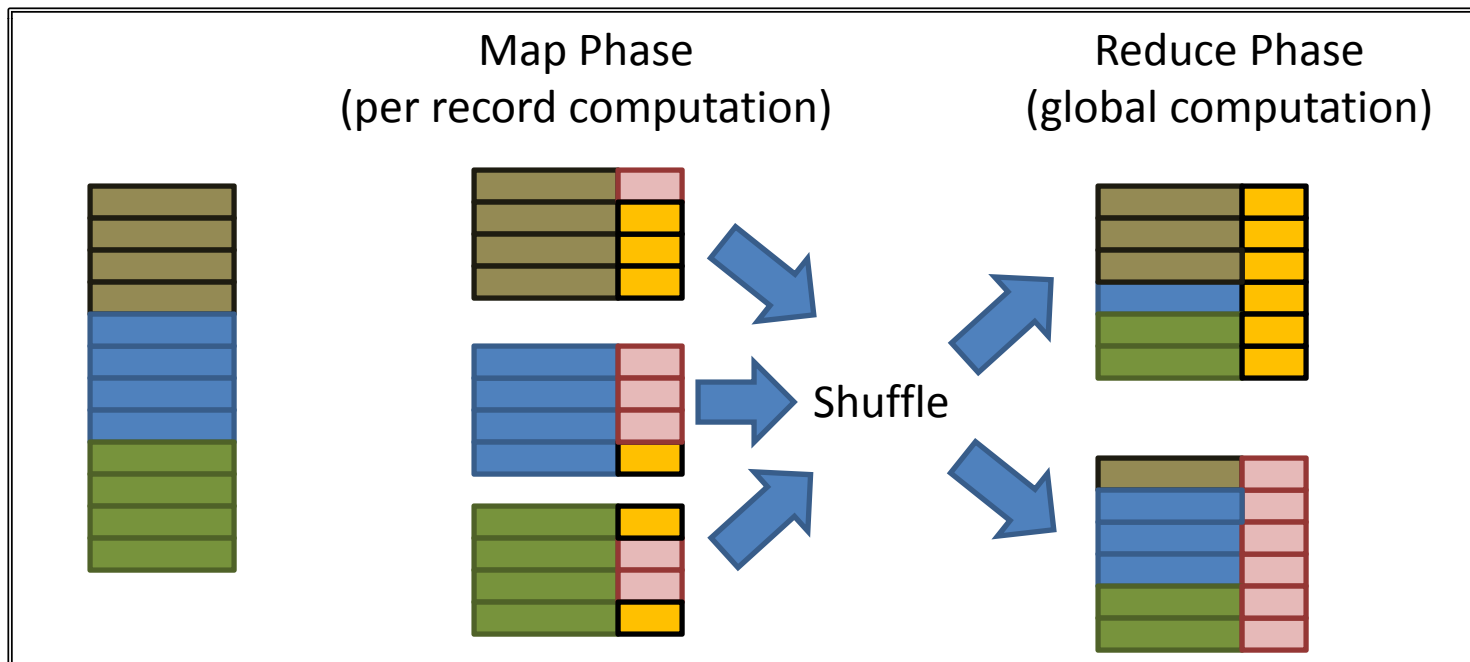
**DISTRIBUTED ER**

# Distributed ER

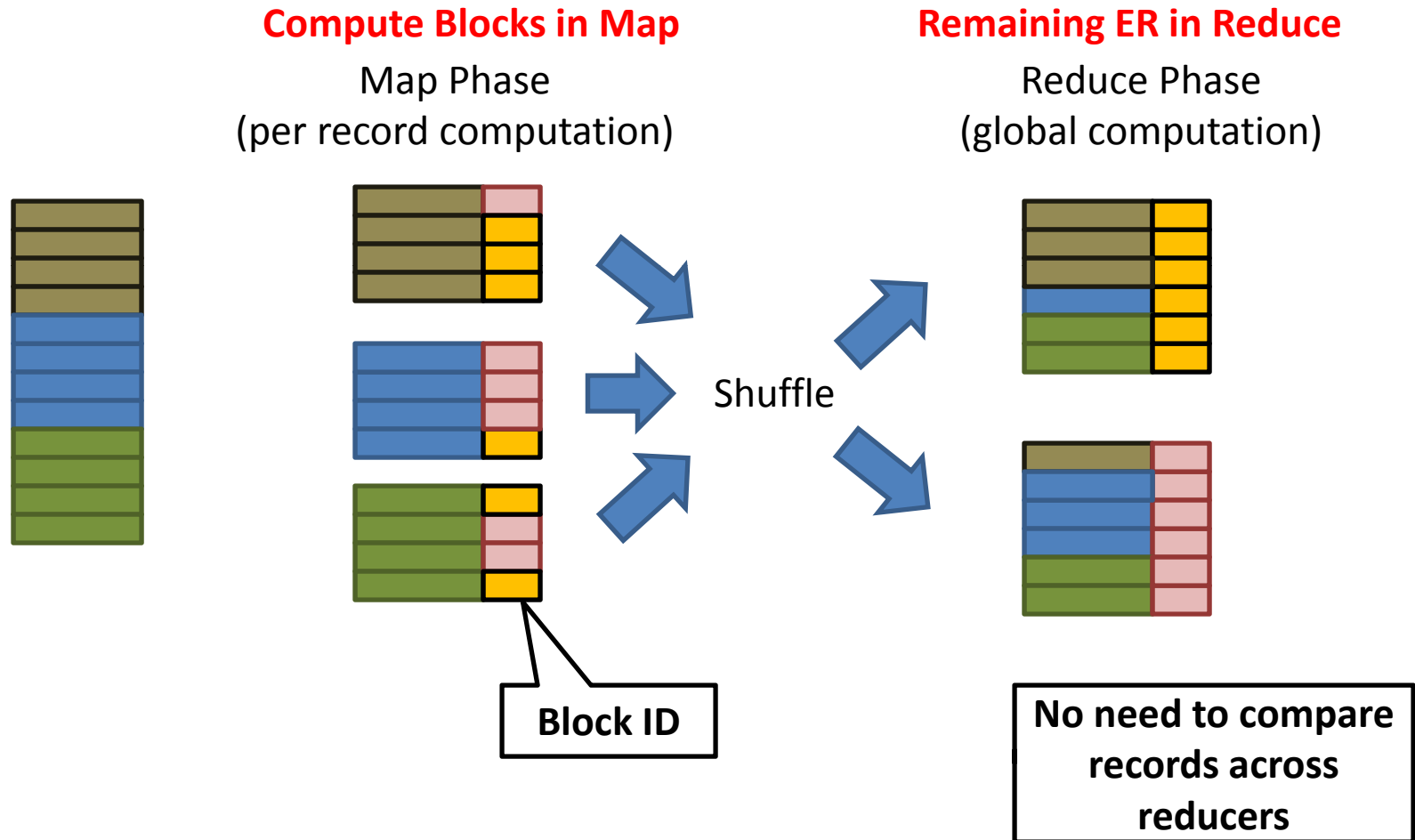
- Map-reduce is very popular for large tasks
  - Simple programming model for massively distributed data

```
map    (k1,v1)    → list(k2,v2);  
reduce (k2,list(v2)) → list(k3,v3).
```

- Hadoop provides fault tolerance and is open source



# ER with Disjoint Blocking



# Non-disjoint Blocking

- How to block?
  - Hash-based: need an efficient technique to group records if they match on *n-out-of-k* blocking keys [Vernica et al SIGMOD'10]
  - Distance-based: canopy clustering on map-reduce [Mahout]
  - Iterative Blocking [Whang et al SIGMOD '09]

## Problem: Information needed for a record is in multiple reducers.

- Information needed for a record is in multiple reducers.
  - Example 1:
    - Reducer 1: “a” matches with “b”
    - Reducer 2: “a” matches with “c”
    - Need to communicate in order to correctly resolve “a”, “b”, “c”

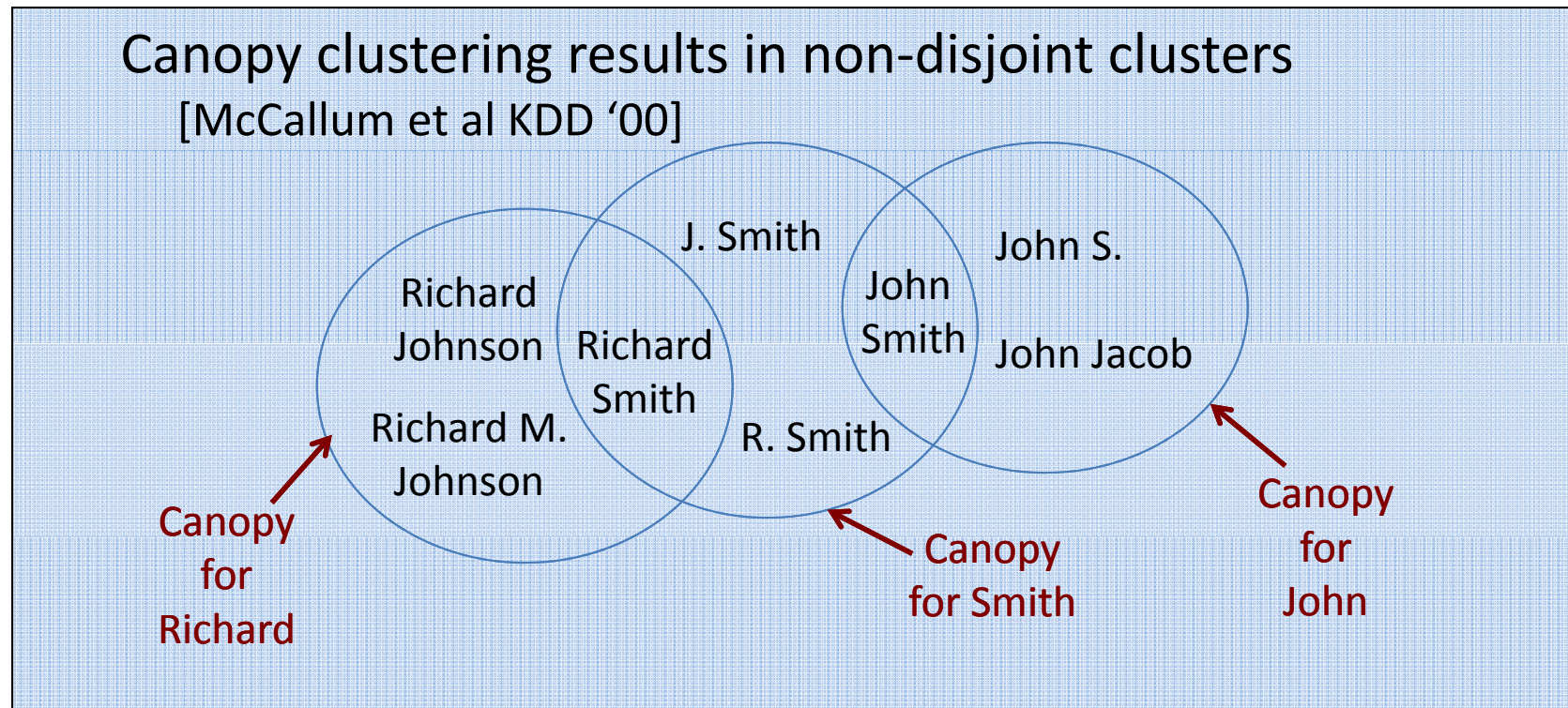
Problem: Information needed for a record is in multiple reducers.

**Example 2:** Dedup papers and authors

Id	Author-1	Author-2	Paper
A <sub>1</sub>	John Smith	Richard Johnson	Indices and Views
A <sub>2</sub>	J Smith	R Johnson	SQL Queries
A <sub>3</sub>	Dr. Smyth	R Johnson	Indices and Views

*Slide adapted from [Rastogi et al VLDB11] talk*

# Problem: Information needed for a record is in multiple reducers.



*Slide adapted from [Rastogi et al VLDB11] talk*

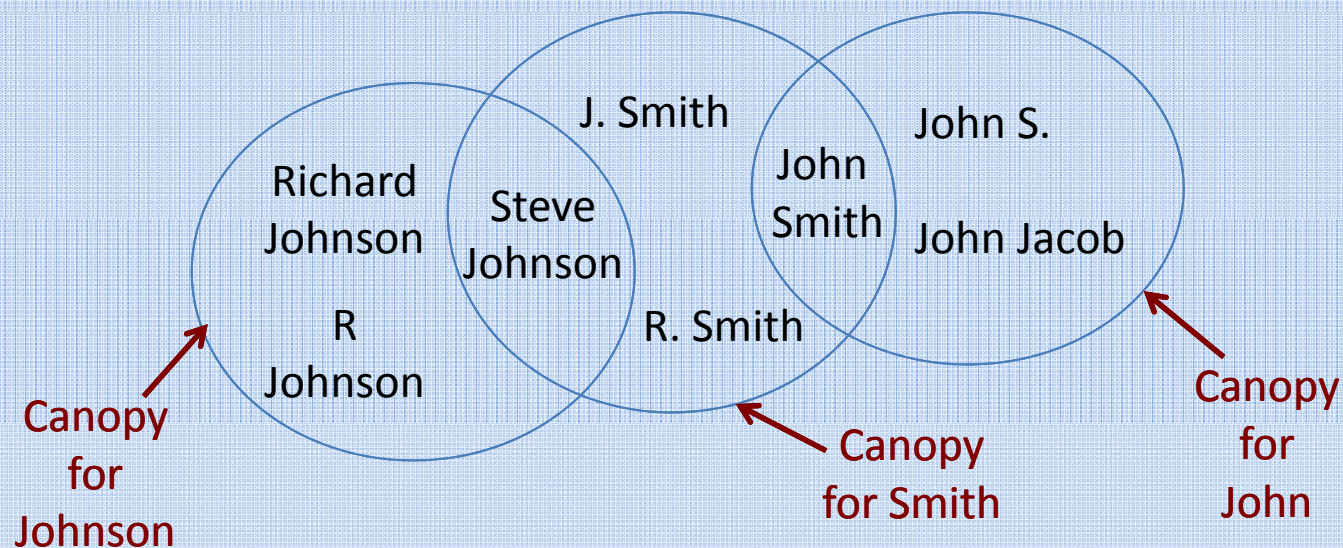


# Problem: Information needed for a record is in multiple reducers.

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

CoAuthor rule grounds to the correlation

$\text{match}(\text{Richard Johnson}, \text{R Johnson}) \Rightarrow \text{match}(\text{J. Smith}, \text{John Smith})$



Slide adapted from [Rastogi et al VLDB11] talk

# Problem: Information needed for a record is in multiple reducers.

Solution 1: Efficiently find Connected Components [Rastogi et al 2012, Kang et al ICDM 2009]  
+ Correlation Clustering / Collective ER in each component

Solution 2: Correlation Clustering / Collective ER in each canopy  
+ Message Passing [Rastogi et al VLDB'11]

# Problem: Information needed for a record is in multiple reducers.

Solution 1: Efficiently find Connected Components [Rastogi et al 2012, Kang et al ICDM 2009]  
+ Correlation Clustering / Collective ER in each component

**Connected components can be large in relational/multi-entity ER.**

Solution 2: Correlation Clustering / Collective ER in each canopy  
+ Message Passing [Rastogi et al VLDB'11]

# 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  $\leq$  no. of matches

**Consistency:** Output independent of the canopy order

**Soundness:** Each output match is actually a true match

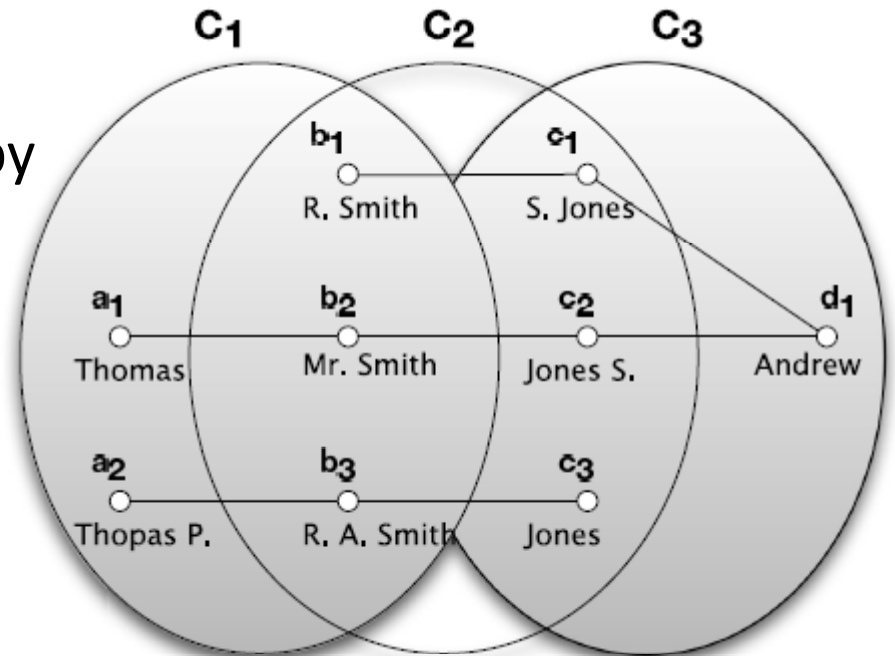
~~**Completeness:** Each true match is also a output match~~



# Completeness

Papers 2 and 3 match only if a canopy knows that

- $\text{match}(a_1, a_2)$
- $\text{match}(b_2, b_3)$
- $\text{match}(c_2, c_3)$



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 Scalability

- $O(|R|^2)$  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.
- Neighborhood based Blocking:
  - Form overlapping canopies of records based on similarity.
  - Only compare records within a cluster.
- Computing connected components/Message Passing in addition to blocking can help distribute 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]
- 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?

# Open Issues

- 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?
- ER Benchmarks
  - Need for large-scale real-world ER datasets with groundtruth
  - Synthetic data useful for scaling but hard to capture rich complexities of real world

# 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/user generated 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
- T. Herzog et al, “Data Quality and Record Linkage Techniques”, Springer 2007
- A. Elmagrid et al, “Duplicate Record Detection”, TKDE 2007
- P. Christen, “Data Matching”, Springer 2012
- N. Koudas et al, “Record Linkage: Similarity measures and Algorithms”, SIGMOD 2006
- X. Dong & F. Naumann, “Data fusion--Resolving data conflicts for integration”, VLDB 2009
- L. Getoor & A. Machanavajjhala, “Entity Resolution: Theory, Practice and Open Challenges”, AAAI 2012

# 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

# References – Single Entity ER (contd.)

- 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
- J. Wang et al, “CrowdER: Crowdsourcing Entity Resolution”, PVLDB 5(11), 2012
- A. Marcus et al, “Human-powered Sorts and Joins”, PVLDB 5(1), 2011

# 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 – Constraints & Multi-Relational ER

- R. Ananthakrishna et al, “Eliminating fuzzy duplicates in data warehouses”, VLDB 2002
- A. Arasu et al, “Large-Scale Deduplication with Constraints using Dedupalog”, ICDE 2009
- S. Chaudhuri et al., “Leveraging aggregate constraints for deduplication”, SIGMOD07
- X. Dong et al, “Reference Recconciliation 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
- P. Bohannon et al., “Conditional Functional Dependencies for Data Cleaning”, ICDE 2007
- M. Broecheler & L. Getoor, “Probabilistic Similarity Logic”, UAI 2010
- W. Fan, “Dependencies revisited for improving data quality”, PODS 2008
- 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”, AAI 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”, AAI 2005
- P. Singla & P. Domingos, “Entity Resolution with Markov Logic”, ICDM 2006
- Whang et al., “Generic Entity Resolution with Negative Rules”, VLDBJ 2009
- Whang et al., “Joint Entity Resolution”, ICDE 2012

# 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”, AAI 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
- G. Papadias et al, “Beyond 100 million entities: large-scale blocking-based resolution for heterogenous data,” WSDM 2012
- 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
- L. Kolb et al, “Dedoop: Efficient deduplication with Hadoop”, (demo) PVLDB 5(12), 2012
- R. Vernica et al, “Efficient Parallel Set-Similarity Joins Using MapReduce”, SIGMOD 2010
- Apache Mahout: Scalable Machine Learning and Data Mining, <http://mahout.apache.org/>
- S. Whang et al, “Entity Resolution with Iterative Blocking”, SIGMOD 2009
- U. Kang et al, “PEGASUS: A Peta-Scale Graph Mining System - Implementation and Observations”, ICDM 2009
- V. Rastogi et al, “Finding Connected Components on Map-reduce in Poly-Log Rounds”, Corr 2012
- 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. Berté-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