Multi-Source Data Matching and Clustering
Erhard Rahm

German AI Centers

5 new, permanent German AI centers (in addition to DFKI):

• Berlin (BIFOLD)
• Dortmund / Bonn (ML2R)
• Dresden / Leipzig (ScaDS.AI)
• München (MCML)
• Tübingen (tuebingen.ai)

www.humboldt-foundation.de
**ScaDS.AI**

- **ScaDS.AI**: Center for Scalable Data Analytics and Artificial Intelligence
- extends previous Big Data center ScaDS Dresden/Leipzig (est. 2014)
- since 2019: AI / Data Science center ScaDS.AI
- July 2022: institutional funding starts
  - co-financed by BMBF and state of Saxony

---

**ScaDS.AI: Overall structure**
Research Areas

Data Integration

Provision of uniform access to data originating from multiple, autonomous sources

Physical data integration
• original data is combined within a new dataset / database for access and analysis
• approach of data warehouses, knowledge graphs and most Big Data applications

Virtual data integration
• data is accessed on demand in their original data sources, e.g. based on an additional query layer
• approach of federated databases and linked data
2 Levels of data integration

Metadata (schema/ontology) level
- **Schema Matching**: find correspondences between source schemas and target schema
- **Schema Merge**: combine source schemas into integrated target schema

Instance (entity, data) level
- transform heterogeneous source data into uniform representation
- identify and resolve data quality problems
- identify and resolve equivalent instance records: link discovery / data matching / entity resolution ...
- fusion of matching objects

Knowledge Graphs
uniform representation and semantic categorization of entities of different types
- examples: DBPedia, Yago, Wikidata, Google KG, MS Satori, Facebook, ...
- entities often extracted from other resources (Wikipedia, Wordnet etc.) or web pages, documents, web searches etc.
- Knowledge Graphs provide valuable background knowledge for enhancing entities (based on prior entity linking), improving search results ...

The Scale of Knowledge Graphs

Shao, Li, Ma (Microsoft Asia): Distributed Real-Time Knowledge Graph Serving (slides, 2015)
Main steps in data integration

Data Extraction → Data transformation → Schema matching → Entity resolution → Entity fusion
DATA INTEGRATION CHALLENGES 1

- Data quality
  - unstructured, semi-structured sources
  - need for data cleaning and enrichment
- Large-scale data integration
  - large data/metadata volume or/and many sources
  - improve runtime by reducing search space (e.g. with blocking) and parallel processing (Hadoop clusters, GPUs, etc.)
  - many sources require holistic data integration: clustering of schema elements and entities, not only binary matching
- High match quality
  - needs effective combination of several similarities
  - use of supervised ML approaches
  - representation learning (embeddings) can provide improved data input

DATA INTEGRATION CHALLENGES 2

- Support for evolution and change
  - addition of new sources and new entities without having to integrate everything again
  - incremental / dynamic vs batch / static data integration
- Graph-based data integration, e.g. to create knowledge graphs
  - integrate entities of multiple types and their relationships
  - requires holistic and incremental data integration
- Privacy for sensitive data
  - privacy-preserving record linkage and data mining
Holistic Data Integration*

scalable approaches for integrating N data sources (N >>2)

increasing need due to numerous sources, e.g., from the web
  • many thousands of web shops
  • data lakes with thousands to millions of tables

pairwise matching/linking does not scale
  • 200 sources -> 20,000 mappings

clustering-based approaches
  • represent matching entities from k sources in single cluster
  • determine cluster representative for further processing/matching
  • new entities are only compared with clusters rather than entities of all sources


AGENDA

- Introduction to Data Integration
- Entity resolution and clustering
  - introduction / ER workflow / tools
  - FAMER
  - entity clustering for clean and mixed sources (CLIP, MSCD-HAP)
- Incremental entity clustering / repair
- Summary and outlook
DATA MATCHING / ENTITY RESOLUTION

- Identification of semantically equivalent objects
  - within one data source or between different sources

Fujifilm FinePix S6800

DATA MATCHING / ENTITY RESOLUTION

DUPLICATE PUBLICATION ENTRIES

Data cleaning: Problems and current approaches
Cited by 2455 Related articles All 24 versions

Data Cleaning: Problems & Current Approaches *
D Hang-Hai, E Eihard - IEEE bulletin of the technical committee on Data ... , 2000
Cited by 5 Related articles

Problems and Current Approaches *
E Rahm, DC Do HH - IEEE Bulletin on Data Engineering, 2006, 23 (4), 2015
Cited by 5 Related articles

E Rahm, HH Do - 2010
Cited by 5 Related articles

Hong Hai Do *
E Rahm - IEEE Bulletin of the Technical Committee on Data ... , 2000
Cited by 4 Related articles

Do, H. 2000. Data cleaning: Problems and current approaches *
E Rahm - IEEE Data Engineering Bulletin
Cited by 4 Related articles

Do, H. 2000*
E Rahm, H Do - Data cleaning: Problems and current approaches, 2011
Cited by 3 Related articles

Data engineering—Special Issue on data cleaning *
E Rahm, H Do - Data Engineering, 2006
Cited by 3 Related articles

Data Cleaning: Problems and Current Approaches, IEEE Techn *
E Rahm, H Do - Bulletin on Data Engineering, 2006
Cited by 3 Related articles
ENTITY RESOLUTION WORKFLOW

- mostly only 1 or 2 sources
- \( n \geq 2 \): duplicate-free (clean) sources or not
  - clean sources: at most one entity per cluster (cluster sizes \( \leq n \))

BLOCKING & FILTERING

- naïve: pairwise matching of all entities
  - quadratic complexity, not scalable
  - strong need to reduce match search space

- Blocking
  - group similar objects within blocks / partitions
  - only compare entities of the same block
  - many variations: Standard Blocking, LSH, Sorted Neighborhood, ...

- Filtering
  - typically applied for similarity joins with fixed threshold \( t \): \( \text{sim}(e_1, e_2) \geq t \)
  - utilizes characteristics of similarity function, e.g., for string similarity
  - can utilize triangle inequality for metric similarity/distance functions
**BLOCKING TECHNIQUES**

![Genealogy trees of non-learning (a) schema-aware and (b) schema-agnostic Block Building techniques. Hybrid, hash-, and sort-based methods are marked in blue, black, and red, respectively.](image)

**MATCHING**

- combined use of several similarity values
  - attribute similarities, e.g. using numeric or string similarity measures
  - context-based matchers

- general match rules with multiple similarities
  - e.g. pubs match if $\text{title sim.} \geq 0.9$ & $\text{author sim.} > 0.4$

- learned/supervised match classification models
  - need suitable training data
DEDOOEP: DEDUPLICATION WITH HADOOP

- Parallel execution of match workflows with Hadoop
- Library of match and blocking techniques
- Learning-based match configuration
- GUI-based workflow specification
- Automatic generation and execution of Map/Reduce jobs on different clusters
- Automatic load balancing for optimal scalability

"This tool by far shows the most mature use of MapReduce for data deduplication"
www.hadoopsphere.com
RECENT ER TOOLS

- Magellan
  - PyMatcher component provides several blocking and similarity algorithms to customize match approach
  - support for machine learning, including deep learning
- JedAI
  - supports matching for structured and unstructured data
  - plethora of methods for blocking, matching and clustering
  - provides GUI

RECENT ER TOOLS 2

- FAMER
  - FAst Multi-source Entity Resolution system
  - built on Apache Flink
  - Blocking, linking and clustering module for multiple sources
  - many clustering approaches included for clean and dirty sources
  - support for incremental matching and clustering
### TOOL COMPARISON

<table>
<thead>
<tr>
<th>feature</th>
<th>Magellan</th>
<th>JedAI</th>
<th>FAMER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blocking</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Matching</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clustering</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Incremental ER</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GUI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Big Data Architecture</td>
<td>only in commercial CloudMatcher</td>
<td></td>
<td>Apache Flink</td>
</tr>
</tbody>
</table>

### FAMER TOOL

- **FAst Multi-source Entity Resolution System**
  - scalable linking & clustering for many sources

![FAMER TOOL Diagram](image)
**FAMER BATCH PIPELINE**

Data Sources
- source 1
- source 2
- source k

**LINKING**
- Blocking
- Pair-wise Comparison
- Match Classification

**CLUSTERING**
- Similarity Graph
- Cluster Set

**EXISTING CLUSTERING ALGORITHMS**

* Hassanzadeh et al.: Clustering for Duplicate Detection. VLDB 2009
PROBLEMS

overlapping clusters

source-inconsistent clusters for clean (duplicate-free) sources

each cluster should not have more than one entity per source

sources A B C D

CLIP APPROACH (ESWC BEST RESEARCH PAPER)

- CLIP (CLustering based on Link Priority) uses link strength
  - strong: maximum link from both ends
  - normal: maximum link from one end
  - weak: maximum link from no end

- CLIP
  - ignores weak links
  - focuses on strong links
  - also considers normal links
CLIP guarantees source-consistent and non-overlapping clusters

EVALUATION: GEO. DATASET

Precision  Recall  F-Measure

Threshold (θ)  0.75  0.8  0.85  0.9

InputGraph  ConCon  CCP vot  Center  MC Center  Star1  Star2  CLIP
RUNTIME AND SPEED-UP

- Experiments based on Hadoop and Apache Flink (16 machines)

<table>
<thead>
<tr>
<th></th>
<th>North Carolina Voters (10 mil.)</th>
<th>North Carolina Voters (5 Mi)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>runtimes on 16 workers - th = 0.8</td>
<td>Increasing</td>
</tr>
<tr>
<td>Connected Components</td>
<td>79 sec.</td>
<td>Near linear speed up</td>
</tr>
<tr>
<td>Star1/2</td>
<td>197/173 sec.</td>
<td></td>
</tr>
<tr>
<td>CLIP</td>
<td>228 sec.</td>
<td></td>
</tr>
<tr>
<td>Center</td>
<td>423 sec.</td>
<td></td>
</tr>
<tr>
<td>Merge Center</td>
<td>695 sec.</td>
<td></td>
</tr>
<tr>
<td>CCPiv</td>
<td>1303 sec.</td>
<td></td>
</tr>
</tbody>
</table>

MULTI-SOURCE CLEAN/DIRTY CLUSTERING

- previous assumption: data sources are duplicate-free
- more realistic assumption: some sources are dirty
  - solution: first deduplicate dirty sources
  - problem: requires immense effort and perhaps not completely successful [7]
  - solution: MSCD approaches
    - approaches that can deal with dirty sources
    - only a fraction (possibly 0%) of sources have to be clean
    - goal: achieve better match quality than general clustering scheme while avoiding limitation of requiring duplicate-free sources
    - two approaches added to FAMER based on hierarchical agglomerative clustering (HAC) and affinity propagation (AP)
MSCD-HAC

- modify Hierarchical Agglomerative Clustering \(\rightarrow\) MSCD-HAC
- iterative approach
  - initially each entity forms a cluster
  - continuously determine most similar pair of clusters \((c_i, c_j)\) as long as minimal merge similarity threshold is exceeded. Merge clusters \(c_i, c_j\) only when
    - they are Reciprocal Nearest Neighbours (RNN), i.e. \(NN(c_j) = c_i\) and \(NN(c_i) = c_j\)
    - merge results in source-consistent clusters, i.e., at most one entity of a clean source in a cluster
- 3 approaches to determine cluster similarity \(\operatorname{sim}(c_i, c_j)\)
  - Single linkage (S-LINK): \(\operatorname{sim}(c_i, c_j) = \max\{\operatorname{sim}(e_m, e_n)\}\)
  - Complete linkage (C-LINK): \(\operatorname{sim}(c_i, c_j) = \min\{\operatorname{sim}(e_m, e_n)\}\)
  - Average linkage (A-LINK): \(\operatorname{sim}(c_i, c_j) = \frac{1}{2} \cdot \left(\sum\{\operatorname{sim}(e_m, e_n)\}\right)\)

EVALUATION SUMMARY

- camera dataset* (23 sources, \(~21\) K entities)
  - combination of clean and dirty sources
  - all approaches are experimented on all MSC and MSCD datasets
  - MSCD clustering schemes MSCD-HAC and MSCD-AP are compared with
    - generic clustering schemes
    - CLIP

<table>
<thead>
<tr>
<th>MSCD dataset</th>
<th>%entities from clean sources</th>
</tr>
</thead>
<tbody>
<tr>
<td>DS-C0</td>
<td>0%</td>
</tr>
<tr>
<td>DS-C26</td>
<td>26%</td>
</tr>
<tr>
<td>DS-C32</td>
<td>32%</td>
</tr>
<tr>
<td>DS-C50</td>
<td>50%</td>
</tr>
<tr>
<td>DS-C62</td>
<td>62%</td>
</tr>
<tr>
<td>DS-C80</td>
<td>80%</td>
</tr>
<tr>
<td>DS-C100</td>
<td>100%</td>
</tr>
</tbody>
</table>

* ACM Sigmod programming contest 2020
F-MEASURE: CAMERA DATASET

match threshold = merge threshold ($\theta$)

high recall of MSCD$_{S-LINK}$
high precision of MSCD$_{S-LINK}$

as the ratio of clean sources increases, MSCD-HAC$_{S-LINK}$ obtains better F-Measure.

AGENDA

- Introduction to Data Integration
- Entity resolution and clustering
  - introduction / ER workflow / tools
  - FAMER
  - entity clustering for clean and mixed sources (CLIP, MSCD-HAP)
- Incremental entity clustering / repair
- Summary and outlook
**MOTIVATION**

- static one-time matching and clustering insufficient
- need for incremental approaches
  - data sources change over time
  - new relevant data sources are added continuously
- expensive re-computation of similarity graph/clusters to be avoided
- order in which new entities are added should have minimal impact
  - need to repair wrong clusters

**FAMER INCREMENTAL PIPELINE**
**MAX-BOTH MERGE (MBM)**

- MBM inserts new entity either into existing cluster or forms a new cluster out of it
- merging only for *max-both* (strong) links and when source-consistency constraint is met (at most one entity per clean source)

**N-DEPTH RECLUSTERING**

- reclusters new entities in $G_{new}$ with their neighbors
- can repair old cluster decisions
- limits the amount of reclustering for the sake of efficiency
- independent from order of source/entity additions
2-DEPTH RECLUSTERING: EXAMPLE

Input: Grouped Similarity Graph

2-Depth Neighbors

Output: Updated Clustered Graph

EVALUATION

- Geo. dataset
  - conf1: the best order
  - conf2: the worst order

Comparison with base approach: Greedy
[Incremental Record Linkage (Gruenheid et al., VLDB 2014)]

nDR approach is robust against source order
**EVALUATION: RUNTIME**

- North Carolina Voters, 10 Mill. entities

Incremental approaches are faster than Batch

<table>
<thead>
<tr>
<th>#worker</th>
<th>Batch</th>
<th>MB</th>
<th>1DR</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>117,852</td>
<td>5,648</td>
<td>21,179</td>
</tr>
<tr>
<td>8</td>
<td>33,791</td>
<td>2,178</td>
<td>4,283</td>
</tr>
<tr>
<td>16</td>
<td>8,542</td>
<td>1,778</td>
<td>2,513</td>
</tr>
</tbody>
</table>

threshold (θ): 0.7

MB is faster than nDR

---

**INCREMENTAL METHODS CONTRIBUTIONS**

- Incremental approaches are much faster and similarly effective than batch ER
- Reclustering approach nDR achieves same quality than batch ER while being much faster
- Quality of nDR does not depend on the order in which new entities are added
SUMMARY

- Data integration still faces many challenges: automation, data quality, efficiency/scalability, privacy support, continuous change ...
- Need for multi-source entity resolution with clustering
- FAMER integrates new and effective approaches for:
  - Consideration of duplicate-free (clean) data sources
  - Support for incremental matching/clustering and cluster repair

OPEN RESEARCH PROBLEMS

- Largely automatic creation/refinement of large-scale knowledge graphs
- Requires tackling of several tasks/challenges:
  - Development and evolution of KG ontology
  - Initial population of KG
  - Data acquisition/extraction/cleaning for new data to be integrated
  - Learning-based classification of new entities
  - Incremental schema/property matching for many entity types
  - Incremental entity resolution/clustering for many entity types
  - Entity fusion...
- Multi-modal data integration
REFERENCES (1)

- P. Christen: Data Matching. Springer 2012
- L. Dong: Challenges and Innovations in Building a Product Knowledge Graph. Tutorial, KDD 2018
- A. Gruenheid et al.: Incremental record linkage. VLDB 2014
- O. Hassanzadeh et al.: Clustering for Duplicate Detection. VLDB 2009
- S. Lerm, A. Saeedi, E. Rahm: Extended Affinity Propagation Clustering for Multi-source Entity Resolution. BTW 2021
- M. Nentwig, A. Groß, Anika; M. Möller, E. Rahm: Distributed Holistic Clustering on Linked Data. LNCS 10574, 2017, pp 371-382

REFERENCES (2)

- G. Papadakis et al.: The return of jedAI: end-to-end entity resolution for structured and semi-structured data. PVLDB 2018
- D. Obraczka, A. Saeedi, A. E. Rahm, E: Knowledge Graph Completion with FAMER. Proc. KDD DI2KG, 2019
- A. Saeedi, L. David, E. Rahm, E: Matching Entities from Multiple Sources with Hierarchical Agglomerative Clustering. KEOD 2021
- A. Saeedi, M. Nentwig, E. Peukert, E. Rahm: Scalable matching and clustering of entities with FAMER. CSIM Quarterly 2018
- A. Saeedi, E. Peukert, E. Rahm: Using Link Features for Entity Clustering in Knowledge Graphs. ESWC 2018
- A. Saeedi, E. Peukert, E. Rahm: Incremental Multi-source Entity Resolution for Knowledge Graph Completion. ESWC 2020
- J. Shao, Q.; Wang, A. Wijesinghe, E. Rahm: ERGAN: Generative Adversarial Networks for Entity Resolution. ICDM 2020
- M. Wilke, E. Rahm: Towards Multi-modal Entity Resolution for Product Matching. GVDB 2021