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

- 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

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

- Active Collaboration
Penn State University

- In 2005, IST hired a faculty from NUS
  - Dr. Heng Xu
- 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
  - "Googled 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 => talk

Credits

- Students
  - Ergin Elmacioglu (Penn State, USA)
  - Yee Tan Fan (NUS, Singapore)
  - Byung-Won On (Penn State, USA)
- Collaborators
  - C. Lee Giles (Penn State, USA)
  - Min-Yen Kan (NUS, Singapore)
  - Jaewoo Kang (Korea U., Korea)
  - Nick Koudas (U. Toronto, Canada)
  - Prasenjit Mitra (Penn State, USA)
  - 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.
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. 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

**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: \( \text{dist}(D(e1), D(e2)) = d \)
  - 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 $e_i$ as graph $g_1$ using common tokens
  - Author: co-author
  - Venue: common venues
  - Title: common keywords
- Superimpose the graph $g_1$ onto base graph $B_1$ to get a final graph representation $G_1$
  - Author: entire collaboration graph as $B_1$
  - Venue: entire venue similarity graph as $B_1$
  - Title: entire token co-occurrence graph $B_1$
- Measure the similarity of two entities $e_1$ and $e_2$ w.r.t. $G_1$ and $G_2$

**Superimposition**

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

**Quasi-Clique**

- Graph $G$
  - $V(G)$: set of vertices
  - $E(G)$: set of edges
  - $\Gamma$-quasi-complete-graph ($0<\Gamma\leq1$)
    - Every vertex in $G$ has at least $\Gamma$ degrees
  - $V(S) \subseteq V(G)$
    - $\Gamma$-Quasi-Clique
    - $\Gamma$-Clique
  - Use $\text{Quasi-Clique (QC)}$ to measure contextual distances
    - E.g., Function $\text{QC}(G(a), G(b), \Gamma=0.3, S=3)$
** Experimental Validation

<table>
<thead>
<tr>
<th>JC+QC</th>
<th>TC+QC</th>
<th>IC+QC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jaccard similarity</td>
<td>TF-IDF Cosine similarity</td>
<td>IntelliClean (venue hierarchy)</td>
</tr>
</tbody>
</table>

**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
  
  $sim(g_1, g_2) = \frac{|g_1 \cap g_2|}{|g_1 \cup g_2|}$

- 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 \( (N = g_1 \cup g_2, E = g_1 \times g_2) \)

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

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

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 \( \preceq S_1 \)
  - For each element \( e_i \) in \( g_2 \), find an element \( e_j \) in \( g_1 \) with the highest element-level similarity \( \preceq S_2 \)

Upper/Lower Bounds

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

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

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

Theorem & Algorithm

\[
BM_{\text{sim,}\rho}(g_1, g_2) \leq UB_{\text{sim,}\rho}(g_1, g_2)
\]

- IF \( UB(g_1, g_2) < \theta \rightarrow BM(g_1, g_2) < \theta \rightarrow g_1 \neq g_2 \)

\[
LB_{\text{sim,}\rho}(g_1, g_2) \leq BM_{\text{sim,}\rho}(g_1, g_2)
\]

- 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
    - \( BM(g, g) \geq \theta \)

Experiment

Left: 300 groups
Right: 700,000 groups
Experiment

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 e1 is a duplicate of another entity e2, and if e1 frequently appears together with information I on the Web, then e2 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 3%
  + "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 "I_1", "I_2", "I_3", … (the most k important data pieces)
    - Conjunction or Disjunction or Hybrid
      - "J. D. Ullman" AND "Aho" AND "database" AND "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 ec, ei and information tc
- Page Count
  - Jeffrey D. Ullman = 1/(174,000 - 41,000)
  - Shimon Ullman = 1/(174,000 - 66)
- URLs
  - Jeffrey D. Ullman = 3/16
  - portal.acm.org = 1/19
- Web Page Contents
  - Use top-k returned Web pages for each entity
  - Two alternatives for sim(ec, ei):
    - Group distance between two sets of top-k web pages
    - Represent each set by a single Virtual Document
    - Apply document comparison metrics on Virtual Doc.
      - Heuristics for creating Virtual documents:

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
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

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

Scalability

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

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

Conclusion

- 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!