Data Integration and Analysis
05 Entity Linking and Deduplication

Matthias Boehm

Graz University of Technology, Austria
Computer Science and Biomedical Engineering
Institute of Interactive Systems and Data Science
BMVIT endowed chair for Data Management

Last update: Nov 08, 2019
Announcements/Org

- **#1 Video Recording**
  - Link in TeachCenter & TUbe (lectures will be public)

- **#2 Coding Contest**
  - IT Community Styria online or in-person
  - Inffeldgasse 25/D, HS i3, **Nov 08, 3pm**

- **#3 Kafka Meetup Graz**
  - **Nov 27, 5.45pm - 9pm**, NETCONOMY
  - [https://www.meetup.com/de-DE/Graz-Kafka/events/265837901/](https://www.meetup.com/de-DE/Graz-Kafka/events/265837901/)

- **#4 Apache Spark 3.0**
  - **Nov 07**: Spark 3.0 preview announcement
Agenda

- Motivation and Terminology
- Entity Resolution Concepts
- Entity Resolution Tools
- Projects and Exercises
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

**Uniqueness & duplicates**

**Contradictions & wrong values**

**Missing Values**

**Ref. Integrity**

<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>

**Zip City**

- 98120 San Jose
- 90001 Lost Angeles

**Typos**

[Credit: Felix Naumann]
Motivation and Terminology

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 → Entity Linkage, Record Linkage
  - Entity Resolution → 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


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

---

Entity Resolution Concepts

Entity Resolution Pipeline

Entity Resolution Concepts

Prepare Data

Blocking/Sorting

Matching

Clustering

A1, C1, D1
A2
A1
C1
C2
D1
B1
B3
B2

A r1, r4
C r2, r7
D r3
B r5, r6, r8
Entity Resolution Concepts

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.

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

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

- **Deep learning**
  - Deep learning
  - Entity embedding

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

- **#1 Naïve All-Pairs**
  - Brute-force, naïve approach
  - \( n*(n-1)/2 \) pairs \( \rightarrow O(n^2) \) 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: \( \rightarrow JR01111 \)
    - **Learned**: Minimal rule set via greedy algorithms
    - **Significant reduction**: 1M records \( \rightarrow \) 1T pairs
      - 1K partitions w/ 1K records \( \rightarrow \) 1G pairs (1000x)

---

[ Nicholas Chammas, Eddie Pantrige: Building a Scalable Record Linkage System, Spark+AI Summit 2018 ]
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**
  - Compute word/attribute embeddings + tuple embeddings
  - **Locality-Sensitive Hashing (LSH)** for blocking
  - K hash functions h(t) → k-dimensional hash-code
  - L hash tables, each k hash functions

\[
X \%*\% Y
\]

\[
\begin{align*}
v[t1] &= [0.45, 0.8, 0.85] \\
v[t2] &= [0.4, 0.85, 0.75]
\end{align*}
\]

\[
\begin{align*}
h1 &= [-1, 1, 1] \\
h2 &= [1, 1, 1] \\
h3 &= [-1, -1, 1] \\
h4 &= [-1, 1, -1]
\end{align*}
\]

\[
\begin{align*}
[1.2, 2.1, -0.4, -0.5] &\rightarrow [1, 1, -1, -1] \\
[1.2, 2.0, -0.5, -0.3] &\rightarrow [1, 1, -1, -1]
\end{align*}
\]

[Muhammad Ebraheem et al: Distributed Representations of Tuples for Entity Resolution. PVLDB 2018]
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**: Learned string similarity measures for selected attributes
- **Phase 2**: Training matching decisions from similarity metrics
- Selection of samples for labeling (sufficient, suitable, balanced)
- **SVM** and decision trees, logistic regression, random forest, XGBoost

References:
- [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]
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**
  - Text and dirty data
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

\[ F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \]

---

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

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?

[Anja Gruenheid, Xin Luna Dong, Divesh Srivastava: Incremental Record Linkage. PVLDB 2014]
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'}
]

# 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


SystemER (IBM Almaden – Research)

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)

Learns explainable ER rules (in HIL)


[Mauricio A. Hernández, Georgia Koutrika, Rajasekar Krishnamurthy, Lucian Popa, Ryan Wisnesky: HIL: a high-level scripting language for entity integration. EDBT 2013]
Projects and Exercises
Exercise: Distributed Data Deduplication

- **Two-Part DIA Exercise**
  - **Topic**: Distributed Duplicate Detection on publication dataset
  - **Part 1**: Entity resolution primitives (prep, blocking, matching, clustering)
  - **Part 2**: Scalable implementation in Apache Spark
  - Combines various aspects of entire DIA course (part A and B)
  - Example related work:

[Xu Chu, Ihab F. Ilyas, Paraschos Koutris: Distributed Data Deduplication. PVLDB 2016]

- **Administrative Notes**
  - Alternative to programming projects in SystemDS (2 ECTS → 50 hours)
    - **pro**: work independently, many topics, **con**: impact, no review
  - No teams, individual assignment
  - Students: **Julian Holzegger**, TBD
  - **Deadline: Jan 31**, submitted in TeachCenter
Projects – Scripts, Algorithms, Language APIs

- **#1 Scripts for Cloud Deployment** (AWS EMR, Azure HDInsight) → Florijan Klezin
- **#2 2x Python Language Bindings** (lazy eval, builtins, packaging)
- **#3 Bayesian Optimization for Hyper-Parameter Optimization**
- **#4 Stable Marriage Algorithms in Linear Algebra** → Thomas Wedenig
- **#5 XSLT or JSON mapping UDFs** (local, distributed)
- **#6 Large-Scale Slice Finding for ML Model Debugging** → Svetlana Sagadeeva
Projects – Data Cleaning and Augmentation

- 7 Hidden Markov Models for Missing Value Imputation NLP → Afan Secic
- #8 Missing Value Imputation for Continuous/Categorical Columns
- #9 Time Series Outlier Removal and Preprocessing
- #10 Reconstruction of Aggregated Time Series
- #11 Data Augmentation for ML-based Data Cleaning (data corruption)
Projects – Schema Detection and Data Prep

- #12 Inclusion and Functional Dependency Discovery (local and distributed)
- #13 Schema Detection from JSON and XML
- #14 Semantic Schema Detection (see Sherlock)
- #15 Feature Transform: Feature Hashing (local, distributed)
- #16 Feature Transform: Equi-Height/Custom Binning (local, distributed)
Projects – Compiler and Runtime

- #17 Consolidated Cost Model for HOPs and Instructions (for lineage)
- #18 4x Basic Distributed Tensor Operations (distributed, federated) → Kevin Innerebner / Valentin Leutgeb
- #19 Basic Sparse Tensor Representations (homogeneous/heterogeneous)
- #20 JSON/JSONL reader/writer into Data Tensor (local, distributed) → Lukas Erlbacher
- #21 Protobuf reader/writer into Data Tensor (local, distributed)
- #22 Lineage Tracing for Spark Operations (ops and parfor loops) → Benjamin Rath
- #23 Lineage Trace Difference Detection (incl deduplicated items)
Summary and Q&A

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

- Projects and Exercises
  - **Nov 08**: project/exercise selection
  - **Nov 14**: grace period ends
  (after that all unassigned students removed from course)

- Next Lectures (Data Integration and Preparation)
  - **06 Data Cleaning and Data Fusion** [Nov 15]
  - **07 Data Provenance and Blockchain** [Nov 22] → potential move to **Nov 29**

---

SystemDS: A Declarative Machine Learning System for the End-to-End Data Science Lifecycle

Matthias Boehm, Julian Antonov, Sebastian Baumgaertner, Mark Dokter, Robert Gantner, Kevin Innerebner, Florian Klaizin, Stefanie Lindstaedt, Armin Phani, Benjamin Rath, Berthold Reinhwald, Shafeq Siddiqi

1 Graz University of Technology, Graz, Austria
2 Know-Center GmbH, Graz, Austria
3 IBM Research – Almaden, San Jose, CA, USA

CIDR’20

Students who contribute to SystemDS by Dec 16 are included in acknowledgements

706.520 Data Integration and Large-Scale Analysis – 05 Entity Linking and Deduplication
Matthias Boehm, Graz University of Technology, WS 2019/20