Web Based Linkage

Ergin Elmacioglu
Min-Yen Kan
Dongwon Lee
Yi Zhang

1The Pennsylvania State University, USA
2National University of Singapore, Singapore
3University of California, Santa Cruz, USA

Motivation

Real world examples of the linkage problem

Name Linkage

Problem Definition:
The process of detecting and merging duplicate named entities that represent the same real-world object

Other real world examples:
Electronic devices (Apple iPod Nano vs. 4GB iPod nano)
Automobile models (Honda Fix vs. Honda Jazz)
Companies (T-Fal vs. Tefal)
Person names
- Customs (Jane Doe vs. Doe, Jane)
- Marriage (Carol Dusseau vs. Carol Arpaci-Dusseau)
- Misc. (Sean Engelson vs. Shlomo Argamon)

Name Linkage using Collective Knowledge

There are many effective solutions if enough and discriminative contents are available
- E.g., Contents based record linkage techniques

Becomes challenging when
- Not enough content to compare
- Content does not help to identify

Proposal: use external knowledge
- Ask people what they think
- Collective knowledge of people from the Web

Main Idea: Use the Web as a collective knowledge of people in solving the Name Linkage Problem

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.

Small Test

Search results from Google:
- "Jeffrey D. Ullman" 384,000 pages 10%
- "J. Ullman" 174,000 pages 10%
- "J. Ullman" 124,000 pages 10%
- "J. Ullman" 41,000 pages 10%
- "Shimon Ullman" 27,300 pages 0%
- "Shimon Ullman" + "aho" 66 pages 0%
Web Based Linkage: Overview

Step 1. Select Representative Data
- What to select
  - A single token "aho"
  - A key phrase "stanford professor"
  - A sentence or more?
- How to select
  - Assess importance
    - 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 Knowledge from Web
- How to form the query?
  - Single information "I" (the most important data piece)
    - "Jeffrey D. Ullman" AND "Aho"
  - Multiple information "I_{1, 2, 3, ...}" (the most k important data pieces)
    - Conjunction: "Jeffrey D. Ullman" AND "Aho" AND "database" AND "vldb"...
    - Disjunction: "Jeffrey D. Ullman" AND "Aho" OR "database" OR "vldb"...
    - Hybrid: "Jeffrey D. Ullman" AND "Aho" AND ("database" OR "vldb").

Step 3. Interpret the Collective Knowledge
- For entities e_c, e_i, and information t_c
  - Page Counts
  - URLs

Virtual Document Creation
- Web Page Contents
  - Use top-k returned Web pages for each entity
  - Represent each set by a Virtual Document
  - Some heuristics
    - D (m): Top m (≤ k) documents are concatenated
    - T (all, n): Top n tokens with the highest weight from all top-k web pages
    - Snippet (m): Snippets of top m (≤ k) web pages
    - Probabilistic Language Model: KL-divergence
      - \( \text{sim}(e_c, e_i) = \text{doc}_\text{sim}(\text{vdoc}(e_c), \text{vdoc}(e_i)) \)
Presented Next

Experimental Validation

- Tested against
  - ACM, ArXiv, IMDB
- Variations tested
  - Step 1: single token, top-k tokens
  - Step 2: conjunctive query only
  - Step 3
    - Page count, URL
    - Virtual Document: 10 heuristics and 2 language models
- Search Engines
  - Google, MS Live Search

Experimental Validation

ACM data set:
- 45 authors
- 14.2 citations/author
- 21 candidates/block
- 1.8 citations/candidate
- 6.7 citations/variant/block

Variations tested
- Step 1: single token, top-k tokens
- Step 2: conjunctive query only
- Step 3
  - Page count, URL
  - Virtual Document: 10 heuristics and 2 language models

Search Engines
- Google, MS Live Search


Experimental Validation

IMDB data set:
- 30 actors
- 24 titles/candidate
- 84% improvement

Variations tested
- Step 1: single token, top-k tokens
- Step 2: conjunctive query only
- Step 3
  - Page count, URL
  - Virtual Document: 10 heuristics and 2 language models

Search Engines
- Google, MS Live Search

Scalability

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

Related Work

- Abundant research on related problems
  - DB: approximate join, merge/purge, record linkage
  - DL: citation matching, author name disambiguation
  - AI: identity uncertainty
  - LIS: name authority control
- 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 < threshold, then e1 and e2 are matching entities
- Work well for common applications
- Ours performs better when
  - Entities lack useful information

Conclusion & Future Work

The Name Linkage Problem
- incomplete, noisy, non-descriptive data
- usage of the Web to get additional information for the linkage process

Future Work
- A formal framework
- Extension to “name disambiguation”
- Experimentation on more & (larger) data sets
- Scalability