Entity Resolution in SERF

Omar Benjelloun
Stanford University

Joint work with:
Hector Garcia-Molina, Hideki Kawai, Tait E. Larson,
David Menestrina, Qi Su, Sutthipong Thavisomboon,
Jennifer Widom
Entity Resolution (ER)

• Many applications:
  – customer files,
  – counter-terrorism,
  – comparison shopping...

• Aka: deduplication, record linkage, object co-identification, reference reconciliation, …
Challenges (1)

• No keys!
• Value matching
  – “Kaddafī”, “Qaddafī”, “Kadafī”, “Kaddaffī”...
  – Many techniques developed
• Record matching

Nm: Tom
  Ad: 123 Main St
  Ph: (650) 555-1212
  Ph: (650) 777-7777

Nm: Thomas
  Ad: 132 Main St
  Ph: (650) 555-1212
Challenges (2)

• Merging records

Nm: Tom
  Ad: 123 Main St
  Ph: (650) 555-1212
  Ph: (650) 777-7777

Nm: Thomas
  Ad: 132 Main St
  Ph: (650) 555-1212
  Zp: 94305

Nm: Tom
  Nm: Thomas
    Ad: 123 Main St
    Ph: (650) 555-1212
    Ph: (650) 777-7777
    Zp: 94305
Challenges (3)

• Chaining

Nm: Tom
Ad: 123 Main
BD: Jan 1, 85
Wk: IBM
Nm: Thomas
Ad: 132 Main
Oc: lawyer
Nm: Tom
Wk: IBM
Oc: lawyer
Sal: 500K
Nm: Tom
Ad: 123 Main
BD: Jan 1, 85
Wk: IBM
Oc: lawyer
Sal: 500K
Generic Entity Resolution

- Set of records: $R$ (from domain $\mathcal{R}$)
- Match function: $\mathcal{R} \times \mathcal{R} \rightarrow \text{Boolean}$
  - $M(r_1, r_2) = \text{true}$ if $r_1, r_2$ represent the same entity
- Merge function: $\mathcal{R} \times \mathcal{R} \rightarrow \mathcal{R}$
  - $r_3 = <r_1, r_2>$ (exists if $M(r_1, r_2) = \text{true}$)

- We view match and merge as black boxes
- Focus on performance rather than accuracy
Domination

- Some records are less informative than others

![](tree.png)

- Record r1 is **dominated** by record r2 if \(<r1,r2>=r2\)
- Dominated records should be discarded
The Entity Resolution problem

• Given a set of records R, the Entity Resolution of R:
  – Has only records derived from R
  – Dominates all records derivable from R
  – Contains no matching or dominated records

• We provide simple and natural conditions to
  – Make ER “consistent” (finite and unique)
  – Enable efficient computation strategies
Conditions

• Commutativity:
  - $M(r_1, r_2) = M(r_2, r_1)$
  - $<r_1, r_2> = <r_2, r_1>$

• Idempotence:
  - $M(r_1, r_1) = true; <r_1, r_1> = r_1$

• Merge associativity:
  - $<r_1, <r_2, r_3>> = <<r_1, r_2>, r_3>$ (if they exist)
Conditions (2)

• Representativity
  
  - $r_3 = \langle r_1, r_2 \rangle$
    for any $r_4$ such that $M(r_1, r_4) = \text{true}$
    we also have $M(r_3, r_4) = \text{true}$.
Algorithms

• These conditions enable flexible computation of $ER(R)$
  – Starting from $R$…
  – Find matches, add merged records
  – Find and delete dominated records
  – …in any order

• Optimal algorithm: R-Swoosh
  – Merges records and deletes dominated records early
  – No algorithm performs fewer record comparisons in the worst case
R-Swoosh

<table>
<thead>
<tr>
<th>R</th>
<th>R'</th>
</tr>
</thead>
<tbody>
<tr>
<td>r1</td>
<td></td>
</tr>
<tr>
<td>r2</td>
<td></td>
</tr>
<tr>
<td>r3</td>
<td></td>
</tr>
<tr>
<td>r4</td>
<td></td>
</tr>
<tr>
<td>r5</td>
<td></td>
</tr>
<tr>
<td>r6</td>
<td></td>
</tr>
</tbody>
</table>
R-Swoosh

<table>
<thead>
<tr>
<th>R</th>
<th>R’</th>
</tr>
</thead>
<tbody>
<tr>
<td>M(r3,r1)</td>
<td>r2</td>
</tr>
<tr>
<td>M(r4,r2)</td>
<td></td>
</tr>
<tr>
<td>r7 = &lt;r4,r1&gt;</td>
<td>r3</td>
</tr>
<tr>
<td>r5</td>
<td></td>
</tr>
<tr>
<td>r6</td>
<td></td>
</tr>
<tr>
<td>r7</td>
<td></td>
</tr>
</tbody>
</table>
Also F-Swoosh, a variant that efficiently caches results of value comparisons.
Example

- \([a: v1, b: w1]\)
- \([a: v2, b: w2]\)
- \([a: v3, b: w3]\)
- ...
- \([a: vn, b: wn]\)

Match: \(M( ri, rj ) = True\)

Merge: Union of values

answer: \([a:\{v1, ..., vn\}, b:\{w1, ..., wn\}]\)
Naïve strategy

- [a: v1, b: w1]
- [a: v2, b: w2]
- [a: v3, b: w3]
- [a: v4, b: w4]

- [a: {v1,v2}, b: {w1,w2}]
- [a: {v1,v3}, b: {w1,w3}]
- [a: {v1,v4}, b: {w1,w4}]
- [a: {v2,v3}, b: {w2,w3}]
- [a: {v2,v4}, b: {w2,w4}]
- [a: {v3,v4}, b: {w3,w4}]
Naïve strategy (2)

- [a: {v1,v2}, ...]
- [a: {v1,v3}, ...]
- [a: {v1,v4}, ...]
- [a: {v2,v3}, ...]
- [a: {v2,v4}, ...]
- [a: {v3,v4}, ...]
- [a: {v1,v2,v3}, ...]
- [a: {v1,v2,v4}, ...]
- [a: {v2,v3,v4}, ...]
- [a: {v1,v2,v4}, ...]
- [a: {v1,v2,v3,v4}, ...]

... A lot of useless work!
R-Swoosh

- [a: v1, b: w1]
- [a: v2, b: w2]
- [a: v3, b: w3]
- [a: v4, b: w4]

- \( M(r_1, r_2) \) \( \text{®} \)
  
  - \([a: \{v_1, v_2\}, \ldots]\)

- \( M(r_3, r_{12}) \) \( \text{®} \)
  
  - \([a: \{v_1, v_2, v_3\}, \ldots]\)

- \( M(r_4, r_{123}) \) \( \text{®} \)
  
  - \([a: v_1, a: v_2, a: v_3, a: v_4, \ldots]\)
Distributed ER

• ER is expensive:
  – Many records
  – Match comparisons are costly

• Distribute the work across multiple processors
  – Make sure no matches are missed
  – Minimize computation, communications and storage

• Use domain knowledge when available
  – E.g., DOB within 5 years, same product category
D-Swoosh

Processor Pi

M(r3,r4)?
r6 = <r3,r4>

add(r6)
del(r3)
del(r6)
ad(r9)
del(r4)

r6 = <r3,r4>
D-Swoosh

- Where to send records?
  - scope function (e.g., \( \text{scope}(r2) = \{P2, P5, P7\} \))

- Who is responsible for comparisons?
  - resp predicate (e.g., \( \text{resp}(P6, r3, r5) = \text{true} \))

- scope and resp must satisfy coverage property (related to mutual exclusion problem -- coteries)

- Schemes without domain knowledge
  - Majority, grid

- Schemes with domain knowledge
  - Value equality, linear ordering, hierarchies
D-Swoosh performance

- Computation cost per processor (10 processors)
- Experiments on Yahoo! comparison shopping data
ER with confidences

• Each record has a “confidence” \(0 \leq c \leq 1\)
  – Not tied to specific interpretation (e.g., probabilistic)
  – Match function may exploit confidences
  – Merge function propagates confidences

• Some conditions do not hold anymore:
  – Representativity: Confidence decreases with merges
  – Associativity: Different derivations produce different confidences

• More costly algorithm is required (Koosh)
  – Optimizations: early detection of domination, thresholds
Summary

- Entity resolution is critical
- Generic approach yields reusable techniques
- Efficient resolution is important
- Currently working on
  - Large scale distributed ER
  - Negative information
  - Uncertainty and lineage in ER
Thank you.