Entity and Identity Resolution

MIT IQ Industry Symposium

July 14, 2010

John Talburt, PhD, CDMP
Department of Information Science

UALR | University of Arkansas at Little Rock

Background

- Professor of Information Science, University of Arkansas at Little Rock
- Coordinator for IQ Graduate Prgm
- Director, ERIQ Laboratory (**E**ntity **R**esolution and **I**nformation **Q**uality) **ualr.edu/eriq/**
- 10 Years in R&D at Acxiom Corporation

Topics

- Principles of Entity Resolution
- Entity Resolution Models

PRINCIPLES OF ER

Pair-wise Definition

- ER is the process of determining whether two references to real-world objects are referring to the same, or to different, objects.
- **Entity** – because of the real-world object
- **Resolution** – because it poses a question

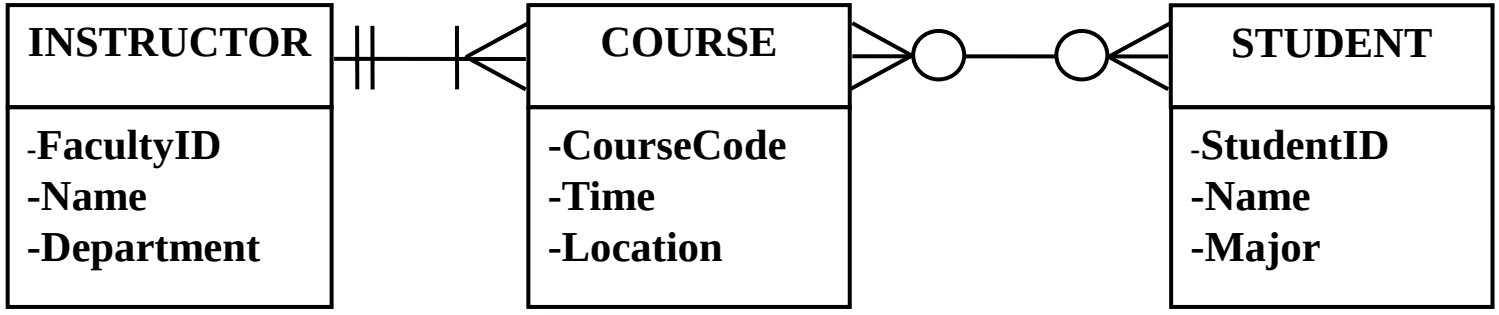
Dataset Definition

- The process of identifying and merging records judged to represent the same real-world entity (Stanford InfoLab)
- Systematic and successive application of pair-wise resolution to a larger set of references

Entity-Relation Model (ERM)

- Foundation of modern data models
- **Entity Types** define objects that have
  - **Attributes**
  - Attributes have **values** that describe a particular **instance** of an entity type
- **Relations** define connections between entity types
- **Identity attributes** – attributes whose values distinguish one instance from another

Example


INSTRUCTOR
-FacultyID
-Name
-Department
COURSE
-CourseCode
-Time
-Location
STUDENT
-StudentID
-Name
-Major

Primary Key Problem

- Every table should have one
- Simplifies bringing together information about the same entity
  - **Table Join Operation**
- Problems
  - Different tables/databases often use different keys for same entity instance
  - Some records may not have keys
  - **Heterogeneous database join**

ER Principle #1

- IS store and manipulate **references** to entities, not the entities.
- Entities are real-world objects -- References are rows in a database table
  - In ER, instance of STUDENT entity type is a reference to a student -- the student is a person walking around campus
  - Data modelers call an instance an "entity", but in ER it's a reference

CDI

- When entity type is a customer, ER is called Customer Data Integration (CDI)

- Essential to support Customer Relationship Management (CRM)

Big ER – Five Activities

1. Entity Reference Extraction
2. Entity Reference Preparation
3. Entity Reference Resolution
4. Entity Identity Management
5. Entity Relationship Analysis

Entity Reference Extraction

- Identifying and extract entity reference from unstructured information
  - Free Text
  - Audio
  - Video
- Easy for people, hard for computers
- 80% of an organizations information is in unstructured text – reports, email, etc. – (Inmon, Nesavitch)

Entity Reference Preparation

- Where IQ meets ER
- References are often
  - Incomplete
  - Inaccurate
  - Inconsistently represented, etc.
- Degrade ER processes and outcomes
- Reference clean-up often consumes large portion of ER effort

Entity Reference Resolution

- Terminology : Linking vs. Matching
- Two references to the same entity are **equivalent** and should be **linked**
- **Matching** reference have the same (or mostly the same) identity attribute values
  - Matching records may not be equivalent
  - Equivalent records may not match
  - Mary Doe, Elm St – Mary Smith, Oak St
  - John Doe, Elm St – John Doe, Elm St

ER Principle #2

- ER is about linking equivalent references – matching is a means to an end
- Fundamental Law of ER
  Two entity references should be linked if and only if they reference the same entity (i.e. are equivalent).

False Negatives/Positives

- Two equivalent references that are not linked makes a **False Negative**
- Two non-equivalent references that are linked makes a **False Positive**
- Matching attribute values between two references is the most common (an intuitive) basis for making linkage decision, **but not the only one**

ER Principle #3

- False negative links are a more difficult problem to detect and solve in ER than false positive links
- Because ratio of true positives to true negatives is usually small –
  more non-links to checks for false,
  than links to check for false
- By definition, system doesn't give you something to look at

ER Principle #4

- ER processes are generally designed to favor false negatives over false positives
- In business applications - Impact of a false positive decision is considered higher than impact of false negative decision – In other applications may be different
- False negative decisions are easier to defend that false positive decisions

Identity Resolution

- Identity resolution is resolving an entity reference against a collection of known identities
- When known identities are for customers it is called **Customer Recognition**
- Identity resolution implies ER, but ER does not imply identity resolution

ER Principle #5

- Entity resolution is not the same as identity resolution
- Like fingerprints at a crimes scene
  - Can determine if two sets are for same or different suspects without knowing identity
  - Must get a "hit" against fingerprint database of known identities to identify
- Determining that references are to different entities without identifying them is called **disambiguation**

Entity Identity Management

- All ER systems use identity, but not all systems manage (store and update) identity information
- ER system that manage identity can append **persistent links** -- consistently assign references to the same entity the same link identifier over time
- Allows transactional ER processing
- Allows linking by association and assertion

ER Principle #6

- ER systems that provide persistent link values must also implement some form of identity management
- Identity resolution systems
- Identity capture systems
  - "smart" merge-purge

Four Methods for Linking

By

- Direct Matching
- Transitive Linking
- Linking by Association
- Asserted Linking

By Direct Matching

- Comparing the attributes between two references
- **Deterministic matching** – link if and only if all attributes agree
- **Probabilistic matching** – link if and only if certain combinations of attributes agree
- **Fuzzy matching** – "similar" attribute values can be counted as "agreeing"

By Transitive Linking

- Linking references through a chain of intermediate links
- If A links to B, B links to C,
  then A links to C
- Also called transitive closure
- Example: Probabilistic match on 2 out of 3 attributes
  - "Joe, GX, 56" matches "Joe, GX, 75"
  - "Joe, GX, 75" matches "Joe, TW, 75"
  - Link "Joe, GX, 56" and "Joe, TW, 75"

By Association

- Linking entity references based on relationships to other entities
- Example
  - Have an established link between "John Doe, Elm St" and "John Doe, Oak St"
  - Household association between "John Doe, Elm St" and "Sue Doe, Elm St"
  - Household association between "John Doe, Oak St" and "Sue Doe, Oak St"
  - Decision to link by association "Sue Doe, Elm St" and Sue Doe, Oak St"

By Assertion

- Linking references based on information from a reliable, external source – **knowledge-based linking**
- Example
  Magazine publisher reports that
  "Mary Doe, Oak St"
  is the same subscriber as
  "Mary Smith, Elm St"

Approximate (Fuzzy) Matching

- Approximate String Matching (ASM) is based on the similarity of two strings in terms of shared characters and character sequences (Syntax)
  - "KELLEY" and "KELLY" differ by 1 char
- Alias Matching is based on the similarity of two strings in terms of their meaning (Semantics)
  - "ED" and "EDWARD" differ by 4 chars, but one is a "nickname" for the other

ASM – Edit Distance

- Levenshtein Edit Distance
  - Minimum number of transformations needed to change one string into another (delete, insert, replace)
  - "SALLIE" to "SALLY" distance = 2
  - Usually normalized by length of longest string, e.g. (6-2)/6 = 4/6 = 0.667
  - Does not consider phonetic similarity
  - Does not consider position of difference
    "THOMPSON" to "THOMAS" = 3
    "THOMPSON" to "COMPTON" = 3

ASM - Soundex

- Capitalize all letters, drop punctuation
- Remove 'A', E', 'I', 'O', 'U', 'H', 'W', 'Y' after the first letter
- Change letters to digits as
  - 1 = 'B', 'F', 'P', 'V'
  - 2 = 'C', 'G', 'J', 'K', 'Q', 'S', 'X', 'Z'
  - 3 = 'D','T'
  - 4 = 'L'
  - 5 = 'M','N'
  - 6 = 'R'

Soundex (Continued)

- Replace consecutive sequences of same digit with a single digit if original letters were adjacent or separated by “H” or “W”
- Truncate or pad with zeros to make a total of 4 characters
- Example:
  - PHILLIP – PLLP – P441 – P41 - P410
  - PHILIP – PLP – P41 – P410
  - PETERSON – PTRSN – P3625 – P362

Soundex Examples

- LEE -> L000 (both "E"s are dropped)
- SHAW -> S000 ("H","A","W" in drop list)
- GAUSS->GSS->G22->G2->G200
- CHERRY->CRR->C66->C6->C600
- CHECKER->CCKR->C226->C26->C260
- COUSSACSK->C22 222 ->C22->C220

Soundex Anomalies

- Group 1
  - LEE -> L000
  - LEIGH -> L200
  - LIU -> L000
- Group 2
  - GAUSS & GHOSH -> G200
  - WACHS & WAUGH -> W200
- Other issues
  - Lloyd, van Buren, von Munching

ASM - Jaro String Comparator

- Accounts for
  - Difference in length
  - Transposition of characters
    "JHON" vs "JOHN"
  - Number of characters in common
- Let s_1 and s_2 be strings
  - If index of char x is n_1 in s_1
  - If index of char x is n_2 in s_2
  - If $|n_1 - n_2| \leq \min\{|s_1|, |s_1|\}/2$
  - Then x is counted as a common char

Jaro Formula

If c >0 then

$$\Phi(s_1, s_2) = W_1 \cdot \frac{c}{L_1} + W_2 \cdot \frac{c}{L_2} + W_3 \cdot \frac{(c - \tau)}{c}$$

W_1 = Weight assigned to first string
W_2 = Weight assigned to second string
W_3 = Weight assigned to transpositions
$W_1+W_2+W_3 = 1$
c = common character count
L_1 = Length of first string
L_2 = Length of second string
τ = Number of chars transposed

If c = 0 then $\Phi(s_1, s_2) = 0$

Example 1

- Higbee – Higvee
- $L_1 = L_2 = 6$, $c = 5$, $\tau = 0$, $W_1 = W_2 = W_3 = 1/3$

$$\Phi_J(s_1, s_2) = W_1 \cdot \frac{c}{L_1} + W_2 \cdot \frac{c}{L_2} + W_3 \cdot \frac{(c-\tau)}{c}$$

$$= \left(\frac{1}{3}\right) \cdot \left(\frac{5}{6}\right) + \left(\frac{1}{3}\right) \cdot \left(\frac{5}{6}\right) + \left(\frac{1}{3}\right) \cdot \left(\frac{5-0}{5}\right)$$

$= 8/9 = 0.889$

Example 2

- Shackleford– Shackelford
- $L_1 = L_2 = 11$, $c = 11$, $\tau = 2$, $W_1 = W_2 = W_3 = 1/3$

$$\Phi_J(s_1, s_2) = W_1 \cdot \frac{c}{L_1} + W_2 \cdot \frac{c}{L_2} + W_3 \cdot \frac{(c-\tau)}{c}$$

$$= \left(\frac{1}{3}\right)\cdot\left(\frac{11}{11}\right) + \left(\frac{1}{3}\right)\cdot\left(\frac{11}{11}\right) + \left(\frac{1}{3}\right)\cdot\left(\frac{11-2}{11}\right)$$

$= 31/33 = 0.939$

ASM- Winkler String Comparator

- Modification of the Jaro Comparator
- Gives higher weight to agreement of initial characters of strings

$$\Phi_W(s_1, s_2) = \Phi_J(s_1, s_2) + i \cdot 0.1 \cdot (1 - \Phi_J(s_1, s_2))$$

- Where
  - $i = \min\{j, 4\}$
  - j = number of initial chars in common
- Example Shackleford – Shackelford
- = 0.939+4*0.1*(0.061) = 0.963

Other ASM

- n-grams (q-grams) based on number of shared substrings of length n
- LCS - longest common substring
- Variations of Soundex
  - NYSIIS - New York State Identification and Intelligence System – avoids first letter problem
  - Phonex – preprocess names before using Soundex
  - Phonix – an improved version of Phonex

ER MODELS

Fellegi-Sunter Model

- Standard for probabilistic matching
- Context
  - Two unduplicated lists of references A, B
  - Both lists have N corresponding identity attributes
- Given a false positive rate P and false negative rate N, the model defines a linking strategy that will
  - Not exceed P and N,
  - Minimize cases requiring intervention

Fellegi-Sunter Conditions

- A and B two lists of references
- Consider A x B (all pairs)
- M = True positives, i.e. $(a, b) \in M$ if and only if "a" should be linked to "b"
- U = True negatives, i.e. $(a, b) \in U$ if and only if "a" should NOT be linked to "b"
- Γ = all attribute match/no-match combinations of the N attributes. There will be 2^N of these.

Fellegi-Sunter Weight Ratios

- For an agreement pattern $\gamma \in \Gamma$ define

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

- Represents the ratio of the "probability of Good Links" to "probability of Bad Links" for a given match pattern
- Very large value means good link rule
- Very small value means bad link rule

Fellegi-Sunter (cont)

- Establish two values U (upper) and L (lower) in the series of decreasing values of R

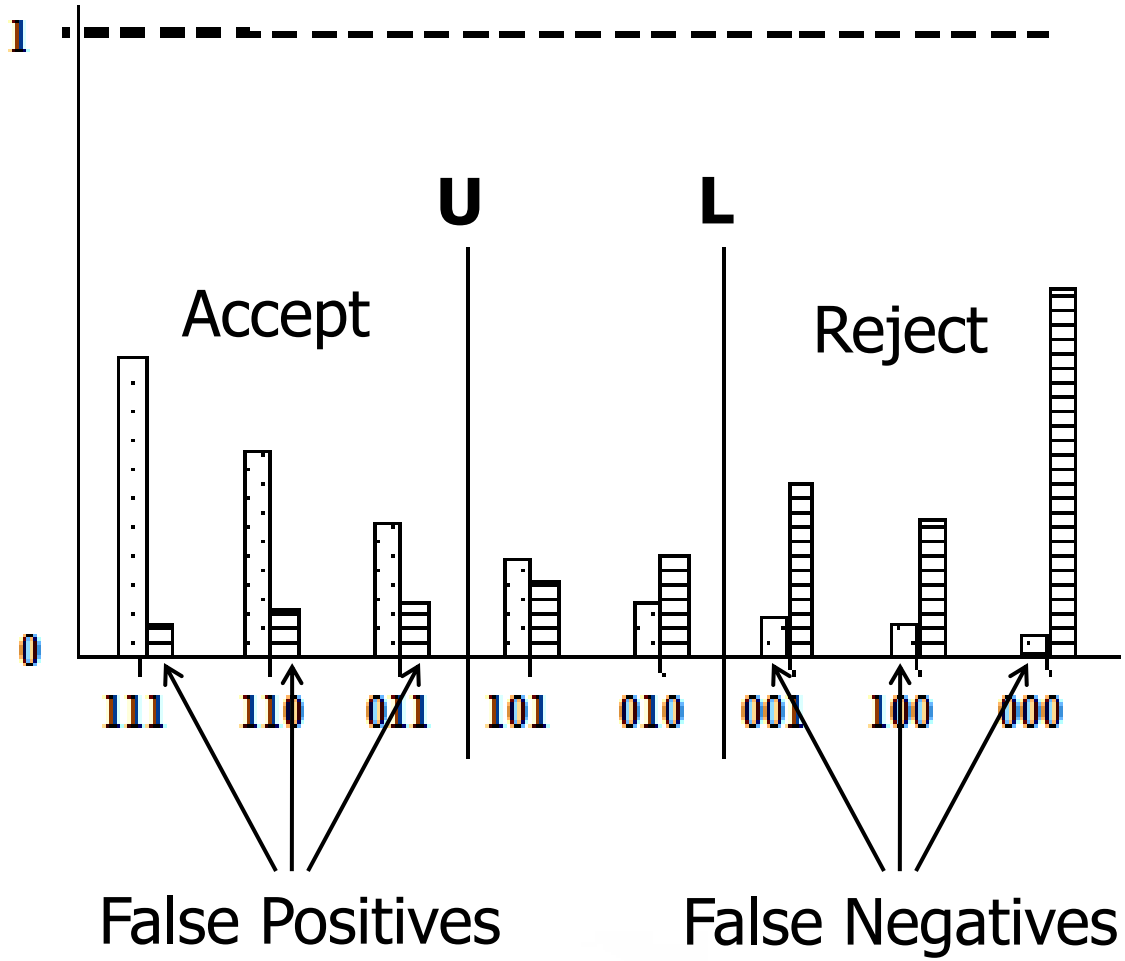
$$\underbrace{R_1 > R_2 > \ldots >}_{\text{Accept}} \mathbf{U} > \underbrace{\ldots R_n > R_{n+1} > \ldots >}_{\text{Review}} \mathbf{L} > \underbrace{\ldots R_m > R_{m+1}}_{\text{Reject}}$$

- U and L are selected so that N and P (respectively) are not exceeded

Example: Student Records

- Two enrollments list from consecutive years
- Match first name, last name, DOB
- Expect large overlap, but
  - Some first year students leave
  - Some new students second year
- Not all records have DOB
- Use 3-bit binary numbers to represent agreement patterns

True and False Positives



Stanford SERF Model

- Developed at Stanford InfoLab
- **S**tanford **E**ntity **R**esolution **F**ramework
- Intended to be a "generic" ER Model
- Fellegi-Sunter gives a way to evaluate matching, SERF does not
- SERF does describe
  - Conditions that must hold for ER outcome to be unique
  - How pair-wise matching can resolve a set (merge-purge algorithm)

Match and Merge Functions

- R is set of references
- Two functions defined

  Match Function M

  – **M: D x D → {true, false}**

  – **R ⊂ D the domain of M**

  Merge Function μ

  – **If a, b ∈ D, M(a, b) = true, then μ(a, b)∈D**

  Definition

  – **If μ(a, b) = a, then "a dominates b"**

SERF definition of ER

ER(R) $\subseteq$ D such that

- Any record that can be derived from R is either in ER(R) or is dominated by a record in ER(R)
- No two records in ER(R) match and no record in ER(R) is dominated by any other

- Think of merged records in ER(R) as clusters of equivalent records

Consistent ER

- Consistent ER means that ER(R) exists, is finite, and is unique
- Will be consistent if the following condition hold
  - **M(a, b) = M(b, a) & μ(a, b) = μ(a, b)**
  - **M(a, a) = true & μ(a, a) = a**
  - **M(a, μ(a, b)) = M(b, μ(a, b)) = true**
  - **μ(a, μ(b, c)) = μ(μ(a, b), c)**

R-Swoosh Algorithm

- Systematic way to find ER(R) if match & merge functions are consistent

1. Start: D = R, and ER(R) =Ø
2. Start comparing first record **x** in D to each record **y** in ER(R)
3. If M(**x**, **y**) = true
   - Stop comparing
   - Replace x in D with μ(**x**, **y**)
   - Remove y from ER(R)

R-Swoosh Algorithm (cont)

4. If M(**x**, **y**) not true for any **y** in ER(R)
   - Put **x** in ER(R)
   - Remove **x** from D
5. If more items in D to process, go back to Step 3,
   otherwise algorithm is finished

Example: D at Start of Process

	First	Last	DOB	SCode
r1	Edgar	Jones	20001104	G34
r2	Mary	Smith	19990921	G55
r3	Eddie	Jones	20001104	G34
r4	Mary	Smith	19990921	H17
r5	Eddie	Jones	20001104	H15

- Match if references agree on
  - First, Last, DOB, or Last, DOB, SCode
- Merge combines attributes

Example: ER(R) at End

	First	Last	DOB	SCode
r7	Mary	Jones	20001104	{H17,G55}
r8	{Eddie, Edgar}	Jones	20001104	{G34, H15}

- r7 represents original r1, r3, r5
- r8 represents original r2 and r4

Algebraic Model (Background)

Definitions

- Given a set S and a subset $T \subseteq SxS$, then T is said to be a relation on S
- T is said to be an equivalence relation on S if and only if
  - For every $a \in S$, then $(a, a) \in T$
  - If $(a, b) \in T$, then $(b, a) \in T$
  - If $(a, b) \in T$ and $(b, c) \in T$, then $(a, c) \in T$

Background Continued

- If T is an equivalence relation on S then $[a] = \{b \in S \mid (b, a) \in T\}$ is the <u>equivalence class of a</u>
- A partition P of a set S is a collection of subsets $P_1, P_2, \ldots P_n$ such that
  - $P_j \neq \emptyset$ for all j=1…n
  - $P_j \cap P_k = \emptyset$ whenever $j \neq k$
  - $S = P_1 \cup P_2 \cup \ldots \cup P_n$
- If T is an equivalence relation on S then $P = \{[a] \mid a \in S\}$ is a partition of S.

Algebraic Model Defined

- Defines ER only in terms of outcome
  - Let R be a set of references where every $a \in R$ references one and only one real-world object
  - Define $E \subseteq R \times R$ by $(a, b) \in E$ if and only if a and b reference the same real-world object.
- Then
  - E is an equivalence relation on R
  - The equivalence classes of E define a unique partition of R

From Previous Example

- R = {r1, r2, r3, r4, r5}, then
- E = {(r1, r1), (r2, r2), (r3, r3), (r4, r4), (r5, r5), (r1, r3), (r3, r1), (r1, r5), (r5, r1), (r3, r5), (r5, r3), (r2, r4), (r4, r2)}
- Partition defined by E is P(E) = {{r1, r3, r5}, {r2, r4}}

Comparing ER Outcomes

- Comparing ER outcomes is same as comparing partitions
- Let P and Q be two partitions of S
- Define $V = \{P_j \cap Q_k \mid P_j \cap Q_k \neq \emptyset\}$
- The Talburt-Wang Similarity Index (TWI) is defined by

$$TWI = \frac{\sqrt{|P| \cdot |Q|}}{V}$$

- TWI is a number from 0 to 1
- TWI = 1 iff P = Q

Example

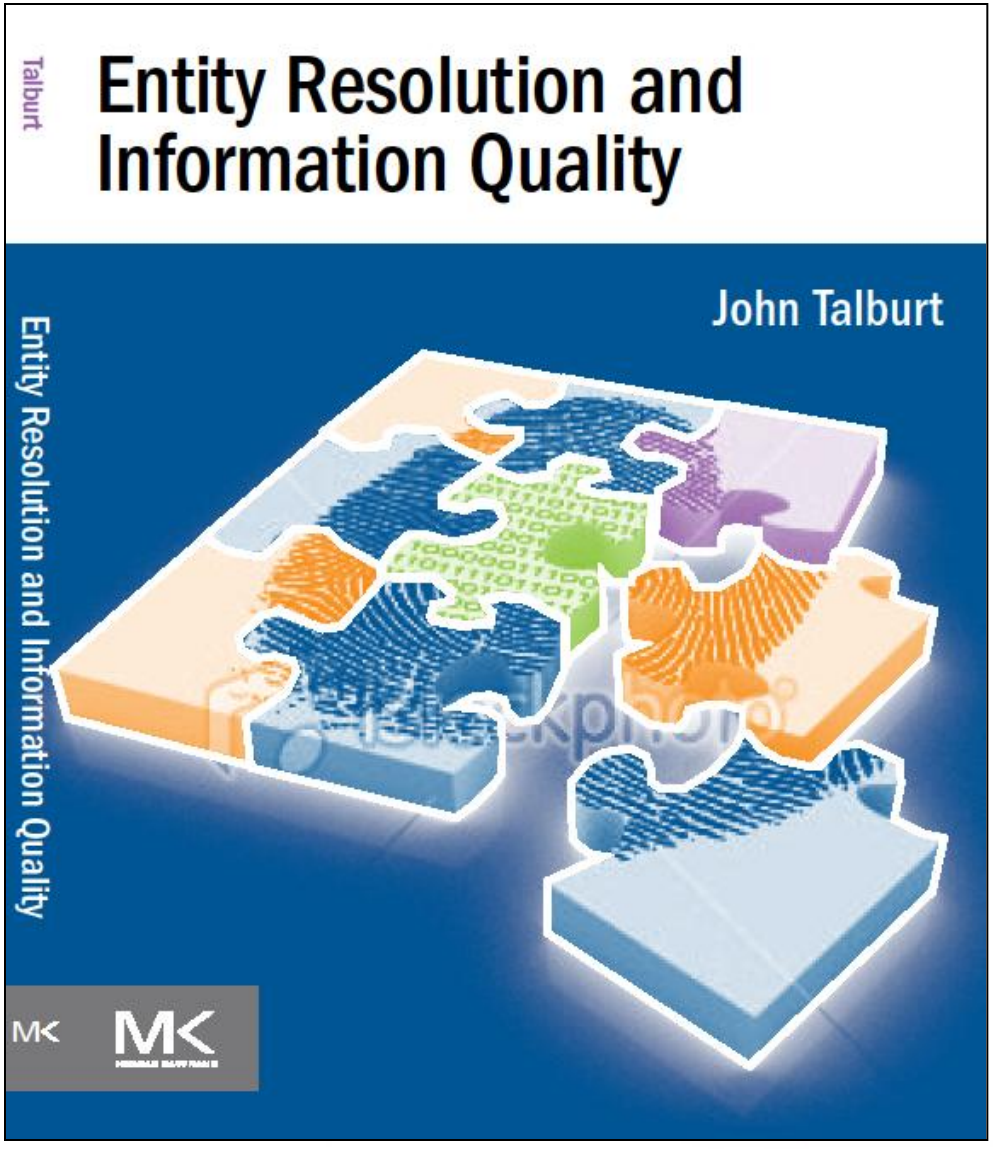
- S = {a, b, c, d, e, f, g, h}
- P = {{a, d, e}, {b}, {c, f, g}, {h}}
- Q = {{a, b, d}, {e}, {c, f}, {g}, {h}}
- V = {{a, d}, {e}, {b}, {c, f}, {g}, {h}}
- |P| = 4, |Q| = 5, |V| = 6
- TWI = SQRT(4 x 5)/6 = SQRT(20)/6 = 0.745

Areas of Research

- Integration of IQ and ER
- Entity Reference Extraction from unstructured information
  - Named Entity Recognition (NER)
- Entity Association Analysis – Entity Analytics
  - Fraud, Law Enforcement
- Application of High-Performance Computing (HPC) to ER

Coming Soon from ERIQ Lab

- OYSTER Open-Source Entity Resolution System
  - Java
  - XML
- Entity Resolution and Information Quality
  - Morgan Kaufman Publishing
  - November 2010


Talburt
Entity Resolution and Information Quality
John Talburt
Entity Resolution and Information Quality
MK
MK

Questions and Discussion

John R. Talburt
jrtalburt@ualr.edu

ualr.edu/eriq

UALR | University of Arkansas at Little Rock