POZNAN UNIVERSITY OF TECHNOLOGY

DW Loading and Refreshing Techniques: ETL

Deduplication

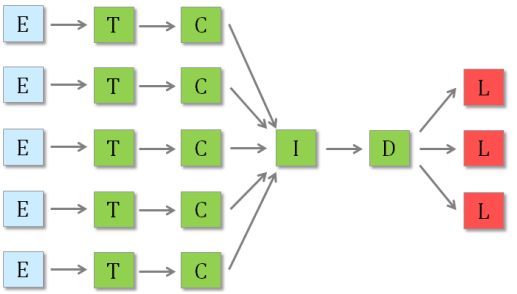
Robert Wrembel
Poznan University of Technology
Institute of Computing Science
Poznań, Poland
Robert.Wrembel@cs.put.poznan.pl
www.cs.put.poznan.pl/rwrembel

Outline

- Standard deduplication workflow
- Deduplication techniques

Deduplication

- Removing duplicate data from an integrated dataset



Data deduplication

- Deduplication = entity matching = duplicate identification = record linkage = entity resolution = reference reconciliation
- Entity Resolution (ER) aims to identify different descriptions (entities, records, objects, data instances) that refer to the same real-world entity
  - appearing either within the same or different data sources
  - when unique entity identifiers are not available
  - https://blog.acolyer.org/2020/12/14/entity-resolution/
- No single best algorithm
  - dedicated algorithms for different domains
  - dedicated algorithms for different data types

Data deduplication

- Simpler case: comparing relational records
- More complex case: comparing complex objects (with nested components)
  - XML data
  - OO data
- Special case
  - writing Arabic names in Latin alphabet

http://www.cjk.org/data/arabic/proper/database-arabic-names/

Me'ezer
Meezer
Ma'aser
Maaser
Ma'eser
Maeser
Me'eser
Meeser
Ma'asser
Maasser
Me'ether
Meether

Data deduplication

- Before comparing, records should be cleaned
  - homogenizing values (abbreviations, units of measurement, symbols, ...)
  - no special signs, no punctuations
  - no abbreviations
- Problem: how to decide if 2 records represent the same entity?
  - [Wrembel, Robert, ul. Matejki, Poznań]
  - [Wrębel, Robert, ul. Matejki, Poznań]
- Case 1: natural identifiers (e.g., ID, SSN, PESEL, email, mobile#) available
  - but email, mobile# may change in time for the same person
- Case 2: no natural identifiers available
  - approximate/probabilistic decision based on a **similarity measure**

Inconsistent naming

B. CHROBREGO	BOLERSŁAWA CHROBREGO
B CHROBREGO	BOLESLAWA CHROBREGO
B CHROBREGO 10	BOLESŁ. CHROBREGO
B CHROBREGO 42B	BOLESŁAW CHROBREGO
B,CHROBREGO	BOLESŁAWA CHROBREGO
B.CHROBREGO	BOLESŁAWA CHROBREGO
B.CHROBREGO 33/72	BOLESŁ.CHROBREGO
B.CHROBREGO OŁOBOK	BOLESŁWA CHROBREGO
B.CHROBREGO SKR.60	BOOESŁAWA CHROBREGO
BOL. CHROBREGO	CHROBREGO
BOL CHROBREGO	CHROBREGO 10
.BOL.CHROBREGO	CHROBREGO 22A/6
BOL.CHROBREGO	CHROBREGO KOWALEW
BOL.CHROBREGO OŁOBOK	CHROBREGO PAŃSTWOWY DOM D

Data deduplication

- **How to compare records?**
  - **the worst case: each entity compared to all the other entities**
    - **$O(n^2)$ → deduplication**
    - **O(n*m) → record linkeage**
  - **problem: efficiency**

Data deduplication

- Improvement 1: hashing
  - on some attributes (hash keys) → O(n)
  - drawback: records must have identical values of hash keys to hash into the same bucket → applicable for exact match

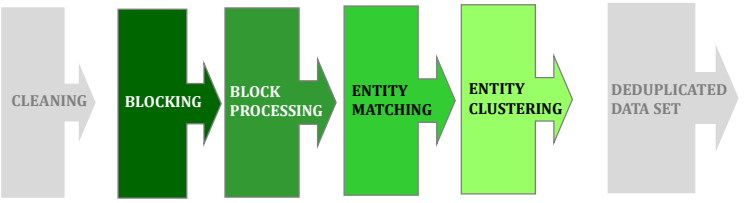
Data deduplication

- Improvement 2: sorted neighbourhood
  - sort records by a given attribute (list of attributes)
  - assumption: more similar records are located closer to each other
  - compare records only within a moving window of N records
  - within the window all with all compared
  - open issue: the size of the window

lenovo X1 carbon	lenovo X1 carbon	lenovo X1 carbon
lenovo X1 Carbon	lenovo X1 Carbon	lenovo X1 Carbon
Lenovo X1 carbon	Lenovo X1 carbon	Lenovo X1 carbon
Lenovo X1 Carbon	Lenovo X1 Carbon	Lenovo X1 Carbon
Lenovo X1	Lenovo X1	Lenovo X1
Lenovo x1	Lenovo x1	Lenovo x1

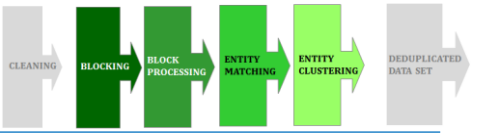
Data deduplication

- Standard data deduplication workflow



Blocking

- Based on some attributes (blocking keys) assigns an entity to a block (group)
  - similar entities reside in the same block
- Types
  - disjoint blocking: a record is included in only one block
  - overlapping blocking: a record may be included in multiple blocks → probability of finding a better match is higher at a cost of more record comparisons
- Goal: to reduce the number of entity comparisons → records are compared in the same block
- Challenge → selecting a blocking key
  - manual selection
  - supervised learning
- 14 different algorithms

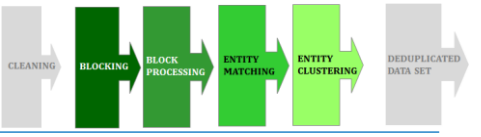


Block processing

- Goal: to further reduce the number of entity comparisons
- Method: eliminating redundant and unnecessary comparisons within blocks
- 18 different algorithms

Block processing

- For overlapping blocking
- Block pruning
  - ordering blocks from the smallest to the largest
    - larger blocks contain more unrelated records
  - discarding blocks whose cost of identifying new matches exceeds a threshold
- Size-based Block Clustering
  - merging small blocks that correspond to similar blocking keys
  - and splitting large blocks into smaller ones
  - to balance block sizes → balancing parallel processing of record matching in blocks



Block processing

- Iterative blocking
  - whenever a new pair of duplicates is detected (r_i,r_m) their descriptions are merged → r_{im}
  - r_{im} replaces (r_i,r_m) in all blocks
- Meta-blocking
  - uses a graph to represent comparisons
  - eliminates the same comparisons in multiple blocks
  - uses labels of graph edges to eliminate comparisons below certain threshold
    - methods for computing values of labels

Meta-blocking: example

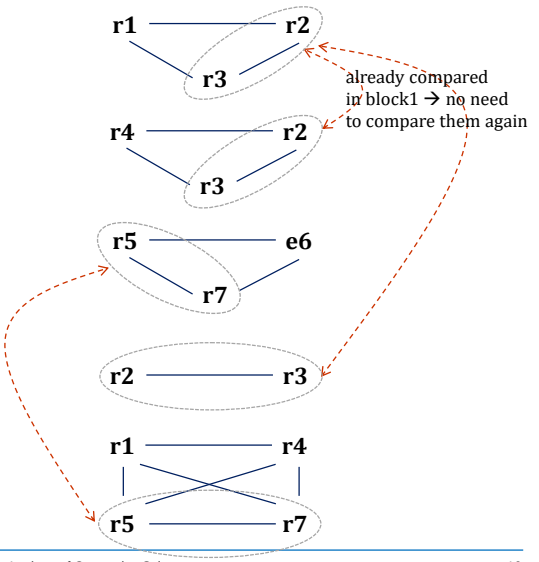
block1: {r1 r2 r3}

block2: {r2 r3 r4}

block3: {r5 r6 r7}

block4: {r2 r3 }

block5: {r1 r4 r5 r7}



Meta-blocking: example

block1: {r1 r2 r3}

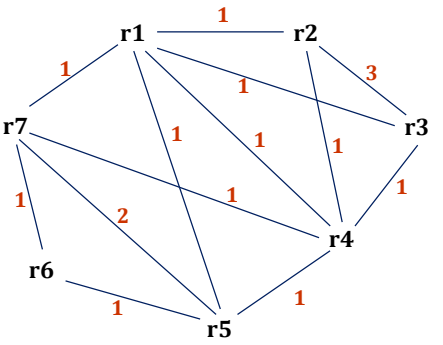
block2: {r2 r3 r4}

block3: {r5 r6 r7}

block4: {r2 r3 }

block5: {r1 r4 r5 r7}

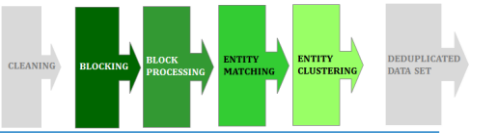
edge weight: #blocks where e_i and e_m are compared



trade off: lower number of entity comparisons at a cost of **lower recall**
recall: TP/(TP+FN)
FN: entities that represent the same object but were not discovered as being the same

Entity matching

- **Goal: to determine whether compared entities refer to the same real-world object**
- **Method: applying a similarity function**
- **20 different algorithms**



Entity matching

- Matching uses similarity function $sim(r_i, r_j)$ that mapps each pair of records (r_i, r_j) to a similarity value
  - φ measures how similar r_i and r_j are
- Matching (variant 1)
  - matching: $sim(r_i, r_j) >= v$
  - not matching $sim(r_i, r_j) < v$
- Matching (variant 2)
  - not matching: $sim(r_i, r_j) < v_1$
  - unknown: $v_1 <= sim(r_i, r_j) < v_2$
  - matching: $sim(r_i, r_j) >= v_2$

Entity matching

- Similarity measure
  - simple → based on a single attribute (key) values
    - e.g., Jaccard, Levenshtein
  - complex → (weighted) combination of similarity measures on multiple attributes of r_i and r_j
  - context-based (semantic relationships)
    - healthcare: data source S1: entity Patient
    - healthcare: data source S2: entity Person
    - banking: data source S3: entity Customer
    - Patient similar to Person since both exist in the same context
    - similarity represented as a graph with weighted arcs
  - hybrid: based on multiple similarity measures
    - e.g., complex + context-based (can also be weighted)

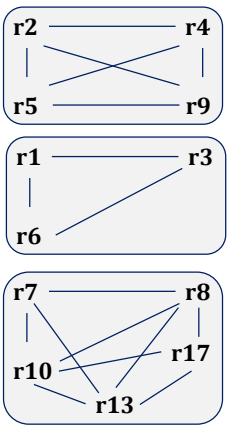
Entity matching

- Soundex similarity
  - returns a code of pronunciation of an input
  - soundex('Smith')=soundex('Smit')=S530
- Levenhstein (edit distance) similarity
  - minimum number of inserts and deletes (updates) of characters in order to convert L1 to L2
  - L1 and L2 identical: distance=0
  - ABC → ABCDEF: distance=3
  - DEFCAB → ABC: distance=5

Entity clustering

- Creating clusters of entities → all entities in a given cluster correspond to the same real-world entity, with a given high probability (similarity measure)
- Multiple clustering algorithms
  - use the similarity measure between records

Entity clustering



Entity clustering

➲ **Merge semantically identical records in a cluster into one final augmented record**

TaxpayerID	Fname	Lname	Country	Title	Domain
132018	Robert	Wrembel	Poland	Prof.	data wrehouses

+

TaxpayerID	1sName	2ndName	Last_Name	Education	City	Born
132018	Robert	Andrew	Wrembel	univ.	Poznań	1968

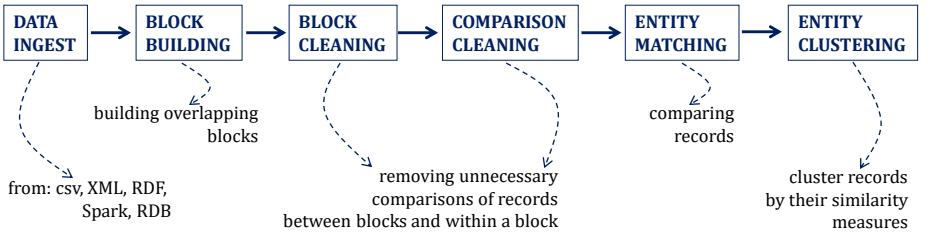
TaxpayerID	1sName	2ndName	Last_Name	Education	City	Born	TaxpayerID	Country	Title	Domain
132018	Robert	Andrew	Wrembel	univ.	Poznań	1968	132018	Poland	Prof.	data wrehouses

JedAI

- **Opensource library (+GUI) including some algorithms used in the entity resolution pipeline**
- **http://jedai.scify.org**

> G. Papadakis, L. Tsekouras, E. Thanos, G. Giannakopoulos, T. Palpanas, M. Koubarakis: JedAI: The Force behind Entity Resolution. ESWC 2017

- **JedAI pipeline**



JedAI

- **Available methods in the pieline**