Automatic training example selection for scalable unsupervised record linkage

Peter Christen

Department of Computer Science,
The Australian National University,
Canberra, Australia

Contact: peter.christen@anu.edu.au


Funded by the Australian National University, the NSW Department of Health,
and the Australian Research Council (ARC) under Linkage Project 0453463.

What is record (or data) linkage?

- The process of linking and aggregating records from one or more data sources representing the same entity (such as a patient, customer, or business)
- Also called data matching, data scrubbing, entity resolution, object identification, merge-purge, etc.
- Challenging if no unique entity identifiers available
  For example, which of these three records refer to the same person?

<table>
<thead>
<tr>
<th>Dr Smith, Peter</th>
<th>42 Miller Street 2602 O'Connor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pete Smith</td>
<td>42 Miller St, 2600 Canberra A.C.T.</td>
</tr>
<tr>
<td>P. Smithers</td>
<td>24 Mill Street; Canberra ACT 2600</td>
</tr>
</tbody>
</table>

Outline

- What is record linkage?
- Record linkage challenges
- The record linkage process
- Record pair comparison and classification
- Two-step record pair classification
  - Step 1: Training example selection
  - Step 2: Classification of record pairs
- Experimental results
- Outlook and future work

Record linkage challenges

- Real world data is dirty
  (typographical errors and variations, missing and out-of-date values, different coding schemes, etc.)
- Scalability
  - Naïve comparison of all record pairs is $O(n^2)$
  - Some form of blocking, indexing or filtering required
- No training data in many linkage applications
  - No data sets with known true match status
  - Possible to manually prepare training data (but, how accurate will manual classification be?)
**The record linkage process**

1. **Database A**
   - Cleaning and standardisation
   - Blocking / Indexing

2. **Database B**
   - Cleaning and standardisation

3. **Weight vector classification**
   - Matches
   - Non-matches
   - Possible matches

4. **Field comparison**

5. **Clerical review**

6. **Evaluation**

**Record pair comparison**

- Pairs of records are compared field (attribute) wise using different field comparison functions
  - Such as exact or approximate string (e.g. edit-distance, q-gram, Winkler), numeric, age, date, time, etc.
  - Return 1.0 for exact similarity, 0.0 for total dissimilarity
- For each compared record pair a **weight vector** containing matching weights is calculated
  - Record 1: ['dr', 'peter', 'paul', 'miller']
  - Record 2: ['mr', 'john', '', 'miller']
  - Matching weights: [0.5, 0.0, 0.0, 1.0]
- Weight vectors (record pairs) are classified into matches, non-matches (and possible matches)

**Record pair classification**

- Traditionally, matching weights are summed, and two thresholds are used for classification
- Various machine learning techniques have been investigated
  - Supervised: SVM, decision trees, neural networks, learnable string comparisons, active learning, etc.
  - Un-supervised: Different clustering algorithms
- Recently, collective entity resolution techniques have been investigated
  - Rather than classifying each record pair independently
  - Using relational attributes (i.e. graph based)
  - However, not all data is relational

**Two-step record pair classification**

- Assumptions
  - Weight vectors that have exact or high similarity values in all elements were most likely generated when two records were compared that refer to the same entity
  - Weight vectors with mostly low similarity values were with high likelihood generated when two records were compared that refer to different entities
- Idea: **Automatically select such weight vectors as training examples in a first step, and then use them to train a binary classifier in a second step**
  - Combined, this will allow fully automated unsupervised record pair classification
**Records and weight vectors example**

<table>
<thead>
<tr>
<th>R1:</th>
<th>Christine</th>
<th>Smith</th>
<th>42</th>
<th>Main</th>
<th>Street</th>
</tr>
</thead>
<tbody>
<tr>
<td>R2:</td>
<td>Christina</td>
<td>Smith</td>
<td>42</td>
<td>Main</td>
<td>St</td>
</tr>
<tr>
<td>R3:</td>
<td>Bob</td>
<td>O’Brian</td>
<td>11</td>
<td>Smith</td>
<td>Rd</td>
</tr>
<tr>
<td>R4:</td>
<td>Robert</td>
<td>Bryce</td>
<td>12</td>
<td>Smythe</td>
<td>Road</td>
</tr>
</tbody>
</table>

\[
\begin{array}{c|c|c|c|c|c}
WV(R1,R2): & 0.9 & 1.0 & 1.0 & 1.0 & 0.9 \\
WV(R1,R3): & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
WV(R1,R4): & 0.0 & 0.0 & 0.5 & 0.0 & 0.0 \\
WV(R2,R3): & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
WV(R2,R4): & 0.0 & 0.0 & 0.5 & 0.0 & 0.0 \\
WV(R3,R4): & 0.7 & 0.3 & 0.5 & 0.7 & 0.9 \\
\end{array}
\]

**Step 1: Training example selection**

- Weight vectors can be selected using either thresholds or nearest based.
- Training examples are likely linearly separable.
- Idea: randomly add more training examples (from gap between match and non-match examples).

**Step 2: Classification of record pairs**

- Any binary classifier can be used (in the following experiments, a linear SVM has been employed).
- Question investigated here: Does the random inclusion of additional weight vectors improve classification accuracy?

- Related work: Similar approaches have been developed for text and Web page classification.
  - Called semi-supervised or partially supervised learning.
  - PEBL (positive example based learning): train a SVM only on positive labeled examples, improve iteratively.
  - S-EM (seed expectation-maximisation): add ‘spy’ documents from positive examples into unlabeled data.

**Experimental evaluation**

- All techniques are implemented in the Febrl open source record linkage system (available from: [https://sourceforge.net/projects/febrl/](https://sourceforge.net/projects/febrl/)).
- Experiments using both real and synthetic data (Secondstring repository and Febrl data set generator).
- Evaluation of step 1 (training example selection).
  - Percentage of true matches and true non-matches in the training example sets.
- Evaluation of step 2 (record pair classification).
  - F-measure (harmonic mean of precision and recall) (average and standard-deviation are shown in graphs).
**Quality of weight vectors selected**

<table>
<thead>
<tr>
<th>Data sets</th>
<th>Thresholds</th>
<th>Nearest</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.3</td>
<td>0.5</td>
</tr>
<tr>
<td>Census</td>
<td>100/-96.2/100</td>
<td>1% 10%</td>
</tr>
<tr>
<td>Restaurant</td>
<td>98.5/-4.5/100</td>
<td>100/100 58.6/100</td>
</tr>
<tr>
<td>Gen-1,000</td>
<td>100/100</td>
<td>100/100</td>
</tr>
<tr>
<td>Gen-2,500</td>
<td>100/100</td>
<td>100/99.5</td>
</tr>
<tr>
<td>Gen-5,000</td>
<td>100/100</td>
<td>100/99.0</td>
</tr>
<tr>
<td>Gen-10,000</td>
<td>100/99.7</td>
<td>100/99.6</td>
</tr>
</tbody>
</table>

- Results given here are percentage values for match/non-match sets

**Record pair classification for Census**

'Census' data set (449 + 392 records, 2093 weight vectors)

**Record pair classification for Gen-10,000**

'Gen-10,000' data set (10,000 records, 132,532 weight vectors)

**Outlook and future work**

- The proposed two-step record pair classification approach shows promising results
  - Can automatically select good quality training examples
  - Random inclusion of additional weight vectors does not improve classification accuracy (unlike improvements in Web and text classification)
- Improvements for second step (classification)
  - Apply classifier iteratively (as done in PEBL approach)
  - Investigate nearest-neighbour based classification
- More experiments on different data are needed
  - Also investigate the scalability of this approach