



**WISCONSIN**  
UNIVERSITY OF WISCONSIN-MADISON

# CS639: Data Management for Data Science

Lecture 22: Entity Resolution

[slides from Getoor and Machanavajjhala]

Theodoros Rekatsinas

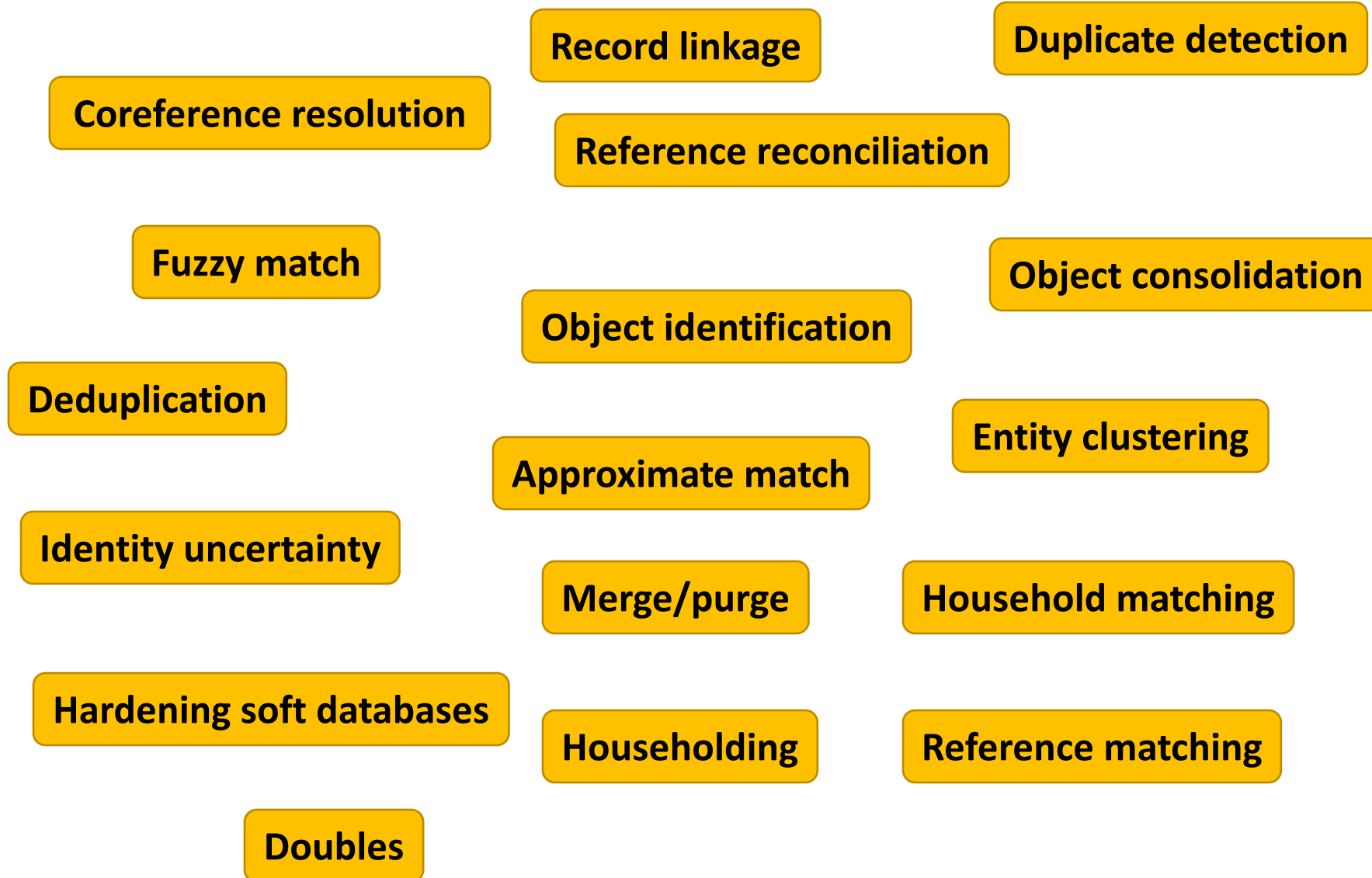
# What is Entity Resolution?

*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.
- ...

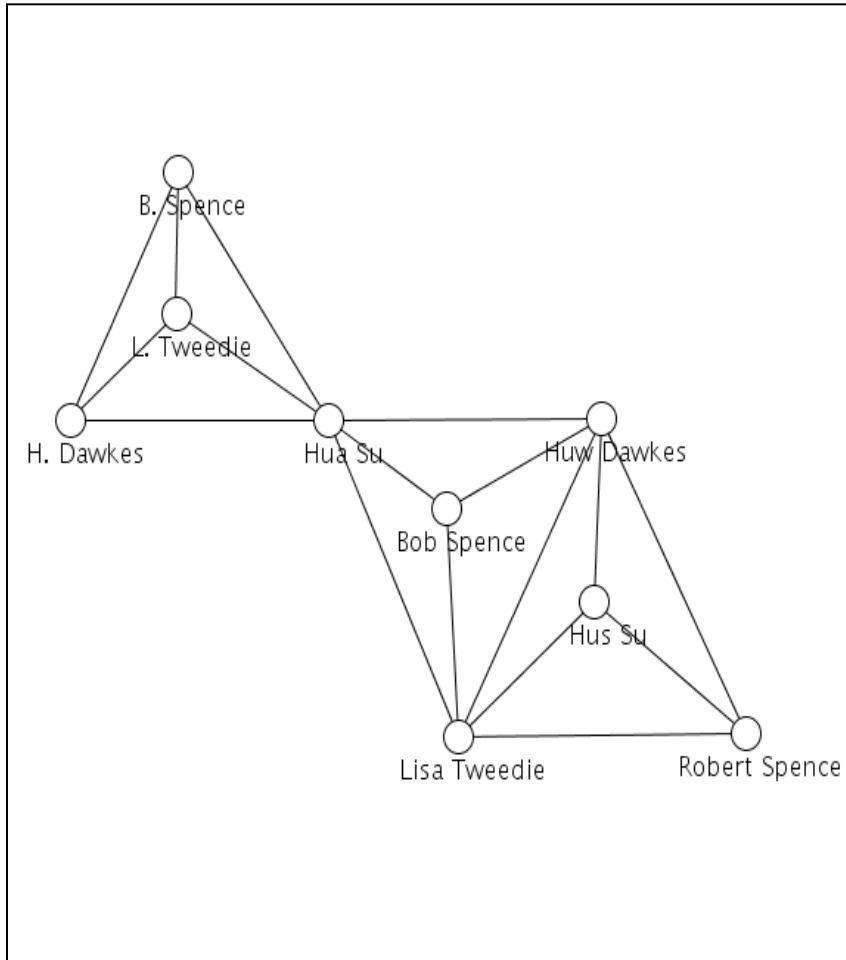
Ironically, Entity Resolution has many duplicate names



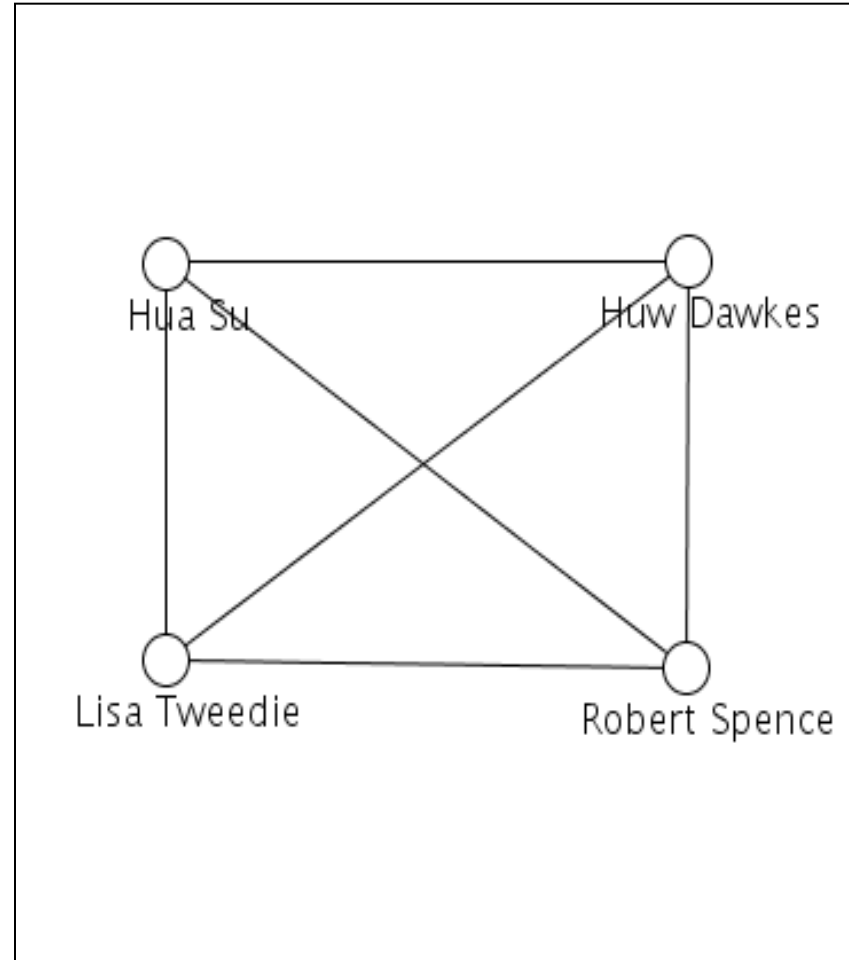
# ER Motivating Examples

- *Linking Census Records*
- *Public Health*
- *Web search*
- *Comparison shopping*
- *Counter-terrorism*
- *Knowledge Graph Construction*
- ...

# Motivation: ER and Network Analysis



before



after

# Motivation: ER and Network Analysis

- Measuring the topology of the internet ... using traceroute

```
Command Prompt
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]

Trace complete.

C:\>
```

# IP Aliasing Problem [Willinger et al. 2009]

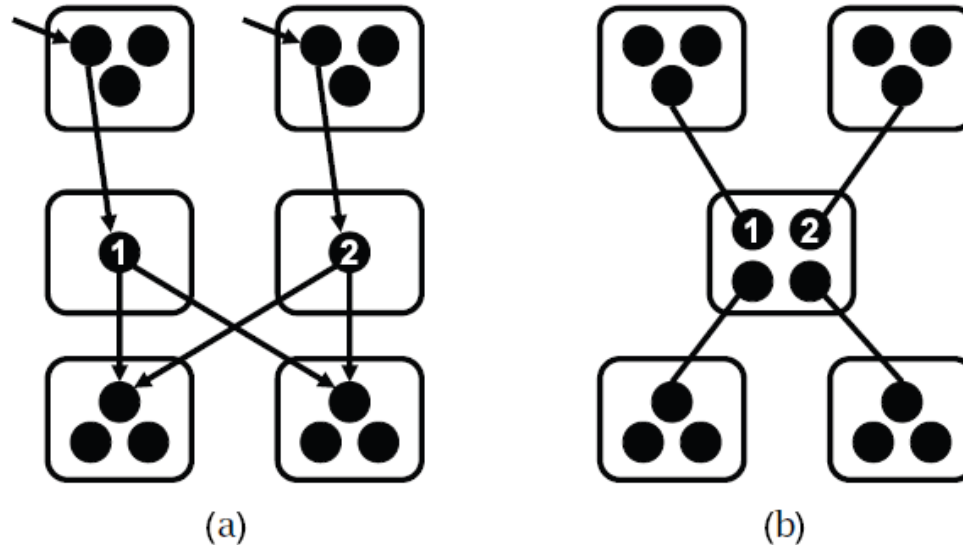


Figure 2. The IP alias resolution problem. Paraphrasing Fig. 4 of [50], traceroute does not list routers (boxes) along paths but IP addresses of input interfaces (circles), and alias resolution refers to the correct mapping of interfaces to routers to reveal the actual topology. In the case where interfaces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

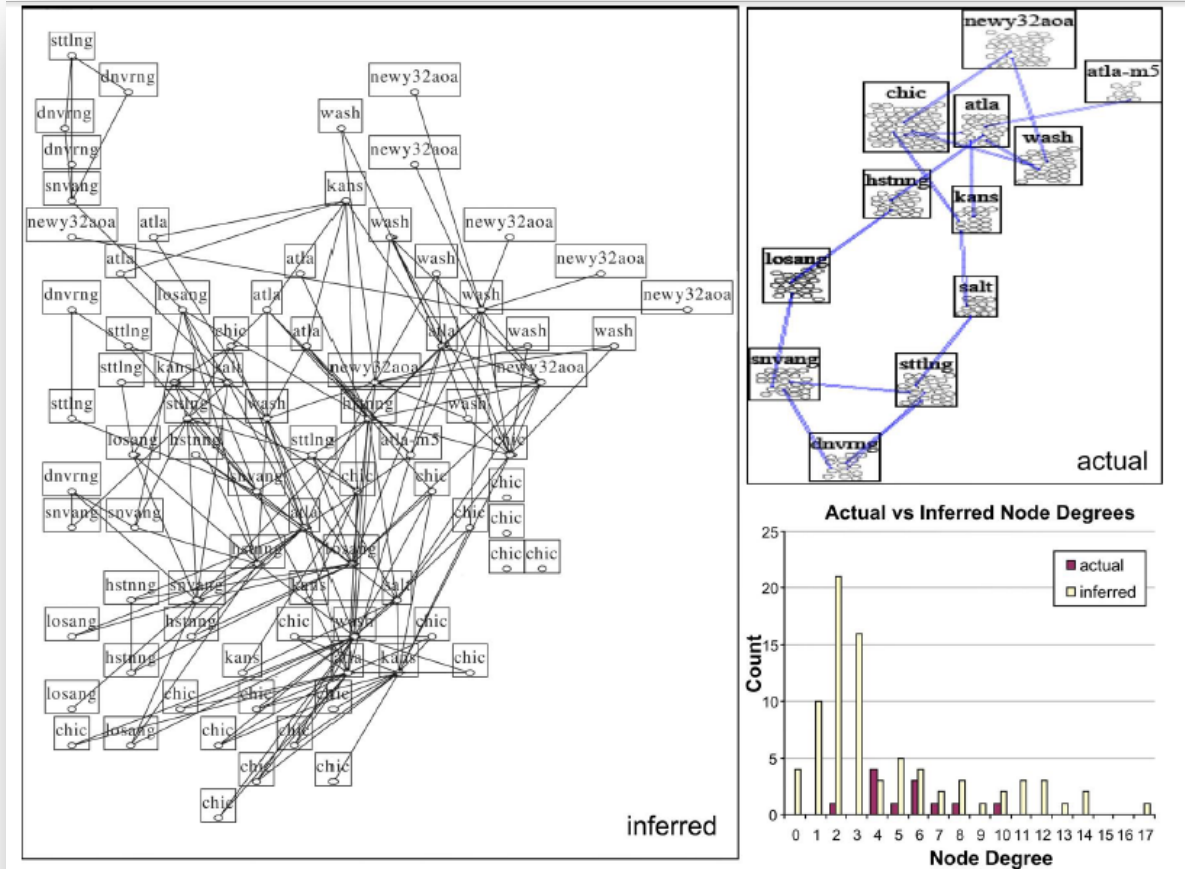
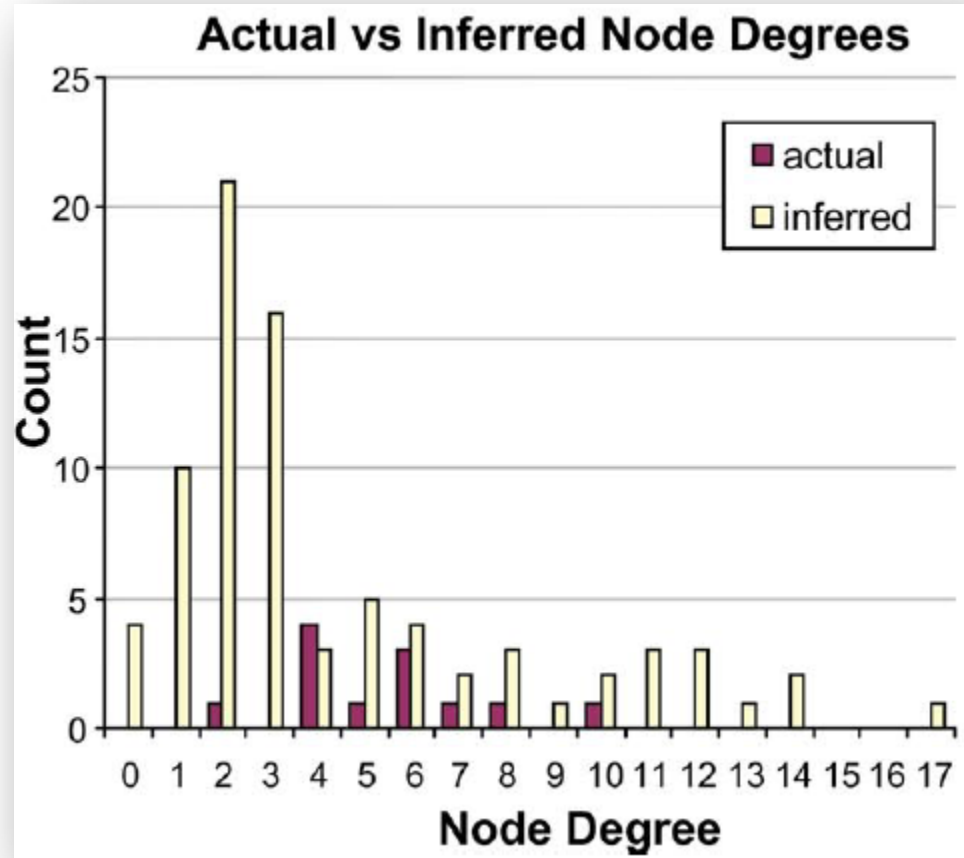


Figure 3. The IP alias resolution problem in practice. This is re-produced from [48] and shows a comparison between the Abilene/Internet2 topology inferred by Rocketfuel (left) and the actual topology (top right). Rectangles represent routers with interior ovals denoting interfaces. The histograms of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



# IP Aliasing Problem [Willinger et al. 2009]



# Normalization

- Schema normalization
  - Schema matching: e.g., *contact#* vs. *phone*
  - Compound attributes: e.g., *addr* vs. (*street, city, st, zip*)
  - Nested or set-valued attributes: e.g., properties for rent with a set of tags, multiple phone numbers
- Data normalization
  - Capitalization, white-space normalization
  - Correcting typos, replacing abbreviations, variations, nick names
    - Usually done by employing “dictionaries”: e.g., lists of businesses, postal addresses, etc.

# Matching Features

Give two records , compute a “comparison” vector of similarity scores for corresponding features

- E.g., to match two bibliographical references, compute  $\langle 1^{\text{st}}\text{-author-match-score}, \text{title-match-score}, \text{venue-match-score}, \text{year-match-score}, \dots \rangle$
- Score can be Boolean (match, or mismatch), or reals (based on some distance function)

# Examples of matching features

- Difference between numeric values
- Domain-specific, like Jaro (for names)
- Edit distance: good for typos in strings
  - Levenshtein, Smith-Waterman, affine gap
- Phonetic-based
  - Soundex
- Translation-based
- Set similarity
  - Jaccard, Dice
  - For text fields (set of words) or relational features (e.g., set of authors of a paper)
- Vector-based
  - Cosine similarity, TF/IDF (good for text)

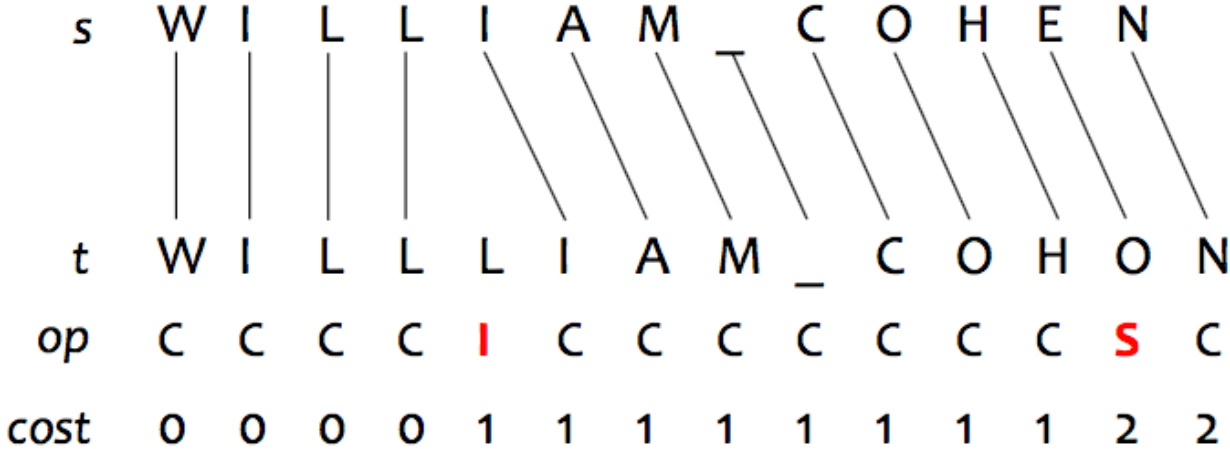
# Jaro

Specifically designed for names by U.S. Census

- Given  $s$  and  $t$ ,  $c$  is common if  $s_i = t_j = c$  and  $|i - j| \leq \frac{\min(|s|, |t|)}{2}$
- $c_1$  and  $c_2$  are a *transposition* if  $c_1$  and  $c_2$  are common but appear in different orders in  $s$  and  $t$
- Jaro similarity =  $\frac{1}{3} \left( \frac{m}{|s|} + \frac{m}{|t|} + \frac{m-x}{2m} \right)$ , where  $m = \#$  commons and  $x =$  some measure of  $\#$  transpositions
- Jaro-Winkler further weighs errors early in the strings more heavily

# Levenshtein

- Distance between strings  $s$  and  $t$  = shortest sequence of edit commands that transform  $s$  to  $t$ 
  - Copy character from  $s$  over to  $t$
  - Delete a character in  $s$  (cost 1)
  - Insert a character in  $t$  (cost 1)
  - Substitute one character for another (cost 1)



# Computing Levenshtein

$D(i, j)$  = score of best alignment between  $s_1 s_2 \cdots s_i$   
and  $t_1 t_2 \cdots t_j$

$$= \min \begin{cases} D(i-1, j-1) + d(s_i, t_j) & \text{sub/copy} \\ D(i-1, j) + 1 & \text{delete} \\ D(i, j-1) + 1 & \text{insert} \end{cases}$$

where  $d(s_i, t_j) = \mathbf{1}[s_i \neq t_j]$ ,

and let  $D(0,0) = 0$ ,  $D(i, 0) = i$ , and  $D(0, j) = j$

- Can then normalize using lengths of  $s$  and  $t$ :  
 $1 - D(|s|, |t|) / \max(|s|, |t|)$

# Set similarity

Given two sets  $A$  and  $B$

- Jaccard distance:  $1 - \frac{|A \cap B|}{|A \cup B|}$
- Dice distance:  $1 - \frac{2|A \cap B|}{|A| + |B|}$ 
  - Not a distance metric (triangle inequality doesn't hold)
  - Note the connection to the F1 measure, which is the harmonic mean of
    - Precision:  $TP / (TP + FP)$
    - Recall:  $TP / (TP + FN)$



# Cosine similarity and TF/IDF

- Let  $U = \{x_1, x_2, \dots, x_n\}$  be the universe of all elements (e.g., possible words in English)
- A multiset  $D$  with elements drawn from  $U$  (e.g., a document) can be represented as an  $n$ -dim vector  $\langle w_1, w_2, \dots, w_n \rangle$ 
  - Each  $w_i$  can be as simple as  $c(D, x_i)$ , count of  $x_i$  in  $D$
- Cosine similarity between  $D_1$  and  $D_2$  is  $\frac{D_1 \cdot D_2}{|D_1| |D_2|}$ , where  $|\cdot|$  is the  $L_2$  (Euclidean) normal

# TF/IDF

Alternatively, if you have a corpus  $\mathcal{D}$  of  $D$ 's, define

- Term frequency  $TF(D, x) = \log_{10}(1 + c(D, x))$ , where  $c(D, x)$  is  $x$ 's number of occurrences in  $D$
- Inverse document frequency  $IDF(\mathcal{D}, x) = \log_{10}\left(\frac{|\mathcal{D}|}{DF(\mathcal{D}, x)}\right)$ , where  $DF(\mathcal{D}, x)$  is the number of  $D$ 's in  $\mathcal{D}$  containing  $x$
- Let  $w_i = TF(D, x_i) \cdot IDF(\mathcal{D}, x_i)$ 
  - Idea: elements that don't serve to distinguish a  $D$  within  $\mathcal{D}$  (e.g., stop words) are weighed down

# Tokening and shingling

What are the “elements” in text?

Do we lose the sequencing information by treating text as a bag of elements?

- Simply split by non-alphanumeric characters?
  - How about “San Francisco”?
  - Can use a language model to find sequences of words that appear “more than random”
- Or additionally treat  $n$ -grams (all subsequences of length  $n$ ) as your “elements” (shingling)

# Pairwise-ER

Given a vector of component-wise similarity scores for records  $x$  and  $y$ , compute  $P(x \text{ and } y \text{ match})$

Possible solutions

- Check the weighed sum of component-wise scores against a threshold to determine match/non-match
  - E.g.,  $0.5 \times 1^{\text{st}}\text{-author-match-score} + 0.2 \times \text{venue-match-score} + 0.3 \times \text{title-match-score} \geq 0.8$
- Formulate rules about what constitutes a match
  - E.g.,  $(1^{\text{st}}\text{-author-match-score} > 0.7 \text{ AND } \text{venue-match-score} > 0.8) \text{ OR } (\text{title-match-score} > 0.9 \text{ AND } \text{venue-match-score} > 0.9)$

Hard to come up with weights, thresholds, and rules!

# Fellegi and Sunter

- Given record pair  $r = (x, y)$  to match, with  $\gamma$  as the score vector
- Let  $M$  denote matches and  $U$  non-matches
- Decision rule:

$$R = \frac{P(\gamma \mid r \in M)}{P(\gamma \mid r \in U)}$$

- Non-match if  $R \leq t_l$ , match if  $t_u \leq R$ , uncertain otherwise
- Naïve Bayes assumption:  
$$P(\gamma \mid r \in M) = \prod_i P(\gamma_i \mid r \in M)$$

# Supervised ML for pairwise ER

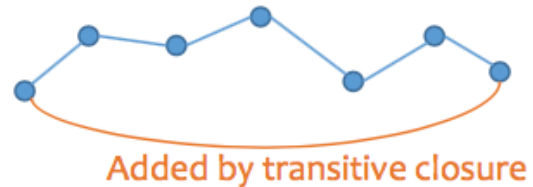
- Naïve Bayes, decision trees (Cochinwala et al., *IS* 2001), support vector machines (Bilenko & Mooney, *KDD* 2003; Christen *KDD* 2008), ensembles of classifiers (Chen et al., *SIGMOD* 2009), Conditional Random Fields (Gupta & Sarawagi, *VLDB* 2009), etc.
- Imbalanced classes: typically many more negatives ( $O(|R|^2)$ ) than positives ( $O(|R|)$ )
- Pairs/matches are not i.i.d.
  - E.g.,  $(x, y) \in M$  and  $(y, z) \in M$  implies  $(x, z) \in M$
- Constructing a training set is hard
  - Most pairs are “easy non-matches”
  - Some pairs are inherently ambiguous (e.g., is Paris Hilton person or business?); others have missing attributes (e.g., Starbucks, Durham, NC)

# Active learning

- Focus labeling efforts to reduce the “confusion region” of classifiers
- To assess uncertainty, use the classifier’s output (e.g., posterior probabilities of a Bayesian classifier), or votes by a “committee” (multiple weak classifiers)
- Again, beware of evaluation metric—0-1 loss is no good; need maximize recall with acceptable precision

# Constraints under deduplication

- Deduplication: given a database containing potential duplicate mentions of the same entities, partition mentions into equivalence classes
- Transitivity constraint:
  - If  $(x, y) \in M$  and  $(y, z) \in M$ , we must have  $(x, z) \in M$
  - Pairwise ER may or may not give us  $(x, z)$  in this case
- A quick fix—compute transitive closure on the inferred match relationships?
  - Bad idea in some cases: graphs resulting from pairwise ER can have diameter  $> 20$  (Rastogi et al. *Corr* 2012)





# Clustering-based ER

- Resolution decisions are not made independently for each pair of records—good
- Unsupervised—good, although often still needs pairwise similarity as input
- Existing clustering algorithms may be used, but
  - Number of clusters not known in advance
  - Many, many small (possibly singleton) clusters—not what most existing clustering algorithms expect

# Possible clustering approaches

- Hierarchical clustering
  - Bilenko et al. *ICDM* 2005
- Nearest-neighbor-based methods
  - Chaudhuri et al., *ICDE* 2005
- Correlation clustering
  - Soon et al. *CL* 2001, Ng et al. *ACL* 2002, Bansal et al. *ML* 2004, Elsner et al. *ACL* 2008, Ailon et al. *JACM* 2008, etc.

# Correlation clustering

- **Key advantage:** no need to give the number of clusters; find the optimal number automatically
- **Key idea:** maximize the sum of
  - Similarities between nodes within the same clusters
  - Disimilarities between nodes in different clusters

# 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