Data Integration and Large Scale Analysis
05 Entity Linking and Deduplication

Shafaq Siddiqi

Graz University of Technology, Austria
Agenda

- Motivation and Terminology
- Entity Resolution Concepts
- Entity Resolution Tools
- Example Applications
Motivation and Terminology
Recap: Corrupted/Inconsistent Data

- **#1 Heterogeneity of Data Sources**
  - Update anomalies on denormalized data / eventual consistency
  - Changes of app/prep over time (US vs us) → inconsistencies

- **#2 Human Error**
  - Errors in semi-manual data collection, laziness (see default values), bias
  - Errors in data labeling (especially if large-scale: crowd workers / users)

- **#3 Measurement/Processing Errors**
  - Unreliable HW/SW and measurement equipment (e.g., batteries)
  - Harsh environments (temperature, movement) → aging

---

**Motivation and Terminology**

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>BDay</th>
<th>Age</th>
<th>Sex</th>
<th>Phone</th>
<th>Zip</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Smith, Jane</td>
<td>05/06/1975</td>
<td>44</td>
<td>F</td>
<td>999-9999</td>
<td>98120</td>
</tr>
<tr>
<td>3</td>
<td>John Smith</td>
<td>38/12/1963</td>
<td>55</td>
<td>M</td>
<td>867-4511</td>
<td>11111</td>
</tr>
<tr>
<td>7</td>
<td>Jane Smith</td>
<td>05/06/1975</td>
<td>24</td>
<td>F</td>
<td>567-3211</td>
<td>98120</td>
</tr>
</tbody>
</table>

**Uniqueness & duplicates**

**Contradictions & wrong values**

**Missing Values**

**Ref. Integrity**

**[Credit: Felix Naumann]**

- **Typos**
  - 98120  →  98120
  - 90001  →  Lost Angeles

- **No Global Keys**
Terminology

- **Entity Linking**
  - “*Entity linking* is the problem of creating links among records representing real-world entities that are related in certain ways.”
  - “As an important special case, it includes *entity resolution*, which is the problem of identifying or linking duplicate entities.”

- **Other Terminology**
  - Entity Linking $\rightarrow$ Entity Linkage, Record Linkage
  - Entity Resolution $\rightarrow$ Data Deduplication, Entity Matching

- **Applications**
  - Named entity recognition and disambiguation
  - Archiving, knowledge bases and graphs
  - Recommenders / social networks
  - Financial institutions (persons and legal entities)
  - Travel agencies, transportation, health care

---

Barack Obama
Barack Hussein Obama II
The US president (2016)

Barack and Michelle
are married ....
Entity Resolution Concepts


[Sairam Gurajada, Lucian Popa, Kun Qian, Prithviraj Sen: Learning-Based Methods with Human in the Loop for Entity Resolution, Tutorial, CIKM 2019]

[Felix Naumann, Ahmad Samiei, John Koumarelas: Master project seminar for Distributed Duplicate Detection. Seminar, HPI WS 2016]
Problem Formulation

- **Entity Resolution**
  - “Recognizing those records in two files which represent identical persons, objects, or events”
  - Given two data sets A and B
  - Decide for all pairs of records $a_i - b_j$ in $A \times B$
    - if match (link), no match (non-link), or not enough evidence (possible-link)

- **Naïve Deduplication**
  - UNION DISTINCT via hash group-by or sort group-by
  - **Problem**: only exact matches

- **Similarity Measures**
  - Token-based: e.g., Jaccard $J(A,B) = \frac{(A \cap B)}{(A \cup B)}$
  - Edit-based: e.g., Levenshtein $\text{lev}(A,B) \rightarrow \min(\text{replace, insert, delete})$
  - Phonetic similarity (e.g., soundex, metaphone), **Python lib Jellyfish**

---


<table>
<thead>
<tr>
<th>Name</th>
<th>Position</th>
<th>Affiliation</th>
<th>Department</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shafaq Siddiqui</td>
<td>Lecturer</td>
<td>Sukkur IBA</td>
<td>CS</td>
</tr>
<tr>
<td>Shafaq Siddiqi</td>
<td>TA</td>
<td>TU Graz</td>
<td>CSBME</td>
</tr>
</tbody>
</table>
Entity Resolution Concepts

Entity Resolution Pipeline

Prepare Data → Blocking/Sorting → Matching → Clustering

- A1, A2, B1, B2, C1, C2, D1, B3
- A1, A2, C1, C2, D1, B1, B3, B2
- B1, B3, B2
- A, r1, r4
- C, r2, r7
- D, r3
- B, r5, r6, r8
Entity Resolution Concepts

Entity Linking Approaches

Entity Linking Approaches

50 Years of Entity Linkage

Rule-based and stats-based
- Blocking: e.g., same name
- Matching: e.g., avg similarity of attribute values
- Clustering: e.g., transitive closure, etc.

1969 (Pre-ML)

Supervised learning
- Random forest for matching
  F-msr: >95% w. ~1M labels
- Active learning for blocking & matching
  F-msr: 80%-98% w. ~1000 labels

2018 (Deep ML)

Sup / Unsup learning
- Matching: Decision tree, SVM
  F-msr: 70%-90% w. 500 labels
- Clustering: Correlation clustering, Markov clustering

~2000 (Early ML)

~2015 (ML)

[Shafaq Siddiqi, Graz University of Technology, WS 2022/23]

[Xin Luna Dong, Theodoros Rekatsinas: Data Integration and Machine Learning: A Natural Synergy. PVLDB 2018]
Step 1: Data Preparation

- **#1 Schema Matching and Mapping**
  - See lecture 04 Schema Matching and Mapping
  - Create **homogeneous schema** for comparison
  - Split composite attributes

- **#2 Normalization**
  - Removal of special characters and white spaces
  - **Stemming**
  - **Capitalization** (to upper/lower)
  - Remove redundant works, resolve abbreviations

- **#3 Data Cleaning**
  - See lecture 06 Data Cleaning and Data Fusion
  - Correct data corruption and inconsistencies
Step 2: Blocking and Sorting

- #1 Naïve All-Pairs
  - Brute-force, naïve approach
  - $n \cdot (n-1)/2$ pairs $\to O(n^2)$ complexity

- #2 Blocking / Partitioning
  - Efficiently create small blocks of similar records for pair-wise matching
  - **Basic**: equivalent values on selected attributes (name)
  - **Predicates**: whole field, token field, common integer, same x char start, n-grams
  - **Hybrid**: disjunctions/conjunctions
  - Blocking Keys: 
    - John Roberts 20 Main St Plainville MA 01111 $\to$ JR01111
    - Learned: Minimal rule set via greedy algorithms
  - Significant reduction: 1M records $\to$ 1T pairs
    - 1K partitions w/ 1K records $\to$ 1G pairs ($1000x$)

[Building a Scalable Record Linkage System with Spark, Xinlin Li, Spark+AI Summit 2018]
Step 2: Blocking, cont.

- #3 Sorted Neighborhood
  - Define **sorting keys** (similar to blocking keys)
  - Sort records by sorting keys
  - Define **sliding window of size** \( m \) (e.g., 100) and compute all-pair matching within sliding window

- #4 Blocking via Word Embeddings and LSH/DL
  - Compute word/attribute embeddings + tuple embeddings
  - **Locality-Sensitive Hashing (LSH)** for blocking
  - \( K \) hash functions \( h(t) \rightarrow k \)-dim hash-code
  - \( L \) hash tables, each \( k \) hash functions

\[
\begin{align*}
V \%*\% H & \quad h_1=[-1, 1,1] , \ h_2=[ 1,1, 1] , \\
& \quad h_3=[-1,-1,1] , \ h_4=[-1,1,-1] , \\
\end{align*}
\]

\[
\begin{align*}
v[t_1]= & [0.45,0.8,0.85] & [1.2,2.1,-0.4,-0.5] & [1,1,-1,-1] & [12] \text{ Hash} \\
v[t_2]= & [0.4,0.85,0.75] & [1.2,2.0,-0.5,-0.3] & [1,1,-1,-1] & [12] \text{ bucket}
\end{align*}
\]
Step 3: Matching

- **#1 Basic Similarity Measures**
  - Pick similarity measure \( \text{sim}(r, r') \) and thresholds: high \( \theta_h \) (and low \( \theta_l \))
  - Record similarity: avg attribute similarity
  - **Match:** \( \text{sim}(r, r') > \theta_h \)  **Non-match:** \( \text{sim}(r, r') < \theta_l \)
  - **Possible match:** \( \theta_l < \text{sim}(r, r') < \theta_h \)

- **#2 Learned Matchers (Traditional ML)**
  - **Phase 1:** Model Generation
  - **Phase 2:** Model Application
  - Selection of samples for labeling (sufficient, suitable, **balanced**)
  - **SVM** and **decision trees**, **logistic regression**, **random forest**, **XGBoost**

---

Entity Resolution Concepts

- **[Mikhail Bilenko, Raymond J. Mooney: Adaptive duplicate detection using learnable string similarity measures. KDD 2003]**
- **[Hanna Köpcke, Andreas Thor, Erhard Rahm: Evaluation of entity resolution approaches on real-world match problems. PVLDB 2010]**
- **[Xin Luna Dong: Building a Broad Knowledge Graph for Products. ICDE 2019]**
Step 3: Matching, cont.

- **Deep Learning for ER**
  - Automatic **representation learning** from text (avoid feature engineering)
  - Leverage pre-trained **word embeddings for semantics** (no syntactic limitations)

- **Example DeepER**
  - [Muhammad Ebraheem et al: Distributed Representations of Tuples for Entity Resolution. *PVLDB 2018*]

- **Example Magellan**
  - **DL for text and dirty data**
Step 3: Matching, cont.

- **Labeled Data**
  - Scarce (experts)
  - Class skew

- **Transfer Learning**
  - Learn model from high-resource ER scenario (w/ regularization)
  - Fine-tune using low-resource examples

- **Active Learning**
  - Select instances for tuning to min labeling

---

[Sairam Gurajada, Lucian Popa, Kun Qian, Prithviraj Sen: Learning-Based Methods with Human in the Loop for Entity Resolution, Tutorial, CIKM 2019]

\[ F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \]
Step 4: Clustering

- Recap: Connected Components
  - Determine connected components of a graph (subgraphs of connected nodes)
  - Propagate max(current, msgs) if ! = current to neighbors, terminate if no msgs

- Clustering Approaches
  - Basic: connected components (transitive closure) w/ edges sim > \( \theta_h \)
    - Issues: big clusters and dissimilar records
  - Correlation clustering: +/- cuts based on sims → global opt NP-hard
  - Markov clustering: stochastic flow simulation via random walks

Incremental Data Deduplication

- **Goals**
  - Incremental stream of updates
    → previously **computed results obsolete**
  - Same or **similar results AND significantly faster** than batch computation

- **Approach**
  - End-to-end incremental record linkage for new and changing records
  - Incremental maintenance of similarity graph and incremental graph clustering
  - Initial graph created by **correlation clustering**
  - Greedy update approach in polynomial time
    - Directly connect components from increment ΔG into Q
    - **Merge** of **pairs of clusters** to obtain better result?
    - **Split** of **cluster into two** to obtain better result?
    - **Move** nodes **between two clusters** to obtain better result?
Entity Resolution Tools
Python Dedupe

- **Overview**
  - *Python library for data deduplication* (entity resolution)
  - *By default:* logistic regression matching (and blocking)

- **Example**

  ```python
  fields = [
    {'field':'Site name', 'type':'String'},
    {'field':'Address', 'type':'String']
  deduper = dedupe.Dedupe(fields)

  # sample data and active learning
  deduper.sample(data, 15000)
  dedupe.consoleLabel(deduper)

  # learn blocking rules and pairwise classifier
  deduper.train()

  # Obtain clusters as lists of (RIDs and confidence)
  threshold = deduper.threshold(data, recall_weight=1)
  clustered_dupes = deduper.match(data, threshold)
  ```

  Do these records refer to the same thing? 
  (y)es / (n)o / (u)nsure / (f)inished
Magellan (UW-Madison)

- **System Architecture**
  - How-to guides for users
  - Tools for individual steps of entire ER pipeline
  - Build on top of existing Python/big data stack
  - Scripting environment for power users

Entity Resolution Tools


Facilities for Lay Users
- GUIs, wizards, …

Power Users
- Development Stage
  - Supporting tools (as Python commands)
  - Data samples
- Production Stage
  - Supporting tools (as Python commands)
  - Original data

EM Workflow

Python Interactive Environment
- Script Language

Data Analysis Stack
- pandas, scikit-learn, matplotlib, …

Big Data Stack
- PySpark, mrjob, Pydoop, …

[Yash Govind et al: Entity Matching Meets Data Science: A Progress Report from the Magellan Project. SIGMOD 2019]
SystemER (IBM Research – Almaden)

Learns explainable ER rules (in HIL)

- support csv and newline delimited json
- can start without labeled data
- an auxiliary tool for learning complex matching functions
- effective active learning algorithm
- scale up with distributing computing
- produce explainable ER models

DBLP.title = ACM.title
AND DBLP.year = ACM.year
AND jaccardSim(DBLP.authors, ACM.authors) > 0.1
AND jaccardSim(DBLP.venue, ACM.venue) > 0.1
→ SamePaper(DBLP.id, ACM.id)


[Mauricio A. Hernández, Georgia Koutrika, Rajasekar Krishnamurthy, Lucian Popa, Ryan Wisnesky: HIL: a high-level scripting language for entity integration. EDBT 2013]
BEER (Blocking for Effective Entity Resolution)

[Sainyam Galhotra, Donatella Firmani, Barna Saha, and Divesh Srivastava: BEER: Blocking for Effective Entity Resolution, SIGMOD 2021]

Feedback after 1% sample
**BrewER (Entity Resolution On-Demand)**

(a) The traditional pipeline: the data scientist specifies how to clean the data with ER; once cleaned, she issues the query.

(b) The ER-on-demand pipeline: the data scientist specifies how to clean the data within the query.

---

**Figure 2: Query syntax in BrewER.**

```
SELECT [TOP k] (αJ(Aj))
FROM D
WHERE φ
GROUP BY ENTITY WITH MATCHER μ
[HAVING (αJ(Aj) LIKE |IN| |≤| |> | |≥| |=| const)]
[ORDER BY αJ(Aj) [ASC | DESC]]
```
Example Applications
DIA Exercise

- **Task: Distributed Entity Resolution on Apache Spark**
  - 1-3 person teams, data: Uni Leipzig Benchmarks

- **Example 1: DBLP, ACM, Google Scholar Publications**
  - (title, authors, venue, year)
  - Basic preprocessing via title capitalization, etc
  - How about leveraging the linked PDF papers?

- **Example 2: Amazon, Google Products**
  - (name, description, manufacturer, price)
  - NLP for matching medium and long descriptions, e.g., word embeddings
  - How about leveraging the product images (different angles)

In practice:
- multi-modal data, and
- feature engineering

https://dbs.uni-leipzig.de/research/projects/object_matching/benchmark_datasets_for_entity_resolution
Data Management – Autograding

- Plagiarism Detection via Entity Resolution
  - [https://issues.apache.org/jira/browse/SYSTEMDS-3191](https://issues.apache.org/jira/browse/SYSTEMDS-3191) (DIA WS21/22)
  - Data preparation: file names/properties, runtime, correctness
  - Blocking: by programming language, results sets
  - Matching
    - Exact matches via basic diff + threshold
    - Code similarity via SotA embeddings
  - Clustering
    - Connected components within each block (min sim threshold)

[Example Applications]
Summary and Q&A

- Motivation and Terminology
- Entity Resolution Concepts
- Entity Resolution Tools
- Example Applications

Fundamental Data Integration Technique, w/ lots of applications + remaining challenges

Next Lectures (Data Integration Architectures)
- 06 Data Cleaning and Data Fusion [Nov 11]