A Graph-Theoretic Fusion Framework for Unsupervised Entity Resolution

Presented by: Dongxiang Zhang
## Entity Resolution

<table>
<thead>
<tr>
<th>Text Records</th>
<th>Identical Entity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Les Celebrites 160 Central Park S New York French</td>
<td>✓</td>
</tr>
<tr>
<td>Les Celebrites 155 W. 58th St. New York City French (Classic)</td>
<td></td>
</tr>
<tr>
<td>Palm 837 Second Ave. New York City Steakhouses</td>
<td>×</td>
</tr>
<tr>
<td>Palm Too 840 Second Ave. New York City Steakhouses</td>
<td></td>
</tr>
</tbody>
</table>

Two examples from the restaurant dataset.
Previous Work

- **Distance-based Methods**
  - Edit Distance, TF-IDF
  - Simple and scalable, but **not effective enough**

- **Learning-based Methods**
  - Learn a distance metric
  - Model ER as a classification task and apply SVM
  - Require considerable amount of training data

- **Crowd-based Methods**
  - CrowdER, TransM, TransNode, GCER, ADC, Power+
  - Achieve state-of-the-art accuracy but **require human intervention**
Our Objective

- Propose an unsupervised approach
  - More accurate when compared with distance-based methods
  - Require no training/labeling efforts when compared with learning-based methods
  - Require no human intervention and financial cost when compared with crowd-based methods
In the traditional unsupervised methods

- Step 1: Craft a distance measure between two records
- Step 2: Tune a threshold such that two records with similarity score higher than the threshold are considered as the same entity

We are motivated to improve these two steps by

- Proposing ITER algorithm to learn record similarity
- Proposing CliqueRank to estimate the likelihood of two records referring to the same entity
- Iteratively Reinforcing these two components
Unsupervised Fusion Framework

1. **Input Dataset**
2. **Bipartite Graph**
3. **Learning Record Similarity**
4. **Iter**
5. **Estimating Matching Probability**
6. **CliqueRank**
7. **Output Matching Pairs**
ITER Algorithm

- If a term only occurs in a group of matching records, then we consider the term as highly discriminative
  - Examples include product models for electronic devices or telephone numbers for restaurant.
  - These terms have low term frequency and may not be emphasized by TF-IDF
- If a term is shared by many non-matching records, its weight will be punished
ITER Algorithm

\[ x_t = \sum_{i \neq j} \frac{p(r_i, r_j) s(r_i, r_j)}{P_t} \]

\[ s(r_i, r_j) = \sum_{t \in r_i \land t \in r_j} \text{norm}(x_t) \]
Algorithm 1: ITER Algorithm

**Input:** Bipartite graph structure with edge weight \( p(r_i, r_j) \);
**Output:** Node salience \( x_t \) and record pair similarity score \( s(r_i, r_j) \);

1. Randomly initialize \( x_t \) in \((0, 1)\);
2. \textbf{while} \( x_t \) does not converge \textbf{do}
3. \hskip 1em \textbf{for} each record pair \((r_i, r_j)\) \textbf{do}
4. \hskip 2em Set its weight \( s(r_i, r_j) \leftarrow \sum_{t \in r_i \land t \in r_j} x_t \);
5. \hskip 1em \textbf{for} each term \( t \) \textbf{do}
6. \hskip 2em Set its weight \( x_t \leftarrow \sum_{i \neq j} \frac{p(r_i, r_j) s(r_i, r_j)}{P_t} \);
7. Set \( x_t = 1/(1 + \frac{1}{x_t}) \)
8. \textbf{return} \( x_t \) and \( s(r_i, r_j) \)
Given $Gr$, our goal is to identify matching probability.

Ideally, the probability should be 1 for matching pairs and 0 for non-matching pairs.
CliqueRank Algorithm

- Random-Walk based interpretation
  - Ideally, if \( r_i \) and \( r_j \) refer to different entities, they should be located in different cliques and not reachable from each other
  - Otherwise, if we start a random walk from one record \( r_i \), it will be very likely to visit the other record \( r_j \) within certain number of steps
Algorithm 2: Random-Surfer Sampling (RSS)

1. Construct a record graph $G_r$ based on $s(r_i, r_j)$;
2. for each edge $(r_i, r_j) \in G_r$ do
3.  $c_1 \leftarrow 0$; $c_2 \leftarrow 0$
4.  for $m \leftarrow 1$; $m \leq M/2$; $m ++$ do
5.  $c_1 \leftarrow c_1 + \text{RandomWalk}(r_i, r_j)$;
6.  for $m \leftarrow 1$; $m \leq M/2$; $m ++$ do
7.  $c_2 \leftarrow c_2 + \text{RandomWalk}(r_j, r_i)$;
8.  $p(r_i, r_j) \leftarrow (c_1 + c_2)/M$;
9. return $p(r_i, r_j)$;
# Random Walk Algorithm

**Algorithm 3: RandomWalk**(start, target)

1. cur ← start;
2. for s ← 1; s ≤ S; s ++ do
3. pick a random value \( b \in (0, 1) \);
4. \( s'(cur, target) \leftarrow (1 + b) \cdot s(cur, target) \);
5. next ← pick a node from neighbors of cur based on the
   new transition probability \( p_b(r_i \rightarrow r_j) \);
6. if next == target then
   7. return 1;
8. if edge \((next, target) \notin G_r\) then
   9. return 0;
10. cur ← next;
11. return 0;

To handle large cliques

\[
p(r_i \rightarrow r_j) = \frac{s(r_i, r_j)^\alpha}{\sum_{r_j \in O(r_i)} s(r_i, r_j)^\alpha},
\]

To champion edge with high score

For early termination
CliqueRank Algorithm

- Iterative sampling is slow, and we switch to matrix operation

- $M_t$ be the matrix with reaching probability from $r_i$ to $r_j$ with 1 step

$$M_t[i, j] = p(r_i \rightarrow r_j)$$

- $M_t^S$ be the matrix with reaching probability from $r_i$ to $r_j$ with $S$ steps

$$M_t^S = M_t \times M_t \times \ldots \times M_t$$

- Iterative sampling is slow, and we switch to matrix operation

- The random surfer algorithm essentially estimates such probability
CliqueRank Algorithm

- We make customizations to the RSS algorithm

- $M_b[i, j]$ be the initial transition probability matrix

- $p_b(r_i \rightarrow r_j) = \frac{(1+b)^\alpha s(r_i, r_j)^\alpha}{\text{norm}(r_i, r_j)}$.

- $M_n[i, j]$ is set to 1 if $r_i$ to $r_j$ are connected in $Gr$

- Finally, we can define the reaching probability with $S$ steps

$$M_t^S = \begin{cases} M_b & \text{if } S = 1 \\ M_t \times (M_t^{S-1} \odot M_n) & \text{if } S > 1 \end{cases}$$
Benchmark Datasets

- **Restaurant**
  - 858 non-identical restaurant records.
  - Each record contains the information of restaurant name and address.

- **Product**
  - 1081 records from the abt website and the other 1092 records from the buy website.
  - Each product record contains its name and descriptive information.

- **Paper**
  - 1865 non-identical publication records.
  - Each record has a cluster id and its textual information consists of authors, title, publication venue and year.
Experimental Setup

- For the three datasets, we use the same setting of parameters:
  - $\alpha=20$
  - $S=20$
  - $\eta=0.98$
  - 5 iterations between the reinforcement of ITER and CliqueRank

- Eigen library is used to boost matrix multiplication

## Accuracy

<table>
<thead>
<tr>
<th>Method</th>
<th>Restaurant</th>
<th>Product</th>
<th>Paper</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>String-distance based approaches</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jaccard</td>
<td>0.836</td>
<td>0.332</td>
<td>0.792</td>
</tr>
<tr>
<td>TF-IDF</td>
<td>0.871</td>
<td>0.658</td>
<td>0.821</td>
</tr>
<tr>
<td><strong>Machine-learning based approaches</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gaussian Mixture Model [5]</td>
<td>0.704</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>HGM+Bootstrap [5]</td>
<td>0.844</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>MLE [5]</td>
<td>0.904</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>SVM [6]</td>
<td>0.922</td>
<td>-</td>
<td>0.824</td>
</tr>
<tr>
<td><strong>Crowd-sourcing based approaches</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CrowdER [8]</td>
<td>0.934</td>
<td>0.800</td>
<td>0.824</td>
</tr>
<tr>
<td>TransM [10]</td>
<td>0.930</td>
<td>0.792</td>
<td>0.740</td>
</tr>
<tr>
<td>GCER [9]</td>
<td>0.930</td>
<td>0.760</td>
<td>0.785</td>
</tr>
<tr>
<td>ACD [12]</td>
<td>0.934</td>
<td>0.805</td>
<td>0.820</td>
</tr>
<tr>
<td>Power+ [13]</td>
<td>0.934</td>
<td>-</td>
<td>0.820</td>
</tr>
<tr>
<td><strong>Graph-theoretic baselines</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SimRank</td>
<td>0.645</td>
<td>0.376</td>
<td>0.730</td>
</tr>
<tr>
<td>PageRank</td>
<td>0.905</td>
<td>0.564</td>
<td>0.316</td>
</tr>
<tr>
<td>Hybrid</td>
<td>0.946</td>
<td>0.593</td>
<td>0.748</td>
</tr>
<tr>
<td><strong>Proposed graph-theoretic fusion framework</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ITER+CliqueRank</td>
<td>0.927</td>
<td>0.764</td>
<td>0.890</td>
</tr>
</tbody>
</table>
## Experiment & Analysis

### Efficiency

<table>
<thead>
<tr>
<th></th>
<th>Restaurant</th>
<th>Product</th>
<th>Paper</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of nodes in $G_r$</td>
<td>858</td>
<td>2173</td>
<td>1865</td>
</tr>
<tr>
<td>Number of edges in $G_r$</td>
<td>5,320</td>
<td>151,939</td>
<td>980,780</td>
</tr>
<tr>
<td>Total running time</td>
<td>1.1min</td>
<td>21.6min</td>
<td>24.2min</td>
</tr>
<tr>
<td>Running time for ITER</td>
<td>3sec</td>
<td>20sec</td>
<td>58sec</td>
</tr>
<tr>
<td>Speedup compared to RSS</td>
<td>1.3x</td>
<td>1.5x</td>
<td>60x</td>
</tr>
</tbody>
</table>
Experiment & Analysis

- Effectiveness of Learned Term Weights

ground-truth score: \[ score(t) = \frac{\sum_{t \in r_i \land t \in r_j} I(r_i, r_j)}{P_t} \]
### Top-Ranked Terms in the Benchmark Datasets

<table>
<thead>
<tr>
<th>Category</th>
<th>Terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restaurant</td>
<td>coyote, chin’s, 702/731-7547, 702/734-0410, 702/791-7111, 3645, 3400, gotham, 2880, arnie, seasons, 12, gramercy, chinois</td>
</tr>
<tr>
<td>Product</td>
<td>85w, trackpad, mirroring, magsafe, led-backlit, isight, displayport, 5400-rpm, spreadsheets, formulas, dramatically, compromising, multi-angle, 30p, 24mbps, diameter, s320, 7mm</td>
</tr>
<tr>
<td>Paper</td>
<td>thurn, wentzel, pachovicz, dzeroski, bloendorn, dze-roski, vafaic, kreusiger, kaufaman, pachowitz, re-ich, dzerowski, weldel, cmu-cd-91-197, jianping, jerzy, janusz, juergen, haleh, cmu-cs-91-197</td>
</tr>
</tbody>
</table>
Experiment & Analysis

- Convergence of ITER

![Graph showing the convergence of ITER with three categories: Restaurant, Product, and Paper.](chart.png)
## Effect of Reinforcement

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Restaurant F1-score</th>
<th>Restaurant Time</th>
<th>Product F1-score</th>
<th>Product Time</th>
<th>Paper F1-score</th>
<th>Paper Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.916</td>
<td>13</td>
<td>0.543</td>
<td>253</td>
<td>0.844</td>
<td>207</td>
</tr>
<tr>
<td>2</td>
<td>0.935</td>
<td>25</td>
<td>0.712</td>
<td>514</td>
<td>0.888</td>
<td>515</td>
</tr>
<tr>
<td>3</td>
<td>0.931</td>
<td>39</td>
<td>0.747</td>
<td>768</td>
<td>0.889</td>
<td>819</td>
</tr>
<tr>
<td>4</td>
<td>0.931</td>
<td>52</td>
<td>0.754</td>
<td>1027</td>
<td>0.890</td>
<td>1135</td>
</tr>
<tr>
<td>5</td>
<td>0.927</td>
<td>64</td>
<td>0.764</td>
<td>1296</td>
<td>0.890</td>
<td>1453</td>
</tr>
</tbody>
</table>
Conclusion

- We propose an unsupervised graph-theoretic framework for entity resolution.

- Two novel algorithms ITER and CliqueRank are proposed, one for term-based similarity and the other for topological confidence. These two components can reinforce each other.

- Experimental results on three benchmark datasets show that our algorithm is accurate.

Codes are available at: https://github.com/uestc-db/Unsupervised-Entity-Resolution
Thank you!

Q&A