TU Graz

SCIENCE
PASSION
TECHNOLOGY

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

ISDS

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**

IT Community Styria

- **#3 Kafka Meetup Graz**
  - **Nov 27, 5.45pm - 9pm**, NETCONOMY
  - 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

No Global Keys

Uniqueness & duplicates

Contradictions & wrong values

Missing Values

Ref. Integrity

[**Credit:** Felix Naumann]

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

Zip	City
98120	San Jose
90001	Lost Angeles

Typos

Terminology

[Douglas Burdick, Ronald Fagin, Phokion G. Kolaitis, Lucian Popa, Wang-Chiew Tan: Expressive power of entity-linking frameworks. **J. Comput. Syst. Sci. 2019**]

- **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

[Xin Luna Dong, Theodoros Rekatsinas: Data Integration and Machine Learning: A Natural Synergy. Tutorials, **SIGMOD 2018**, **PVLDB 2018**, **KDD 2019**]

[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 x B if match (**link**), no match (**non-link**), or not enough evidence (**possible-link**)

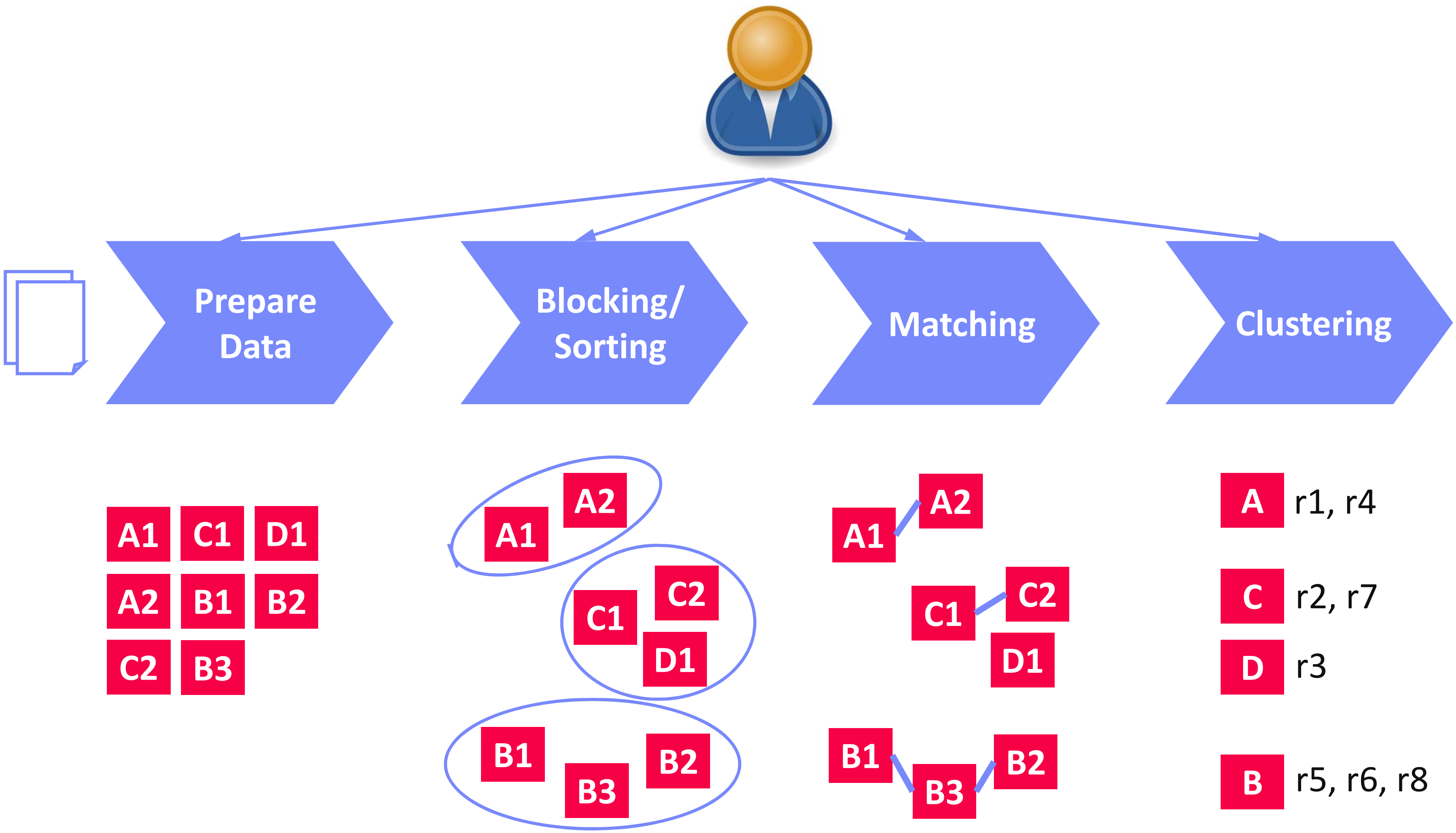
[Ivan Fellegi, Alan Sunter: A Theory for Record Linkage, J. American. Statistical Assoc., pp. 1183-1210, **1969**]

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

Name	Position	Affiliation	Research
Matthias Boehm	RSM	IBM Research – Almaden	Apache SystemML
Matthias Böhm	Prof	TU Graz	SystemDS

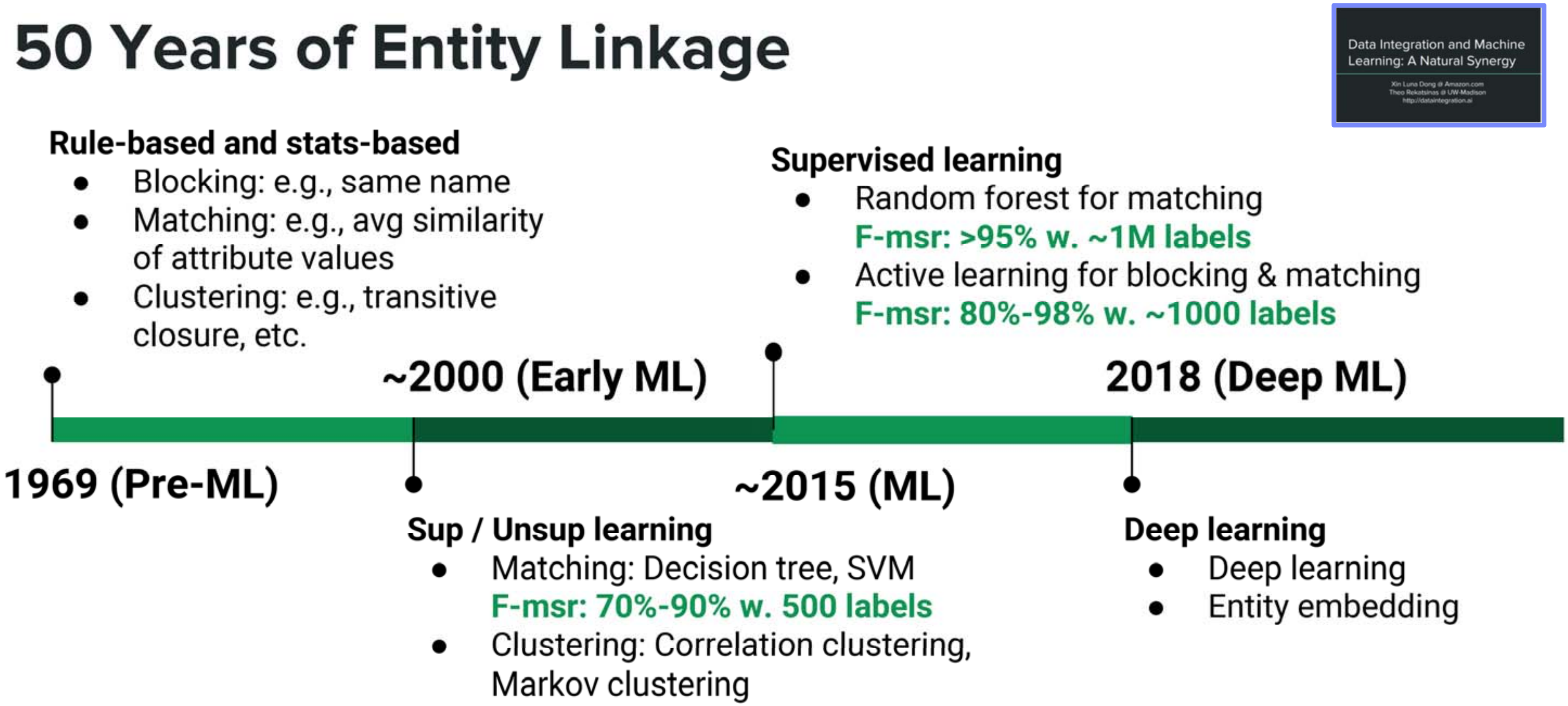
→ **Similarity Measures**
  - Token-based: e.g., Jaccard J(A,B) = $(A \cap B) / (A \cup B)$
  - Edit-based: e.g., Levenshtein lev(A,B) → min(replace, insert, delete)
  - Phonetic similarity (e.g., soundex, metaphone), **Python lib Jellyfish**

Entity Resolution Pipeline



Entity Linking Approaches

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



Data Preparation

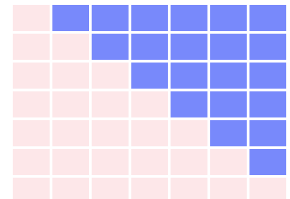
- **#1 Schema Matching and Mapping**
  - See lecture **04 Schema Matching and Mapping**
  - Create **homogeneous schema** for comparison
  - Split composite attributes
- **#2 Normalization**
  - Removal of special characters and white spaces
  - **Stemming**
  - **Capitalization** (to upper/lower)
  - Remove redundant works, resolve abbreviations

likes/liked/likely/liking → like

- **#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 → **O(n²) complexity**

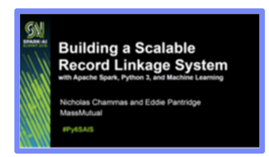


- **#2 Blocking / Partitioning**
  - Efficiently create small blocks of similar records for pair-wise matching
  - **Basic:** equivalent values on selected attributes (name)
  - **Predicates:** whole field, token field, common integer, same x char start, n-grams
  - **Hybrid:** disjunctions/conjunctions
  - Blocking Keys: → JR01111

John Roberts	20 Main St	Plainville	MA	01111

  - Learned: Minimal rule set via greedy algorithms

→ **Significant reduction:** 1M records → 1T pairs
→ 1K partitions w/ 1K records → 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

Distributed Tuple Representation

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

```
X %*% Y      h1=[-1, 1,1], h2=[ 1,1, 1],
             h3=[-1,-1,1], h4=[-1,1,-1],
v[t1]=[0.45,0.8,0.85]  [1.2,2.1,-0.4,-0.5] → [1,1,-1,-1] → [12]
v[t2]=[0.4,0.85,0.75]  [1.2,2.0,-0.5,-0.3] → [1,1,-1,-1] → [12]
```

Matching

- **#1 Basic Similarity Measures**
  - Pick similarity measure sim(r, r') and thresholds: high θ_h (and low θ_l)
  - Record similarity: avg attribute similarity
  - **Match:** sim(r, r') > θ_h **Non-match:** sim(r, r') < θ_l
    possible match: θ_l < sim(r, r') < θ_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

[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.

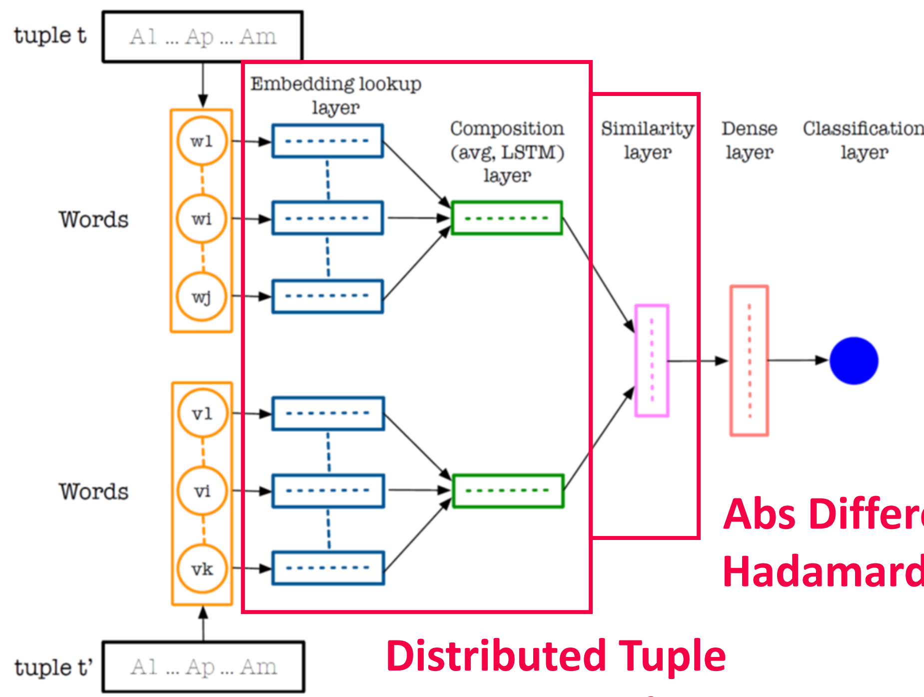
- **Deep Learning for ER**
  - Automatic **representation learning** from text (avoid feature engineering)
  - Leverage pre-trained **word embeddings for semantics** (no syntactic limitations)

- **Example DeepER**

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

- **Example Magellan**
  - Text and dirty data

[Sidharth Mudgal et al: Deep Learning for Entity Matching: A Design Space Exploration. **SIGMOD 2018**]

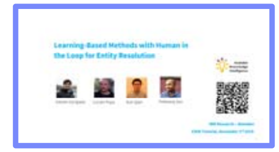


Abs Difference / Hadamard Prod.

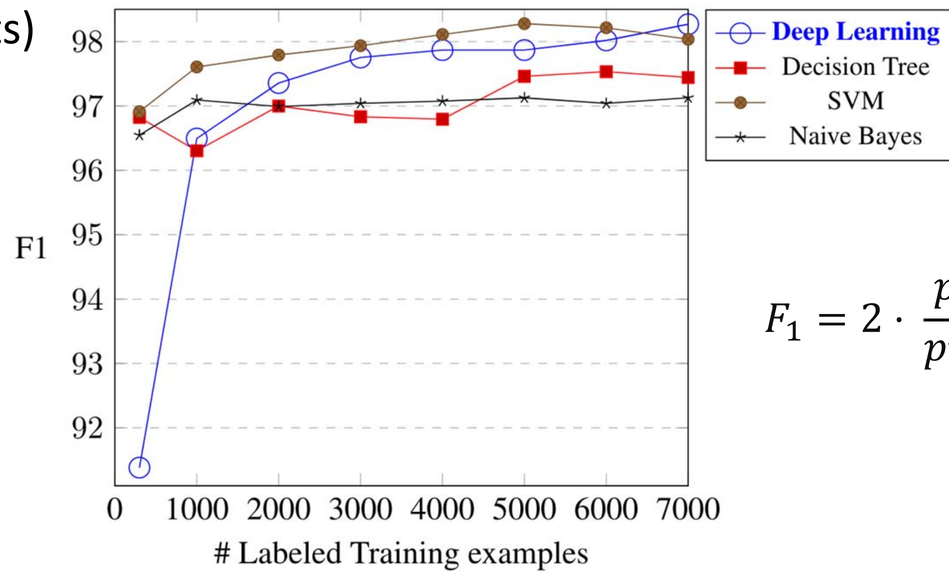
Distributed Tuple Representation

Matching, cont.

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



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



$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

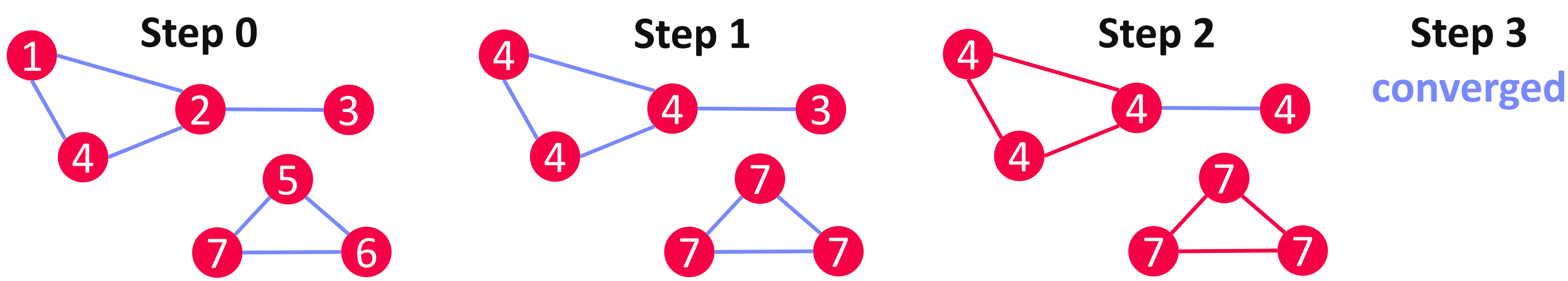
→ Transfer Learning
- Learn model from high-resource ER scenario (w/ regularization)
- Fine-tune using low-resource examples

→ Active Learning
- Select instances for tuning to min labeling

[Jungo Kasai et al: Low-resource Deep Entity Resolution with Transfer and Active Learning. **ACL 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 > θ_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

[Oktie Hassanzadeh, Fei Chiang, Renée J. Miller, Hyun Chul Lee: Framework for Evaluating Clustering Algorithms in Duplicate Detection. **PVLDB 2009**]

Incremental Data Deduplication

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

- **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

https://docs.dedupe.io/en/latest/API-documentation.html
https://dedupeio.github.io/dedupe-examples/docs/csv_example.html

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

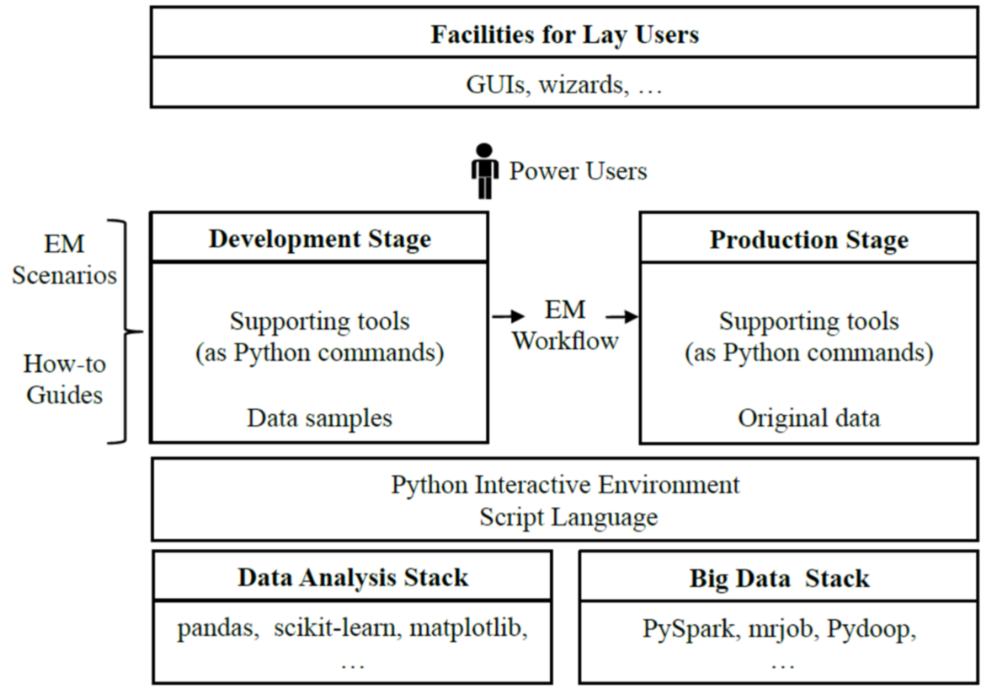
```
fields = [
  {'field':'Site name', 'type':'String'},
  {'field':'Address', 'type':'String'}]

# 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
```

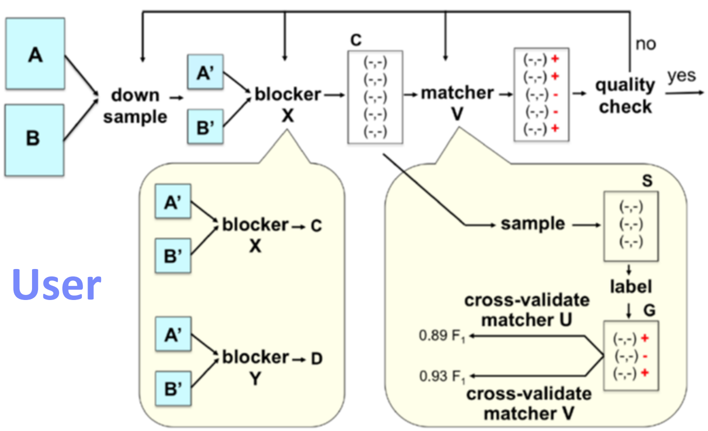
Magellan (UW-Madison)

[Pradap Konda et al.: Magellan: Toward Building Entity Matching Management Systems. **PVLDB 2016**]

- **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



User



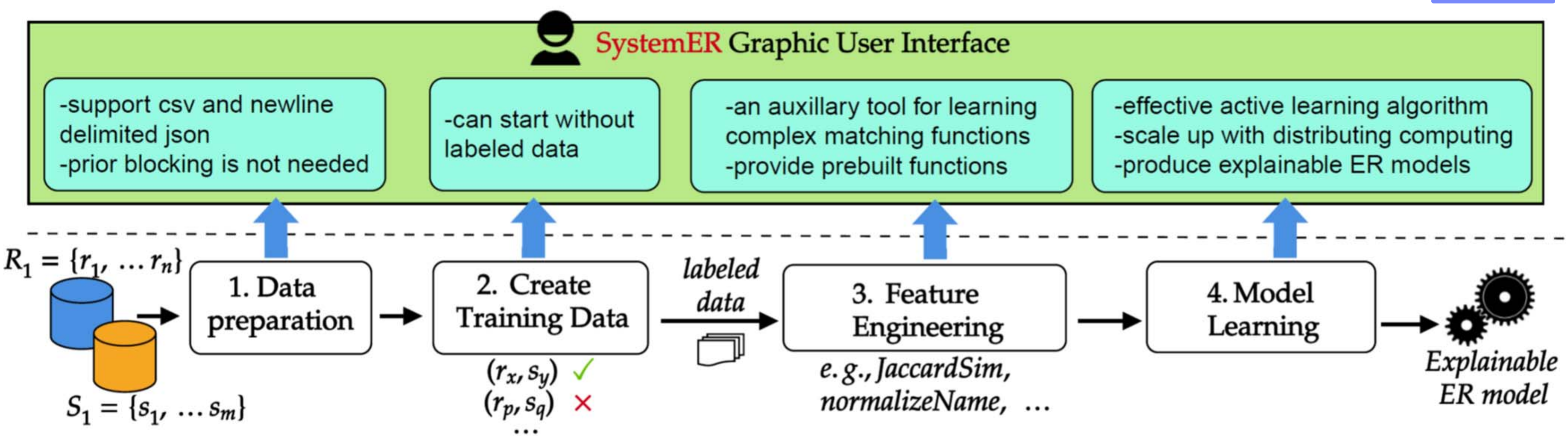
Select the best blocker: X, Y

Select the best matcher: U, V

[Yash Govind et al: Entity Matching Meets Data Science: A Progress Report from the Magellan Project. **SIGMOD 2019**]

SystemER (IBM Almaden – Research)

[Kun Qian, Lucian Popa, Prithviraj Sen: SystemER: A Human-in-the-loop System for Explainable Entity Resolution. **PVLDB 2019**]



Learns explainable ER rules (in HIL)

```
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**]

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)

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

Matthias Boehm[1,2], Iulian Antonov[2], Sebastian Baunsgaard[1]; Mark Dokter[2], Robert Ginthör[2], Kevin Innerebner[1], Florijan Klezin[2], Stefanie Lindstaedt[1,2], Arnab Phani[1], Benjamin Rath[1], Berthold Reinwald[3], Shafaq Siddiqi[1]

[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

- **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**