Online Person Name Disambiguation with Constraints

Madian Khabsa\textsuperscript{1,3}, Pucktada Treeratpituk\textsuperscript{2}, C. Lee Giles\textsuperscript{1}

\textsuperscript{1}The Pennsylvania State University
\textsuperscript{2}Ministry of Science and Technology, Thailand
\textsuperscript{3}Microsoft Research
giles@ist.psu.edu

Person Name Disambiguation

- **Goal**: name mentions => real world people
  - To group all the name mentions of a person together

- **Applications**
  - More accurate people search (search engine, digital libraries)
  - Information integration
    - Merging multiple databases e.g. patient records
  - Enhancing further data analysis
    - Citation counting
    - Social network analysis
    - Analyzing people mentions in blogs, news articles
Background – our work

Information extraction from scholarly documents:
- Traditional metadata
  - Authors, affiliations, abstracts, citations
- Tables
- Figures
- Chemical formulae
- Algorithms

Online system
- http://citeseerx.ist.psu.edu
How important is this?

- 11-17% of queries to AllTheWeb and AltaVista contain personal names [Panderson et al., 09]
- 9-19% of search requests to CiteSeerX are author names
- Generally, at least 4 out of 10 most popular queries on Google (Trends) are people names
- Lots of personal information spreading across various sites
Difficulty

Person Name Ambiguity

1. Name Variation (one to many)
   - One person uses multiple name variations
     - William Jefferson Clinton, William J. Clinton, Bill Clinton
     - Salvador Dali, Salvador Dali Domenech
   - % of Spanish authors who appeared under more than one name: 48.1% in SCI (Science Citation Index), 50.7% in MEDLINE, 69.0% in IME (Indice Medico Espanol). [Ruiz-Perez et al, 02]

2. Common Name (many to one)
   - Two or more people share the same name
   - 1990 US Census: 90,000 names are shared by 100 millions people [Artiles et al, SIGIR05]

3. Data Entry Error – both by human and machines

Person name ambiguity is a many-to-many mapping!!!
Online Disambiguation with Constraints

Problem:
- Given a set of people mentions, profile \( \{ p_i \} \), where each profile \( p_i \) is associated with a set of features \( \langle f_1, f_2, \ldots, f_K \rangle \)
- To generate a set of people clusters \( \{ C_j \} \), where each cluster \( C_j = \{ p_s \} \) and for all profile pair \( \langle p_s, p_t \rangle \), both \( p_s, p_t \) are in the same cluster \( C_j \) if and only if both \( p_s, p_t \) refer to the same person

Prior Work (also a part of NER – named entity recognition)
- Link-structure
  - Hyperlink structure (Rekkeman & McCallum, WWW05)
- Metadata-based
  - Probabilistic model (Torvik et al, JASIST05)
  - SVM (Bilenko et al, IS03, Han et al, JCDL04, Huang et al, PKDD06)
- Content-based
  - Topic model (Song et al, JCDL07, Tang et al, SIGKDD08)
Previous Limitations

Constraint limitations not always easy to implement
- Why constraints? => improve quality of clusters
  - User corrections – e.g. cannot-link constraints
  - Expert knowledge and heuristics

All are in batch mode
- Disambiguate all profiles at once
- New profiles show up
  - have to rerun everything, time-consuming and not very practical
  - Or wait until there are enough new records then rerun, causing delay in the disambiguation result
- Want online disambiguation
  - Iteratively disambiguate new profiles as it show up
  - Discover new people clusters?
## Constraints: Example

| Name | A) Execution Based Evaluation of Multistage Interconnection Networks for Cache-Coherent Multiprocessors  
Name: Akhilesh Kumar  
Affil: Intel Corporation Department of Computer Science, 2200 Mission College Blvd Texas AM University, Santa Clara College Station |
|------|------------------------------------------------------------------------------------------------------|
| Name: A Kumar  
Affil: Department of Computer Science, Texas AM University | B) FFT Implementations on nCUBE Multiprocessor |
| Name: Amit Kumar  
Affil: Department of Computer Science, Texas AM University | C) Real-Time Communication in FDDI-Based Reconfigurable Networks |

A ~ B (both multiprocessors), B ~ C (same affiliation)

- So most likely the algorithm will cluster \{A,B,C\} together
- But we know A != C (Akhilesh Kumar != Amit Kumar)
- So we should enforce constraints on a cluster that all records in the cluster need to have compatible names
Types of Constraints

**Instance-level Constraints**
- Do not perform pairwise comparison if do not satisfy the constraint
- Cheaper to enforce, no maintenance needed

**Temporal proximity**
- Records of a single person should be continuous in time, so only make a comparison within +/- 3 years windows
- e.g. do we need to compare an author from 1985 with an author from 2002

**Cluster-level Constraints**
- Maintain a data structure to keep track of constraints for each cluster
- **Name compatibility**

![Diagram of name compatibility constraints]

- WH Gates
- WH Gates
- WH Gates
- WH Gates
- William Gates
- William H Gates
- William H Gates
- William H Gates
- William H Gates
- William M Gates
- William M Gates
- William M Gates
Basics of our Name Disambiguation Algorithm

- record1
  - name
  - affiliation
  - address
  - paper title, etc.

Linking-Function
- String Distances
- Decision Trees (Tajeda et al, 2001)
- SVMs (Huang et al, 2006)

Clustering Algorithm
- K Means
- HAC
DBSCAN

Density Based Spatial Clustering of Applications with Noise

Basic idea:
- If an object $p$ is **density connected** to $q$,
  - then $p$ and $q$ belong to the same cluster
- If an object is **not density connected** to any other object
  - it is considered noise
Concepts: \( \varepsilon \)-Neighborhood

- **\( \varepsilon \)-Neighborhood** - Objects within a radius of \( \varepsilon \) from an object. (epsilon-neighborhood)

- **Core objects** - \( \varepsilon \)-Neighborhood of an object contains at least MinPts of objects

\( \varepsilon \)-Neighborhood of \( p \)
\( \varepsilon \)-Neighborhood of \( q \)

\( p \) is a core object (MinPts = 4)

\( q \) is not a core object

Thanks to Arpan Maheshwari
Directly density-reachable

An object $q$ is directly density-reachable from object $p$ if $q$ is within the $\varepsilon$-Neighborhood of $p$ and $p$ is a core object.

- $q$ is directly density-reachable from $p$
- $p$ is not directly density-reachable from $q$

Thanks to Arpan Maheshwari
Disambiguation Algorithm

- **DBSCAN (density-based clustering)**
  - Find a cluster based on density, no need to specify K
  - Random Forest – as the similarity function (distance between two profile)

- **DBSCAN_C (DBSCAN + constraints)**
  - Basic idea:
    - when expanding a cluster, filter out records that do not satisfy existing constraints (instant-level and cluster-level)
    - Also update cluster constraints when a record is added to a cluster
  - Define *mergeRecord* procedure
    - Given an existing clustering result and a new record, create a new clustering result by
      - Create a new cluster
      - Add a new record to an existing cluster
      - Merge two existing clusters
Online DBSCAN with Constraints

**Procedure 3 DBSCAN\(c\)\(p\)**

Input: \(D\) - static collections of records to be disambiguated
mark all records in \(D\) as UNVISITED
for all record \(p\) in \(D\) do
  if \(p\) is UNVISITED then
    mark \(p\) as VISITED
    \(N\) ← query\((D, p, ε)\)
    sort records in \(N\) by their distance from \(p\)
    \(N\) ← IConsFilter\((p, N)\)
    if \(|N| < minPts\) then
      assign \(p\) → NOISE
    else
      expandCluster\((p, N)\)
  end if
end for

**Procedure 4 expandCluster\((p, N)\)**

1: \(cid ← nextClusterId()\)
2: assign \(p → cid\)
3: \(Q ← N^c\) /* put records in region into a queue */
4: while \(Q ≠ \emptyset\) do
5:   \(q ← \text{pop a record from } Q\)
6:   if \(q\) is UNVISITED then
7:     mark \(q\) as VISITED
8:     \(N' ← \text{query}(D, q, ε)\)
9:     sort records in \(N'\) by their distance from \(q\)
10: \(N' ← \text{IConsFilter}(q, N')\)
11: \(N' ← \text{orderedIConsFilter}(N')\)
12: \(N' ← \text{CConsFilter}(cid, N')\)
13: if \(|N'| ≥ minPts\) then
14:   /* append \(N'\) to the end of \(Q\) */
15:   \(Q ← Q + N'\)
16: end if
17: end if
18: if \(q\) doesn’t belong to any cluster then
19:   assign \(q → cid\)
20: end if
21: end while

**Procedure 5 mergeRecord\((p)\)**

Input: \(p\) is a new record added to \(D\), not yet processed
1: \(N ← \text{query}(D, p, ε)\)
2: sort records in \(N\) by their distance from \(p\)
3: \(N ← \text{IConsFilter}(p, N)\)
4: if \(|N| < minPts\) then
5:   assign \(p → NOISE\)
else
6:   \(C ← \text{set of clusters } C_i \text{ such that } \forall C_j, C_i ∩ N ≠ \emptyset\)
7:   if \(C ≠ \emptyset\) then
8:     \(L ← \emptyset\)
9:     for all \(C_i ∈ C\) do
10:       if \(\emptyset ≠ \text{CConsFilter}(i, \{p\})\) then
11:         \(L ← L ∪ \{C_i\}\)
12:       end if
13:     end if
14:     sort \(C_i ∈ L\) by \(|C_i ∩ N|\) in descending order
15:     \(C_k ← \text{the cluster } C_i ∈ L\) with the biggest intersection
16:     \(k ← \text{nextClusterId}()\)
17:     end if
18:   else
19:     \(k ← \text{nextClusterId}()\)
20:     end if
21:   end if
22:   assign \(p → k\)
23:   for all \(C_i \in L \setminus \{C_k\}\) do
24:     if \(C_i = \text{CConsFilter}(k, C_i)\) then
25:       \(C_k ← C_k ∪ C_i\) /* merge \(C_i\) to \(C_k\) */
26:     end if
27:   end for
28:   noises ← \(\{q|q ∈ N \text{ and } q \notin C_i, \forall C_i ∈ C\}\)
29:   /* noises retained the sorted order of \(N\) */
30:   noises ← \text{orderedIConsFilter}(noises)
31: noises ← \text{CConsFilter}(*, noises)
32: for all \(q\) in noises do
33:   assign \(q → k\)
34: end for
Online DBSCAN with Constraints

**Idea**

- If the neighborhood of a point is dominated by a cluster, assign the point to that cluster.
- If multiple clusters dominate the neighborhood, pick the one with most intersection.
- Try to merge the clusters that occupy the neighborhood, if they pass the constraints.

---

**Procedure 3 mergeRecord(p)**

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$N \leftarrow \text{query}(D, p, \varepsilon)$</td>
</tr>
<tr>
<td>2</td>
<td>sort records in $N$ by their distance from $p$</td>
</tr>
<tr>
<td>3</td>
<td>$N \leftarrow \text{IConsFilter}(p, N)$</td>
</tr>
<tr>
<td>4</td>
<td>if $</td>
</tr>
<tr>
<td>5</td>
<td>assign $p \rightarrow \text{NOISE}$</td>
</tr>
<tr>
<td>6</td>
<td>else</td>
</tr>
<tr>
<td>7</td>
<td>$C \leftarrow$ set of clusters $C_i$, such that $\forall C_i, C_i \cap N \neq \emptyset$</td>
</tr>
<tr>
<td>8</td>
<td>if $C \neq \emptyset$ then</td>
</tr>
<tr>
<td>9</td>
<td>$L \leftarrow \emptyset$</td>
</tr>
<tr>
<td>10</td>
<td>for all $C_i \in C$ do</td>
</tr>
<tr>
<td>11</td>
<td>if $\emptyset \neq \text{CConsFilter}(i, {p})$ then</td>
</tr>
<tr>
<td>12</td>
<td>$L \leftarrow L \cup {C_i}$</td>
</tr>
<tr>
<td>13</td>
<td>end if</td>
</tr>
<tr>
<td>14</td>
<td>end for</td>
</tr>
<tr>
<td>15</td>
<td>sort $C_i \in L$ by $</td>
</tr>
<tr>
<td>16</td>
<td>$C_k \leftarrow$ the cluster $\in L$ with the biggest intersection</td>
</tr>
<tr>
<td>17</td>
<td>else</td>
</tr>
<tr>
<td>18</td>
<td>$k \leftarrow \text{nextClusterId}()$</td>
</tr>
<tr>
<td>19</td>
<td>end if</td>
</tr>
<tr>
<td>20</td>
<td>assign $p \rightarrow k$</td>
</tr>
<tr>
<td>21</td>
<td>for all $C_i$ in $L \setminus {C_k}$ do</td>
</tr>
<tr>
<td>22</td>
<td>if $C_i = \text{CConsFilter}(k, C_i)$ then</td>
</tr>
<tr>
<td>23</td>
<td>$C_k \leftarrow C_k \cup C_i$ /* merge $C_i$ to $C_k$ */</td>
</tr>
<tr>
<td>24</td>
<td>end if</td>
</tr>
<tr>
<td>25</td>
<td>end for</td>
</tr>
<tr>
<td>26</td>
<td>noises $\leftarrow {q</td>
</tr>
<tr>
<td>27</td>
<td>/* noises retained the sorted order of $N$ */</td>
</tr>
<tr>
<td>28</td>
<td>noises $\leftarrow \text{orderedIConsFilter}(\text{noises})$</td>
</tr>
<tr>
<td>29</td>
<td>noises $\leftarrow \text{CConsFilter}(cido, \text{noises})$</td>
</tr>
<tr>
<td>30</td>
<td>for all $q$ in noises do</td>
</tr>
<tr>
<td>31</td>
<td>assign $q \rightarrow k$</td>
</tr>
</tbody>
</table>
Evaluation: Similarity Function

Random Forest (Treeratpituk and Giles, JCDL09)

Features
- Name (personal names + emails) [6]
- Affiliation [3]
- Coauthors (names + affiliations) [6]
- Venue (venues + years) [4]
- Content (abstracts + titles) [5]
- Keyphrases [5]
- Citations [2]

24 features (JCDL09)
- TFIDF, softTFIDF, Jaccard, #shared, #shared-IDF, etc.
- IDF, Jaccard, nPMI (Sum, Max, Avg)
- bibliographic coupling, co-citations

SEERLAB keyphrase extractor (Treeratpituk et al, ACL10)
Evaluation: Data

- CiteSeer author dataset
  - 10 highly ambiguous names

- Two similarity distances (random forest)
  - MIX
    - 24 features [JCDL09]
  - MIX+CKP
    - With citation and keyphrases features

<table>
<thead>
<tr>
<th>Data</th>
<th>#Rec</th>
<th>#Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>A. Gupta</td>
<td>498</td>
<td>45</td>
</tr>
<tr>
<td>A. Kumar</td>
<td>139</td>
<td>31</td>
</tr>
<tr>
<td>C. Chen</td>
<td>525</td>
<td>99</td>
</tr>
<tr>
<td>D. Johnson</td>
<td>345</td>
<td>40</td>
</tr>
<tr>
<td>J. Anderson</td>
<td>307</td>
<td>40</td>
</tr>
<tr>
<td>J. Robinson</td>
<td>111</td>
<td>27</td>
</tr>
<tr>
<td>J. Smith</td>
<td>729</td>
<td>83</td>
</tr>
<tr>
<td>K. Tanaka</td>
<td>52</td>
<td>19</td>
</tr>
<tr>
<td>M. Jones</td>
<td>348</td>
<td>51</td>
</tr>
<tr>
<td>M. Miller</td>
<td>226</td>
<td>35</td>
</tr>
</tbody>
</table>
Evaluation Criteria

- Standard clustering measures
  - \( C \) = clusters to be evaluated
  - \( L \) = gold standard clusters
  - \( n \) = number of items in \( L \)

- Pairwise Precision
  \[
  \text{Pairwise Precision} = \frac{\text{Number of correctly formed pairs}}{\text{Number of formed pairs}}
  \]

- Pairwise Recall
  \[
  \text{Pairwise Recall} = \frac{\text{Number of correctly formed pairs}}{\text{Number of pairs in } L}
  \]
Pairwise Recall Example

R1 = a, b, c, d, e, f, g, h
R2 = a, b, c, d, e, f, g, h
G = a, b, c, d, e, f, g, h

Pairs:
- ef, eg, eh, fg, fh, gh
- ab, cd, ef, gh
- ab, cd, ef, gh
- ab, cd, ef, eg
- eh, fg, fh, gh
- 6 pairs, all in G
- 4 pairs, all in G
- 8 pairs

Recall = 6/8 = 75%
Recall = 4/8 = 50%
Pairwise precision = 1

Credit: David Menestrina @ Stanford
Evaluation Criteria

- Standard clustering measures
  - $C =$ clusters to be evaluated
  - $L =$ gold standard clusters
  - $n =$ number of items in $L$

\[
Purity = \sum_{i} \frac{|C_i|}{n} \max \text{Precision}(C_i, L_j)
\]

\[
InversePurity = \sum_{i} \frac{|L_i|}{n} \max \text{Precision}(L_i, C_j)
\]

\[
\text{Precision}(C_i, L_j) = \frac{|C_i \cap L_j|}{|C_i|}
\]
Feature Analysis

<table>
<thead>
<tr>
<th>Similarity Model</th>
<th>Accuracy</th>
<th>RCS</th>
<th>pP</th>
<th>pR</th>
<th>pF1</th>
<th>cP</th>
<th>cR</th>
<th>cF1</th>
<th>Purity</th>
<th>InvPurity</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>94.6%</td>
<td>2.08</td>
<td>0.69</td>
<td>0.68</td>
<td>0.65</td>
<td>0.28</td>
<td>0.46</td>
<td>0.34</td>
<td>0.83</td>
<td>0.68</td>
</tr>
<tr>
<td>affiliation</td>
<td>91.3%</td>
<td>2.47</td>
<td>0.61</td>
<td>0.68</td>
<td>0.54</td>
<td>0.53</td>
<td>0.24</td>
<td>0.54</td>
<td>0.73</td>
<td>0.63</td>
</tr>
<tr>
<td>coauthors</td>
<td>93.6%</td>
<td>2.16</td>
<td>0.98</td>
<td>0.48</td>
<td>0.62</td>
<td>0.30</td>
<td>0.61</td>
<td>0.40</td>
<td>0.97</td>
<td>0.58</td>
</tr>
<tr>
<td>venue</td>
<td>89.6%</td>
<td>4.43</td>
<td>0.64</td>
<td>0.17</td>
<td>0.25</td>
<td>0.12</td>
<td>0.49</td>
<td>0.19</td>
<td>0.78</td>
<td>0.28</td>
</tr>
<tr>
<td>abstract</td>
<td>91.6%</td>
<td>1.07</td>
<td>0.45</td>
<td>0.86</td>
<td>0.52</td>
<td>0.41</td>
<td>0.43</td>
<td>0.40</td>
<td>0.61</td>
<td>0.82</td>
</tr>
<tr>
<td>keyphrases</td>
<td>92.5%</td>
<td>1.24</td>
<td>0.46</td>
<td>0.76</td>
<td>0.50</td>
<td>0.36</td>
<td>0.44</td>
<td>0.49</td>
<td>0.65</td>
<td>0.78</td>
</tr>
<tr>
<td>citations</td>
<td>92.5%</td>
<td>1.81</td>
<td>0.73</td>
<td>0.63</td>
<td>0.63</td>
<td>0.32</td>
<td>0.57</td>
<td>0.41</td>
<td>0.83</td>
<td>0.67</td>
</tr>
<tr>
<td>MIX</td>
<td>96.8%</td>
<td>1.03</td>
<td>0.81</td>
<td>0.94</td>
<td>0.86</td>
<td>0.69</td>
<td>0.69</td>
<td>0.69</td>
<td>0.89</td>
<td>0.87</td>
</tr>
<tr>
<td>MIX+CKP</td>
<td>96.9%</td>
<td>1.02</td>
<td>0.85</td>
<td>0.96</td>
<td>0.90</td>
<td>0.76</td>
<td>0.76</td>
<td>0.76</td>
<td>0.92</td>
<td>0.88</td>
</tr>
</tbody>
</table>

- Compared single feature similarity with MIX, MIX+CKP
- Using keyphrases + citations (MIX+CKP) improve quality of clusters pF1=0.90 (+4%), cF1 = 0.76 (+7%)
Constraints

**Temporal Proximity**
- Instance-level constraint
- Disjunctive constraint
  - To satisfy a cluster-level constraint of C, a record only needs to satisfy the instant-level constraint with any records in C

**Name Compatibility**
- Cluster-level constraint
- Conjunctive constraint
  - The name of every record in a cluster C must be compatible with each other
Effect of Constraints

Constraints consistently improve pF1, cF1
- none < instance < cluster
- Cluster-level pF1=0.95 (+5%), cF1=0.79 (+3%) over no constraints

MIX+CKP with cluster constraints outperforms previous technique (LASVM): +4% in pF1 and +15% in cF1
Online Disambiguation

Setup:
1. randomly select 20% of records as initial set
2. Run batch disambiguation on the initial set
3. Add each record in the 80% set 1-by-1, using the *mergeRecord* procedure

- RCS generally stays around 1.0 (or goes down), mean that the new author clusters are being discovered
- pF1 consistently increase, means new record help improve existing clusters (also for invPurity)
Conclusion

Constraints can be particularly useful in a digital library or other situations where users are allowed to make corrections.

We propose a novel variation of the DBSCAN-based clustering algorithm that allows constraints to be injected into the disambiguation processes.

People disambiguation with constraints + online setting
- Constraints => pF1=0.95 (+5%), cF1=0.79 (+3%)
- DBSCANc can be used for iterative disambiguation while maintaining disambiguation quality

Recently disambiguated all 80 million name mentions in PubMed; paper in preparation
Future Work

Constraints
- Cannot-link from user corrections
- More efficient blocking-function (with charNgram indexes)

Scalability issues
- Map reduce, etc.
- Graph models