Name Disambiguation in Digital Libraries

The Pennsylvania State University

Dongwon Lee

dongwon@psu.edu

Outline

- Warm-Up
- Motivation & Problem Def.
- Disambiguation by Graphs
- Disambiguation by Groups
- Disambiguation by Googling
- Conclusion

Penn State University

- Founded in 1855
- 23 campuses throughout PA state
- Main campus at State College, PA
- 84,000 students, 20,800 faculty
- $1.2 billion endowment
- "Nittany Lion"
- Penn State ≠ U. Penn

- Two CompSci-related divisions:
  - Dept. of Computer Science & Engineering (CSE)
  - College of Info. Sciences & Technology (IST)

Penn State University

- 5 DL/DB Faculty
- CSE:
  - Wang-Chien Lee
- IST:
  - C. Lee Giles
  - Dongwon Lee
  - Prasenjit Mitra
  - James Wang
- Active Collaboration

Penn State University

- State College, PA
  - Out of nowhere, but close to everywhere
- West: 2.5 hours to Pittsburgh
- East: 4 hours to New York
- South: 3 hours to Washington DC
- North: 3 hours to Buffalo

Penn State University

- BLAST
- CiteSeer.IST
- Scientific Literature Digital Library
Penn State University
- Qiankun Zhao from NTU
  - PostDoc with Prasenjit Mitra
- In 2007, plan to hire 1-2 faculty on
  - Security
  - Risk Analysis
  - Data Mining
- Encourage to apply
  - http://ist.psu.edu/ist/facultyrecruiting/

QUAGGA Project
- Data Cleaning project @ Penn State
  - http://pike.psu.edu/quagga/
- Goals:
  - Scalable
  - Semantic and context-aware
  - DB-centric system-building

QUAGGA Project
- This talk is mainly based on:
  - “Group Linkage”, ICDE 2007
  - “Improving Grouped-Entity Resolution using Quasi-Cliques”, ICDM 2006
  - “Google Name Linkage”, Penn State TR, 2006
  - “Search Engine Driven Author Name Disambiguation”, JCDL 2006
- Slides for this talk are available at:
  - http://pike.psu.edu => talks

Credits
- Students
  - Ergin Elmacioglu (Penn State, USA)
  - Yee Tan Fan (NUS, Singapore)
  - Byung-Won On (Penn State, USA)
- Collaborators
  - Min-Yen Kan (NUS, Singapore)
  - Jaewoo Kang (Korea U., Korea)
  - Nick Koudas (U. Toronto, Canada)
  - Jian Pei (Simon Fraser U., Canada)
  - Divesh Srivastava (AT&T Labs – Research, USA)
  - Yi Zhang (UC. Santa Cruz, USA)

Outline
- Warm-Up
- Motivation & Problem Def.
- Disambiguation by Graphs
- Disambiguation by Groups
- Disambiguation by Googling
- Conclusion

Eg. ACM DL Portal
Jeffrey D. Ullman
@ Stanford Univ.
Eg. DBLP

1. U. Western Ontario
2. Fudan University
3. U. New South Wales
4. UNC, Chapel Hill

1. U. Western Ontario
2. Fudan University
3. U. New South Wales
4. UNC, Chapel Hill

Eg. WWW

Eg. People Names

- Most common activities of Internet users
  - ~30% of search engine queries include person names (R. Guha et al., WWW 2004)
- Highly ambiguous
  - only 90,000 different names for 100 million people (U.S. Census Bureau)
  - Valid changes:
    Customs: Lee, Dongwon vs. Dongwon Lee vs. LEE Dongwon
    Marriage: Carol Dusseau vs. Carol Arpaci-Dusseau
    Misc.: Sean Engelson vs. Shlomo Argamon
- Results:
  - mixture of web pages or query results about different people with the same name

Eg. IMDB & Wikipedia

"Citizen" (2016) (TV Series)

Clementine's Bittersweet Journey

We learn that it’s 1993 and we’re in the middle of a love triangle.
Eg. Product Names
- Automobile models
  - Honda Fix vs. Honda Jazz
- Companies
  - T-Fal vs. Tefal
- Electronic devices
  - Apple iPod Nano 4GB vs. 4GB iPod nano 4GB
  - Apple iPhone vs. Canadian iPhone
- Location
  - Paris at Europe vs. at USA

Eg. Drug Names
- Confusion due to look-alike or sound-alike drug names:
  - Primaxin (antibiotic inject.) – Primacor (hypertension inject.)
  - Amaryl – Amikin, Flomax – Volmax, Zantac – Xanax
  - 44,000 – 98,000 fatalities each year
  - Institute of Medicine Report, 1999
- Automatic identification of similar drug names has an important implication

Name Disambiguation Problem
- When names of entities (eg, people, products, companies, drugs) are:
  - Mixed ⇔ sort them out
  - Split ⇔ link them out

Name Disambiguation Problem: The process of detecting and correcting ambiguous named entities that represent the same real-world object

Terminology
- Entity: real-world object (eg, person, product, drug, company, etc)
- We view that Entity has two main information
  - name: textual description of the entity
  - contents: metadata or contents describing the entity

Eg.

Landscape
- Abundant research on related problems
- Split names
  - DB: approximate join, merge/purge, record linkage
  - DL: citation matching
  - AI: identity uncertainty
  - LIS: name authority control
- Mixed names
  - DM: k-way clustering
  - DL: author name disambiguation
  - NLP: word sense disambiguation
  - IR: query results grouping

Landscape
- In a nutshell, existing approaches often do:
  - For two entities, \(e1\) and \(e2\), capture their information in data structures, \(D(e1)\) and \(D(e2)\)
  - Measure the distance or similarity between data structures: \(dist(D(e1), D(e2)) = d\)
  - Determine for matching:
    - If \(d < \text{threshold}\), then \(e1\) and \(e2\) are matching entities
- Work well for common applications
- Ours do name disambiguation better when
  - Entities have structures that we can exploit, or
  - Entities lack useful information
Outline

- Warm-Up
- Motivation & Problem Def.
- Disambiguation by Graphs
- Disambiguation by Groups
- Disambiguation by Googling
- Conclusion

Key Idea

- When entities have contents that can be captured as graphs, let’s exploit it
- In DL, entities often have
  - A set of co-authors to work with
  - A set of venues to submit to
  - A set of topics to work on
- If we capture these information as graphs, it may yield better results than using simple distance

Using Graphs

- Represent entity e₁, as graph g₁ using common tokens
  - Author: co-author
  - Venue: common venues
  - Title: common keywords
- Superimpose the graph g₁ onto base graph B₁ to get a final graph representation G₁
  - Author: entire collaboration graph as B₁
  - Venue: entire venue similarity graph as B₁
  - Title: entire token co-occurrence graph B₁
- Measure the similarity of two entities e₁ and e₂ w.r.t. G₁ and G₂

Superimposition

- Overcome the limitation of existing distance metrics
- Unearth the hidden relationships in contents
- Use Quasi-Clique to measure the strong relations

Quasi-Clique

- Graph G
  - V(G): set of vertices
  - E(G): set of edges
  - Γ-quasi-complete-graph (0 ≤ Γ ≤ 1)
    - Every vertex in G has at least Γ * (|V(G)| - 1)
  - V(S) (⊆ V(G))
    - G(S): Γ-Quasi-Clique
      - if V(S) forms the graph satisfying Γ-quasi-complete-graph
    - G(S): Clique
      - if Γ = 1
- Use Quasi-Clique (QC) to measure contextual distances
  - E.g., Function QC(G(a), G(b), Γ=0.3, S=3)
### ACM Dataset

![ACM Dataset Graph](image)

**Precision:**
- k results are returned
- r of k are name variants
- precision = r / k

| JC | Jaccard similarity |
| QC | Quasi-Clique similarity |
| TI | TF-IDF Cosine similarity |
| IC | IntelliClean (venue hierarchy) |
| JC+QC | JC + Quasi-Clique |
| TI+QC | TI + Quasi-Clique |
| IC+QC | IC + Quasi-Clique |

### IMDB Synthetic Dataset

![IMDB Synthetic Dataset Graph](image)

### Outline
- Warm-Up
- Motivation & Problem Def.
- Disambiguation by Graphs
  - **Disambiguation by Groups**
  - Disambiguation by Googling
- Conclusion

### Key Idea
- Graph is a rich data structure
  - Can capture a wealth of information
  - But expensive to manipulate
- Better data structure than Graphs \(\Leftrightarrow\) Groups
  - When entities have a group of elements
  - Authors with citations, Images with \(m \times n\) grids

### Popular Group Similarity
- Jaccard
  \[ \text{sim}(g_1, g_2) = \frac{g_1 \cap g_2}{g_1 \cup g_2} \]
- Bipartite Matching
  - Cardinality
  - Weighted
- Clustering
  - Single vs. Complete vs. Average Link

### Intuition for better similarity
- Two groups are similar if:
  - There is high enough similarity between matching pairs of individual elements that constitute the two groups
  - A large fraction of elements in the two groups form matching element pairs
Group similarity

- Two groups of elements:
  - \( g_1 = \{r_{11}, r_{12}, \ldots, r_{1m_1}\} \)
  - \( g_2 = \{r_{21}, r_{22}, \ldots, r_{2m_2}\} \)
- The group measure \( BM \) is the normalized weight of the maximum bipartite matching \( M \) in the bipartite graph \((N = g_1 \cup g_2, E=g_1 \times g_2)\)

\[
BM_{\text{sim}, \rho}(g_1, g_2) = \frac{\sum_{r_{1j}, r_{2j}} (\text{sim}(r_{1j}, r_{2j}))}{m_1 + m_2 - |M|}
\]

such that

\[
\text{sim}(r_{1j}, r_{2j}) \geq \rho
\]

Challenges

- Large number of groups to match
  - \( O(NM) \)
- \( BM \) uses maximum weight bipartite matching
  - Bellman-Ford: \( O(V^2E) \)
  - Hungarian: \( O(V^3) \)

Solution: Greedy matching

- Bipartite matching computation is expensive because of the requirement
  - No node in the bipartite graph can have more than one edge incident on it
- Let’s relax this constraint:
  - For each element \( e_i \) in \( g_1 \), find an element \( e_j \) in \( g_2 \) with the highest element-level similarity \( \in S_1 \)
  - For each element \( e_i \) in \( g_2 \), find an element \( e_j \) in \( g_1 \) with the highest element-level similarity \( \in S_2 \)

Upper/Lower Bounds

\[
BM_{\text{sim}, \rho}(g_1, g_2) = \frac{\sum_{r_{1j}, r_{2j}} (\text{sim}(r_{1j}, r_{2j}))}{m_1 + m_2 - |M|}
\]

\[
UB_{\text{sim}, \rho}(g_1, g_2) = \frac{\sum_{r_{1j}, r_{2j}} (\text{sim}(r_{1j}, r_{2j}))}{m_1 + m_2 - (|S_1| + |S_2|)}
\]

\[
LB_{\text{sim}, \rho}(g_1, g_2) = \frac{\sum_{r_{1j}, r_{2j}} (\text{sim}(r_{1j}, r_{2j}))}{m_1 + m_2 - (|S_1| \cap |S_2|)}
\]

Theorem & Algorithm

- \( BM_{\text{sim}, \rho}(g_1, g_2) \leq UB_{\text{sim}, \rho}(g_1, g_2) \)
- **IF** \( UB(g_1, g_2) < \theta \rightarrow BM(g_1, g_2) < \theta \rightarrow g_1 \neq g_2 \)
- **ELSE IF** \( LB(g_1, g_2) \geq \theta \rightarrow BM(g_1, g_2) \geq \theta \rightarrow g_1 = g_2 \)
- **ELSE** compute \( BM(g_1, g_2) \)
  - This step is expensive

ACM Dataset

- Left: 300 groups
- Right: 700,000 groups
ACM Dataset

Left: 100 groups
Right: 700,000 groups

Outline
- Warm-Up
- Motivation & Problem Def.
- Disambiguation by Graphs
- Disambiguation by Groups
- Disambiguation by Googling
- Conclusion

Key Idea
- When entities have a wealth of information, we can exploit them by capturing them as either Graphs or Groups
- But when entities do not have a wealth of information or have only noisy information, then what to do?
- Ask people what they think

Hypothesis
- Use the Web as a collective knowledge of people
- Hypothesis:
  If an entity $e_1$ is a duplicate of another entity $e_2$, and if $e_1$ frequently appears together with information $I$ on the Web, then $e_2$ may appear frequently with $I$ on the Web, too.

Eg. ACM DL Case
- Search results from Google:
  - "Jeffrey D. Ullman" 384,000 pages 45%
  - "Jeffrey D. Ullman" + "aho" 174,000 pages 45%
  - "J. Ullman" 124,000 pages 33%
  - "J. Ullman" + "aho" 41,000 pages 33%
  - "Shimon Ullman" 27,300 pages 0%
  - "Shimon Ullman" + "aho" 66 pages 0%

Googled Name Linkage
Step 1. Select representative data

- **What to select**
  - A single token "aho"
  - A key phrase "stanford professor"
  - A sentence or more?

- **How to select**
  - tf, tf*idf, latent topic models, ...

- **How many to select**
  - 1, 2, ... n

- **Where to select from?**
  - Contents of canonical entity, variant, both

Step 2. Acquire the collective knowledge

- **How to form the query?**
  - Single information "I" (the most important data piece)
    - "J. D. Ullman" AND "Aho"
  - Multiple information "I1", "I2", "I3", ... (the most k important data pieces)
    - Conjunction or Disjunction or Hybrid
    - "J. D. Ullman" AND "Aho" AND/AND OR "database" AND/AND OR "vldb"...
  - Formal evaluation of the effectiveness of such variations
    - Different heuristics based on
      - Availability, discriminative power of the data content
      - Popularity of the name, variants, other candidates

Step 3. Interpret the collective knowledge

For entities e_c, e_i and information t_c

- **Page Counts**
  - Jeffrey D. Ullman
    - portal.acm.org
    - = 1/(174,000 - 41,000)
  - Shimon Ullman
    - portal.acm.org
    - = 1/(174,000 - 66)

- **URLs**
  - Jeffrey D. Ullman
    - portal.acm.org
    - = 1/(174,000 - 41,000)
  - Shimon Ullman
    - portal.acm.org
    - = 1/(174,000 - 66)

- **Web Page Contents**

  \[
  \text{sim}(e_c, e_i) = \text{doc\_sim}(\text{vdoc}(e_c), \text{vdoc}(e_i))
  \]

  **Document Similarity metrics:**

  \[
  \text{sim}(D_k, D_l) = \frac{\sum \text{doc}(D_k) \cdot \text{doc}(D_l)}{\sqrt{\sum (\text{doc}(D_k))^2 \cdot \sum (\text{doc}(D_l))^2}}
  \]

  \[
  \text{sim}_{\text{model}}(D_k, D_l) = \text{sim}(\theta_k, \theta_l)
  = -KL(\theta_k \| \theta_l) - \sum_k p(k|\theta_l) \log \frac{p(k|\theta_k)}{p(k|\theta_l)}
  \]

Results with URL and Host

**ACM data set:**

- 43 authors
- 14.2 citations/author
- 21 candidates/block
- 3.1 citations/candidate
- 1.8 name variants/block
- 6.7 citations/variant

**Recall:**

\[
\text{Recall} = \frac{\text{r}}{k}
\]

\[
\text{r} = \text{correct name variants}
\]

\[
\text{k} = \text{results are returned}
\]
Results with Web Pages

ACM data set:
- 43 authors
- 14.2 citations/author
- 21 candidates/block
- 3.1 citations/candidate
- 1.8 name variants/block
- 6.7 citations/variant

29% improvement

Results with Web Pages (cont)

IMDB data set:
- 50 actors
- 24 titles/entity
- 20 candidates/block
- 24 titles/candidate
- 1 name variant/block
- 23.5 titles/variant

193% improvement

Scalability

- Not scalable:
- A large number of Web accesses
- Network traffic, load of search engine and web sites

<table>
<thead>
<tr>
<th>Method</th>
<th>Time (in sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stanford WebBase Project</td>
<td>4.67</td>
</tr>
<tr>
<td>Google</td>
<td>234.96</td>
</tr>
<tr>
<td>Googling: long model 1 (without Google)</td>
<td>12.33</td>
</tr>
<tr>
<td>Googling: long model 2 (using Google) and the local snapshot</td>
<td>12.33</td>
</tr>
<tr>
<td>Googling: long model 2 using Google and the local snapshot</td>
<td>12.33</td>
</tr>
<tr>
<td>Downloaded the half of the data &amp; filtered</td>
<td>12.33</td>
</tr>
</tbody>
</table>

- Solutions:
  - Local snapshot of the Web
    - Stanford WebBase Project
    - ~100 million web pages from >50,000 sites including many .edu domains
    - Local snapshot containing 3.5 million relevant pages

Conclusion

- Name-related problems are common
- Three disambiguation techniques
  - By Graphs
  - By Groups
  - By Googling
  - Helps when entities
    - Have structures to exploit, or
    - Lack useful information

More research needed
- Inputs from AI, NLP, DB, DL

Task #13: Web People Search Task

http://nlp.uned.es/weps/

http://pike.psu.edu/

Thank You!