A Heterogeneous Field Matching Method for Record Linkage

Steven Minton and Claude Nanjo
Fetch Technologies
{sminton, cnanjo}@fetch.com

Craig A. Knoblock, Martin Michalowski, and Matthew Michelson
USC / ISI
{knoblock,martinm,michelso}@isi.edu
Introduction

- Record linkage is the process of recognizing when two database records are referring to the same entity.
  - Employs similarity metrics that compare pairs of field values.
  - Given field-level similarity, an overall record-level judgment is made.
### Record Linkage

An example

<table>
<thead>
<tr>
<th>Company</th>
<th>Address</th>
<th>Industry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Union Switch and Signal</td>
<td>2022 Hampton Ave</td>
<td>Manufacturing</td>
</tr>
<tr>
<td>JPM</td>
<td>115 Main St</td>
<td>Manufacturing</td>
</tr>
<tr>
<td>McDonald’s</td>
<td>Corner of 5&lt;sup&gt;th&lt;/sup&gt; and Main</td>
<td>Food Retail</td>
</tr>
<tr>
<td>Joint Pipe Manufacturers</td>
<td>115 Main Street</td>
<td>Plumbing Manufacturer</td>
</tr>
<tr>
<td>Union Sign</td>
<td>300 Hampton Ave</td>
<td>Signage</td>
</tr>
<tr>
<td>McDonald’s Restaurant</td>
<td>532 West Main St.</td>
<td>Restaurant</td>
</tr>
</tbody>
</table>
Traditional Approaches to Field Matching

Rule Based Approach:

- **Pros:**
  - Highly tailored domain-specific rules for each field
    - E.g., last_name > first_name
  - Leverages domain-specific information.

- **Cons:**
  - Not Scalable
  - Rarely reusable on other domains
Traditional Approaches to Field Matching

Previous Machine Learning Approaches:

- **Pros**
  - Sophisticated decision-making methods at record level (e.g. DT, SVM, etc...)
  - Field matching often generic (TFIDF, Levenshtein)
  - Hence, more scalable

- **Cons**
  - Often used only one such homogeneous field matching approach
    - Thus, unable to detect heterogeneous relationships within fields (e.g. acronyms and abbreviations)
  - Failed to capture some important domain-specific fine-grained phenomena
Introducing the Hybrid Field Matcher (HFM)

(Based on Sheila Tejada’s Active Atlas platform)

Rule Based

- Library of ‘heterogeneous’ transformations that capture complex relationships between fields

Machine Learning

- Customizable transformations using ML

Hybrid Field Matcher

Better field matching results in better record linkage
Field Matching: Our Goals

- To identify important relationships between tokens
- To capture these relationships using an expressive library of ‘transformations’.
- To make these transformations generalizable across domain types.
- To translate the knowledge imparted from their application into a field score.
Field Matching

“JPM” ~ “Joint Pipe Manufacturers” → Acronym
“Hatchback” ~ “Liftback” → Synonym
“Miinton” ~ “Minton” → Spelling mistake
“S. Minton” ~ “Steven Minton” → Initials
“Blvd” ~ “Boulevard” → Abbreviation
“200ZX” ~ “200 ZX” → Concatenation
Map attribute(s) from one datasource to attribute(s) from the other datasource.

Tokenize, then label tokens

Eliminate highly unlikely candidate record pairs.

Use learned distance metric to score field—primary contribution

Pass feature vector to SVM classifier to get overall score for candidate pair.

define schema alignment

Parsing

blocking

field-to-field comparison

SVM – determine match
HFM Overview
Parsing and tagging

Raul De la Torre

Raoul Delatorre

given_name  Raul

De

Ia

Torre
HFM Overview

Blocking

- Provide the best set of candidate record pairs to consider for record linkage
- Blocking step should not affect recall by eliminating good matches
- We used a reverse index
  - datasource 1 used to build index
  - datasource 2 used to do lookup
HFM Overview
Field to Field Comparison

Name Field a

given_name  Raul
surname     De
surname     la
surname     Torre

Name Field b

given_name  Raoul
surname     Delatorre

Score = 0.98
### HFM Overview

#### SVM Classification

<table>
<thead>
<tr>
<th></th>
<th>Record 1</th>
<th>Record 2</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Raoul DelaTorre</td>
<td>Raul De la Torre</td>
<td>0.98</td>
</tr>
<tr>
<td>Gender</td>
<td>Male</td>
<td>M</td>
<td>0.99</td>
</tr>
<tr>
<td>Age</td>
<td>35</td>
<td>36</td>
<td>0.79</td>
</tr>
</tbody>
</table>

Score for candidate pair: 0.975
Training the Field Learner

Transformations =

{ Equal, Synonym, Misspelling, Abbreviation, Prefix, Acronym, Concatenation, Suffix, Soundex, Missing… }

Transformation Graph

“Intl. Animal” \leftrightarrow “International Animal Productions”

Intl. \quad \text{abbreviation} \quad \text{International}

Animal \quad \text{equals} \quad \text{Animal}

missing \quad \text{Productions}
Training the Field Learner

Another Transformation Graph

“Apartment 16 B, 3101 Eades St” $\leftrightarrow$ “3101 Eads Street NW Apt 16B”
Training the Field Learner
Step 1: Tallying transformation frequencies

Generic Preference Ordering
Equal > Synonym > Misspelling > Missing …

Training Algorithm:
I. For each training record pair
   i. For each aligned field pair (a, b)
      i. build transformation graph T(a, b)
         ➢ “complete / consistent”
         ➢ Greedy approach: preference ordering over transformations
Training the Field Learner
Step 2: Calculating the probabilities

- For each transformation type $v_i$ (e.g. Synonym), calculate the following two probabilities:

  $p(v_i|\text{Match}) = p(v_i|M) = \frac{\text{freq. of } v_i \text{ in } M}{\text{size } M}$

  $p(v_i|\text{Non-Match}) = p(v_i|\neg M) = \frac{\text{freq. of } v_i \text{ in } \neg M}{\text{size } \neg M}$

- Note: Here we make the Naïve Bayes assumption
Scoring unseen instances

Naïve Bayes assumption

\[
p(M \mid v_1, v_2, \ldots, v_n) = \frac{p(M) \prod_{i=1}^{n} p(v_i \mid M)}{\prod_{i=1}^{n} p(v_i)}
\]

\[
Score_{HFM} = \frac{p(M \mid V)}{p(M \mid V) + p(-M \mid V)} = \frac{p(M) \prod_{i=1}^{n} p(v_i \mid M)}{p(M) \prod_{i=1}^{n} p(v_i \mid M) + p(-M) \prod_{i=1}^{n} p(v_i \mid -M)}
\]
Scoring unseen instances

An Example

\[ a = "\text{Giovani Italian Cucina Int'l}" \]
\[ b = "\text{Giovani Italian Kitchen International}" \]
\[ T(a,b) = \{ \text{Equal}(\text{Giovani}, \text{Giovani}), \text{Equal}(\text{Italian}, \text{Italian}), \]
\[ \quad \text{Synonym}(\text{Cucina}, \text{Kitchen}), \text{Abbreviation}(\text{Int'q}, \text{International}) \} \]

Training:
\[
\begin{align*}
p(M) &= 0.31 & p(\neg M) &= 0.69 \\
p(\text{Equal} \mid M) &= 0.17 & p(\text{Equal} \mid \neg M) &= 0.027 \\
p(\text{Synonym} \mid M) &= 0.29 & p(\text{Synonym} \mid \neg M) &= 0.14 \\
p(\text{Abbreviation} \mid M) &= 0.11 & p(\text{Abbreviation} \mid \neg M) &= 0.03 \\
\end{align*}
\]

\[
p(M) \prod p(v_i \mid M) = 2.86 \times 10^{-4}
\]
\[
p(\neg M) \prod p(v_i \mid \neg M) = 2.11 \times 10^{-6}
\]

\[ \text{Score}_{\text{HFM}} = 0.993 \rightarrow \text{Good Match!} \]
Consider the following case

Pizza Hut Restaurant  ←→  Pizza Hut Rstrnt

Sabon Gari Restaurant  ←→  Sabon Gari Rstrnt

Should these scores equally well?
Introducing Fine-Grained Transformations

- Capture additional information about a relationship between tokens
  - Frequency information
    - Pizza Hut vs. Sabon Gari
  - Semantic category
    - Street Number vs. Apartment Number
- Parameterized transformations
  - $Equal[HighFreq] \ vs \ Equal[MedFreq]$
  - $Equal[FirstName] \ vs \ Equal[LastName]$
Fine-Grained Transformations
Frequency Considerations

Coarse Grained:

- Pizza Hut Restaurant
  - 2 Equal and 1 Abbreviation Transformation
  - Pizza Hut Rstrnt
- Sabon Gari Restaurant
  - 2 Equal and 1 Abbreviation transformations
  - Sabon Gari Rstrnt

Both score equally well.
Fine-Grained Transformations
Frequency Considerations

Fine Grained:

- Pizza Hut Restaurant
  - 2 high-frequency Equal transformations and 1 Abbreviation transformation
- Pizza Hut Rstrnt

- Sabon Gari Restaurant
  - 2 low-frequency Equal transformations and 1 Abbreviation transformation
- Sabon Gari Rstrnt

Sabon Gari Restaurant scores higher since low frequency equals are much more indicative of a match.
Fine-Grained Transformations
Semantic Categorization

Without Tagging:

123 Venice Boulevard, 405

Equal

405 Venice Boulevard, 123

Equal

Scores well

Equal

Equal
Fine-Grained Transformations
Semantic Categorization

With Tagging:

Scores poorly
A missing surname penalizes a score far more than a missing given name.
Global Transformations

- Applied to entire transformation graph
  - Reordering
    - “Steven N. Minton” vs. “Minton, Steven N.”
  - Subset
    - “Nissan 150 Pulsar with AC” vs. “Nissan 150 Pulsar”
Experimental Results

- We compared the following four systems:
  - HFM
  - TF-IDF (Vector-based cosine)
    - matches tokens
  - MARLIN
    - learned string edit distance
  - Active Atlas (older version)

- We made use of 4 datasets
  - Two restaurant datasets
  - One car dataset
  - One hotel dataset
Experimental Results

  - All methods calculate vector of feature scores
    - Pass to SVM trained to label matches/non-matches
    - Radial Bias Function kernel, $\gamma = 10.0$
  - 20 trials, cross-validation
    - Dataset randomly split into two folds for cross validation
    - Precision interpolated at 20 standard recall levels.
“Marlin Restaurants” Dataset
Fields: name, address, city, cuisine
Size: Fodors (534 records), Zagats (330 records), 112 Matches
Larger Restaurant Set With Duplicates
Fields: name, address
Size: LA County Health Dept. Website (3701), Yahoo LA Restaurants (438), 303 Matches
Car Dataset
Fields: make, model, trim, year
Attributes: Edmunds (3171), Kelly Blue Book (2777), 2909 Matches
Bidding for Travel

Fields: star rating, hotel name, hotel area
Size: Extracted posts (1125), “Clean” hotels (132), 1028 matches
## Result Summary

<table>
<thead>
<tr>
<th>Matching Technique</th>
<th>Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Marlin Res.</td>
</tr>
<tr>
<td>HFM</td>
<td>94.64</td>
</tr>
<tr>
<td>Active Atlas</td>
<td>92.31</td>
</tr>
<tr>
<td>TF-IDF</td>
<td>96.86</td>
</tr>
<tr>
<td>Marlin</td>
<td>91.39</td>
</tr>
</tbody>
</table>

Average maximum F-measure for detecting matching records. Note: red is not significant with respect to a 1-tailed paired t-test at confidence 0.05.
Discussion of Results

- Comparison to TFIDF
  - HFM outperforms TFIDF by identifying complex relationships which improve matching
    - Restaurant Datasets:
      - Tokens related mostly by equality
      - Minor improvement over TFIDF
    - Car Dataset:
      - Transformations yield large improvements (in particular, synonym and ordered concatenation transformations)

- Comparison to Active Atlas
  - HFM introduces fine-grained & global transformations
  - HFM based on a better justified statistical approach. (Improved scoring of transformations based on Naïve Bayes)

- Comparison to Marlin
  - Can handle larger datasets
  - Captures important token-level relationships not accessible to Marlin
  - Token-based and not character-based
Discussion / Conclusion

- Alternative to transformations: normalize/preprocess data
  - No normal form
    - Caitlyn $\rightarrow$ \{Catherine, Lynne\}

- Scalability
  - HFM does well on large, complex datasets
Acknowledgements

- We would like to thank:
  - Mikhail Bilenko for his kind help in helping us set up and run MARLIN on our datasets.
  - Sheila Tejada for her work on Active Atlas, the precursor to HFM
Questions / Comments

Thank you!
Field alignments are defined mappings between attribute(s) from one datasource to attribute(s) from another datasource.

<table>
<thead>
<tr>
<th>First Name</th>
<th>Last Name</th>
<th>Age</th>
<th>Gender</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raoul</td>
<td>DelaTorre</td>
<td>35</td>
<td>Male</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>SS#</th>
<th>Age</th>
<th>Gender</th>
</tr>
</thead>
<tbody>
<tr>
<td>De la Torre, Raul</td>
<td>N/A</td>
<td>36</td>
<td>M</td>
</tr>
</tbody>
</table>