Scalable techniques for Entity Resolution

Prof. Giovanni Simonini

Università degli Studi di Modena e Reggio Emilia

simonini@unimore.it

Entity Resolution (ER)

Neme	Profession	Year	Address
John Abram Jr	car seller	1985	main Street

name1	name2	birth yr	job	Loc
Jon Jr	Abram	85	car retail	main st.
Simon	White	75	auto rental	Lincoln LA

full name	b. date	work info	Address
Jon Abrm Jr	31 oct	car seller	main LA
Ellen Smith	21 may	retail	Abrams street 30 NY

name	name_2	year	occupation	mail
Ellen	Smith	85	auto retail	Abram st. 30 NY

ER is the task of **identifying records** in databases **that refer to the same real world entity**

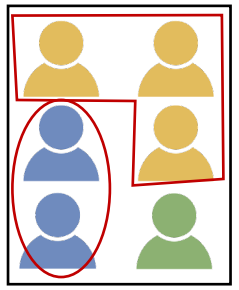
r	name	birth date	work info	Address
r_1 r_2 r_3	John Abram Jr.	31/10/1985	car retail	Main Street LA
r_4 r_5	Ellen Smith	21/05/1985	car retail	Abram Street NY
r_6	Simon White	1/1/1975	car rental	Lincoln Street LA

Types of Entity Resolution

- The input of ER consists of profile collections that can be of two types:
  - **Clean**: each collection is duplicate-free
  - **Dirty**: each collection contains duplicates

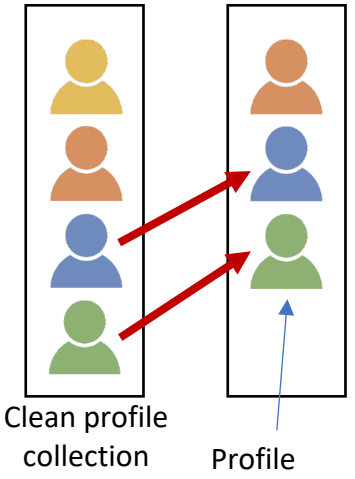
- Based on the input, we distinguish ER into 3 sub-tasks:
  - Clean-Clean ER (a.k.a. **Record Linkage**)
  - Dirty-Clean ER
  - Dirty-Dirty ER a.k.a. Deduplication

Deduplication



Dirty profile collection

Record Linkage



Clean profile collection

Profile

ER is expensive

Neme	Profession	Year	Address
John Abram Jr	car seller	1985	main Street

name1	name2	birth yr	job	Loc
Jon Jr	Abram	85	car retail	main st.
Simon	White	75	auto rental	Lincoln LA

full name	b. date	work info	Address
Jon Abrm Jr	31 oct	car seller	main LA
Ellen Smith	21 may	retail	Abrams street 30 NY

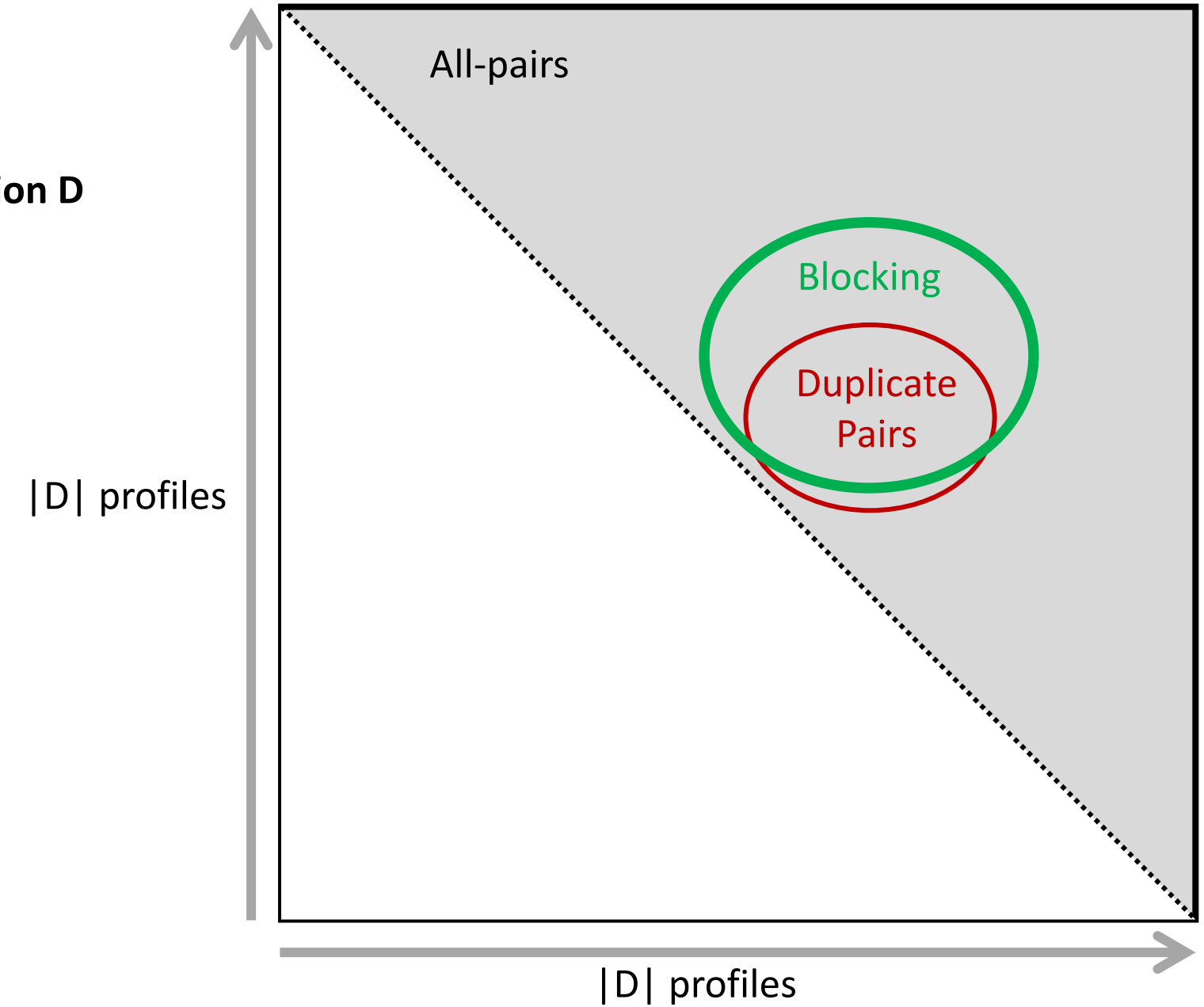
name	name_2	year	occupation	mail
Ellen	Smith	85	auto retail	Abram st. 30 NY

r	name	birth date	work info	Address
r_1 r_2 r_3	John Abram Jr.	31/10/1985	car retail	Main Street LA
r_4 r_5	Ellen Smith	21/05/1985	car retail	Abram Street NY
r_6	Simon White	1/1/1975	car rental	Lincoln Street LA

- Performing ER in the **naïve way**
  - comparing all possible possible paris is **$O(n^2)$**
- To avoid quadratic complexity:
  - Indexing functions are employed
    - to limit the number of comparisons: **#comparisons << n^2**

Computational cost

Input:
Entity Collection D



Example of Computational cost

DBPedia 3.0rc ↔ DBPedia 3.4

1.2 million entities ↔ 2.2 million entities

Entity matching: Jaccard similarity of all tokens
Cost per comparison: 0.045 milliseconds (average of 0.1 billion comparisons)

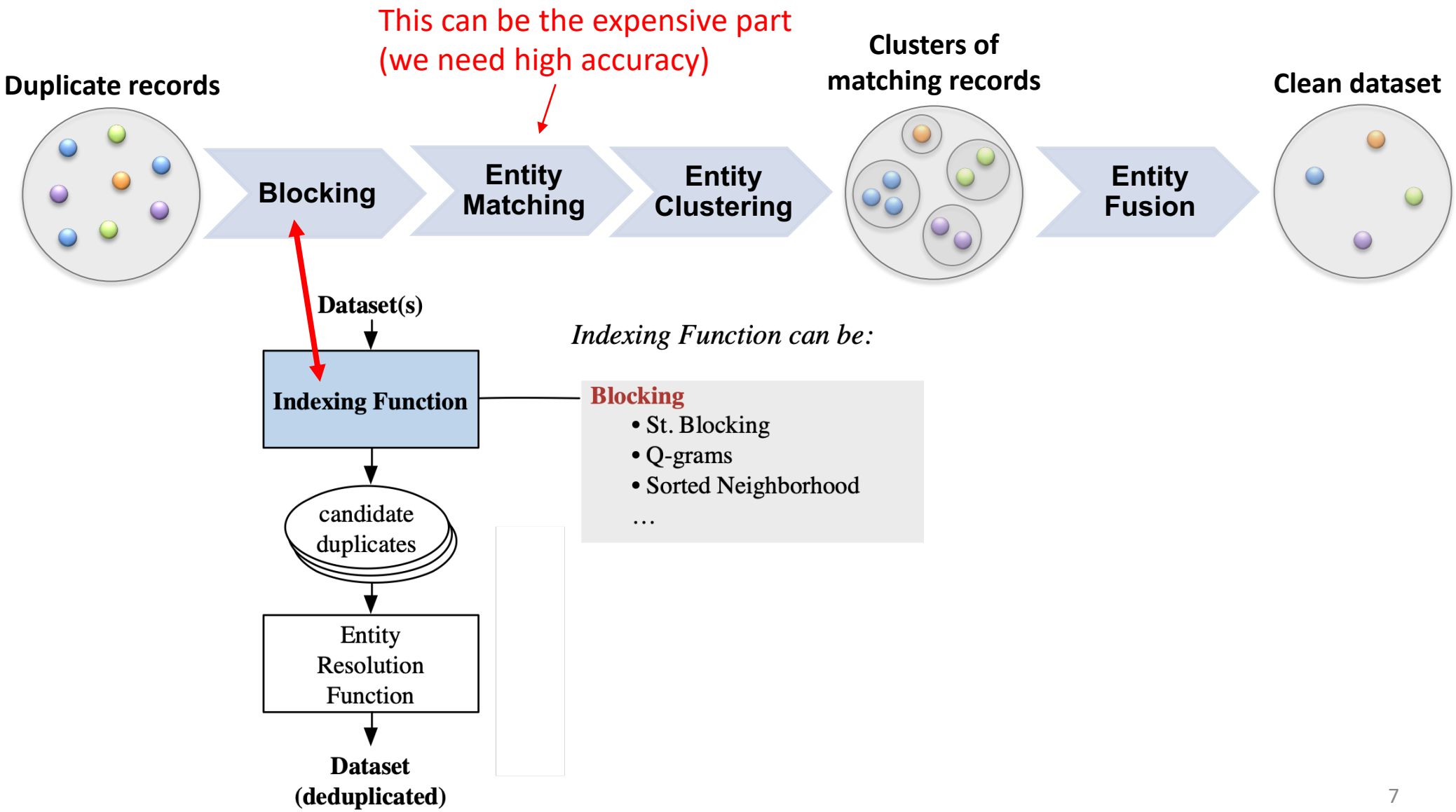
Brute-force approach

Comparisons: $2.58 \cdot 10^{12}$
Recall: 100%
Running time: 1,344 days → **3.7 years**

Optimized Blocking Workflow

Overhead time: 4 hours
Comparisons: $8.95 \cdot 10^{6}$
Recall: 99%
Total Running time: **10 hours**

Blocking


This can be the expensive part
(we need high accuracy)
Duplicate records
Blocking
Entity Matching
Entity Clustering
Clusters of matching records
Entity Fusion
Clean dataset
Dataset(s)
Indexing Function
candidate duplicates
Entity Resolution Function
Dataset (deduplicated)
Indexing Function can be:
Blocking
• St. Blocking
• Q-grams
• Sorted Neighborhood
…

Scaling ER

Neme	Profession	Year	Address
John Abram Jr	car seller	1985	main Street

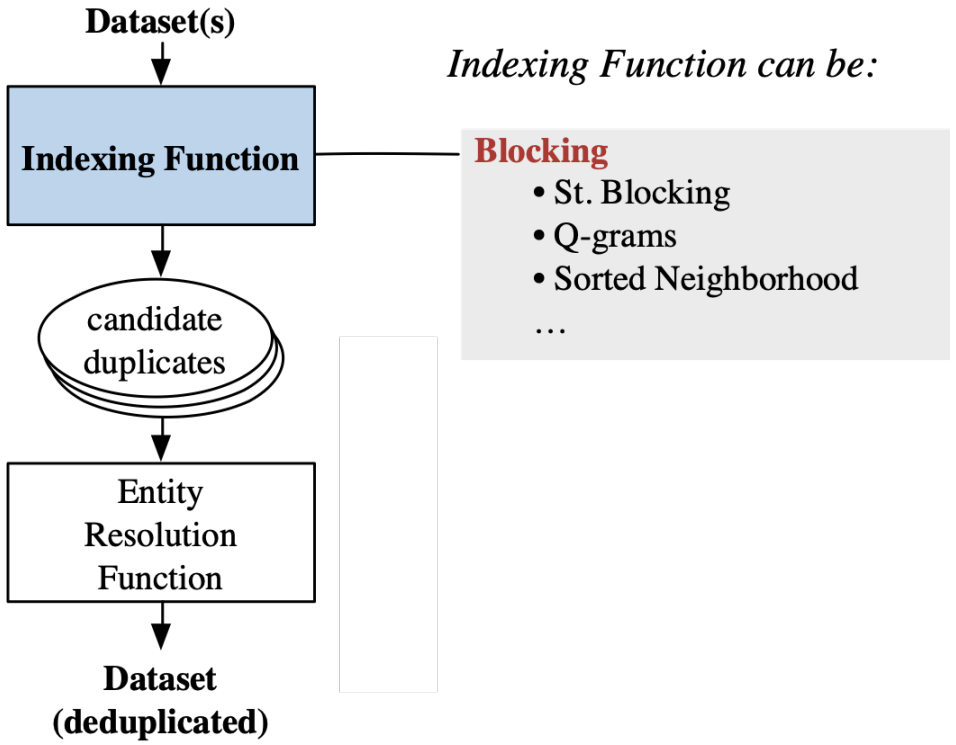
name1	name2	birth yr	job	Loc
Jon Jr	Abram	85	car retail	main st.
Simon	White	75	auto rental	Lincoln LA

full name	b. date	work info	Address
Jon Abrm Jr	31 oct	car seller	main LA
Ellen Smith	21 may	retail	Abrams street 30 NY

name	name_2	year	occupation	mail
Ellen	Smith	85	auto retail	Abram st. 30 NY

r	name	birth date	work info	Address
r_1 r_2 r_3	John Abram Jr.	31/10/1985	car retail	Main Street LA
r_4 r_5	Ellen Smith	21/05/1985	car retail	Abram Street NY
r_6	Simon White	1/1/1975	car rental	Lincoln Street LA

- Indexing
  - **Blocking**
    - creates **blocks** of matching candidates
    - compare only those candidates



Scaling ER: Blocking

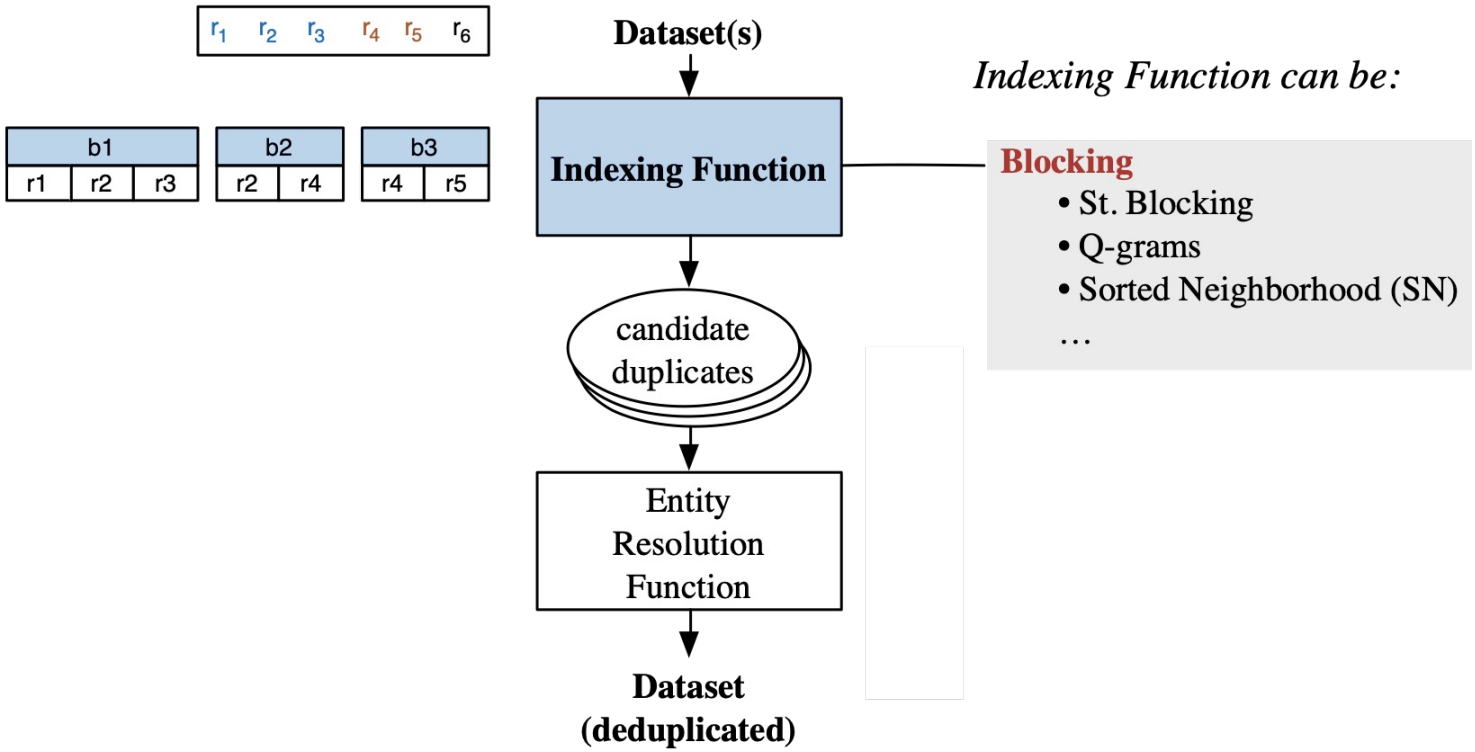
Neme	Profession	Year	Address
John Abram Jr	car seller	1985	main Street

name1	name2	birth yr	job	Loc
Jon Jr	Abram	85	car retail	main st.
Simon	White	75	auto rental	Lincoln LA

full name	b. date	work info	Address
Jon Abrm Jr	31 oct	car seller	main LA
Ellen Smith	21 may	retail	Abrams street 30 NY

name	name_2	year	occupation	mail
Ellen	Smith	85	auto retail	Abram st. 30 NY

r	name	birth date	work info	Address
r_1 r_2 r_3	John Abram Jr.	31/10/1985	car retail	Main Street LA
r_4 r_5	Ellen Smith	21/05/1985	car retail	Abram Street NY
r_6	Simon White	1/1/1975	car rental	Lincoln Street LA



Scaling ER: Candidates paris

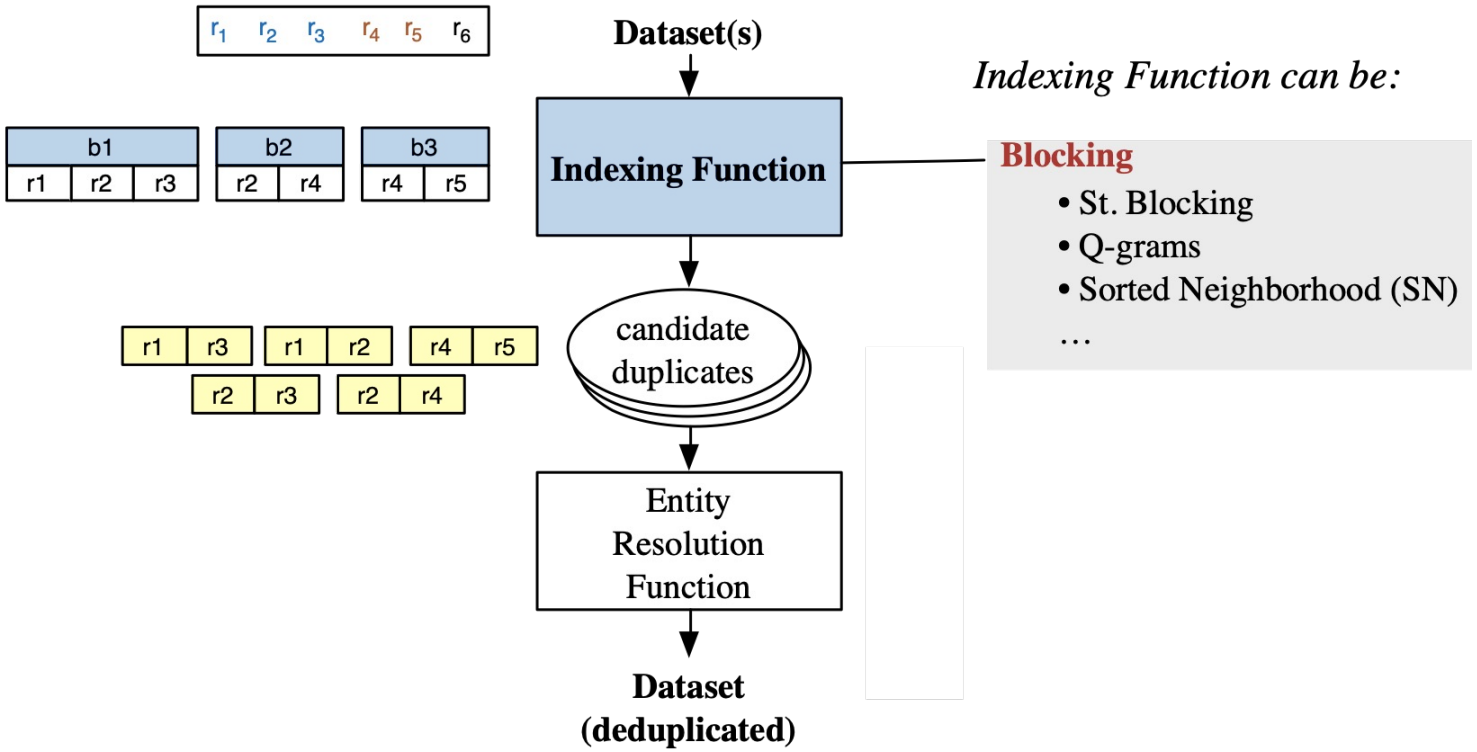
Neme	Profession	Year	Address
John Abram Jr	car seller	1985	main Street

name1	name2	birth yr	job	Loc
Jon Jr	Abram	85	car retail	main st.
Simon	White	75	auto rental	Lincoln LA

full name	b. date	work info	Address
Jon Abrm Jr	31 oct	car seller	main LA
Ellen Smith	21 may	retail	Abrams street 30 NY

name	name_2	year	occupation	mail
Ellen	Smith	85	auto retail	Abram st. 30 NY

r	name	birth date	work info	Address
r_1 r_2 r_3	John Abram Jr.	31/10/1985	car retail	Main Street LA
r_4 r_5	Ellen Smith	21/05/1985	car retail	Abram Street NY
r_6	Simon White	1/1/1975	car rental	Lincoln Street LA



Scaling ER: Resolution

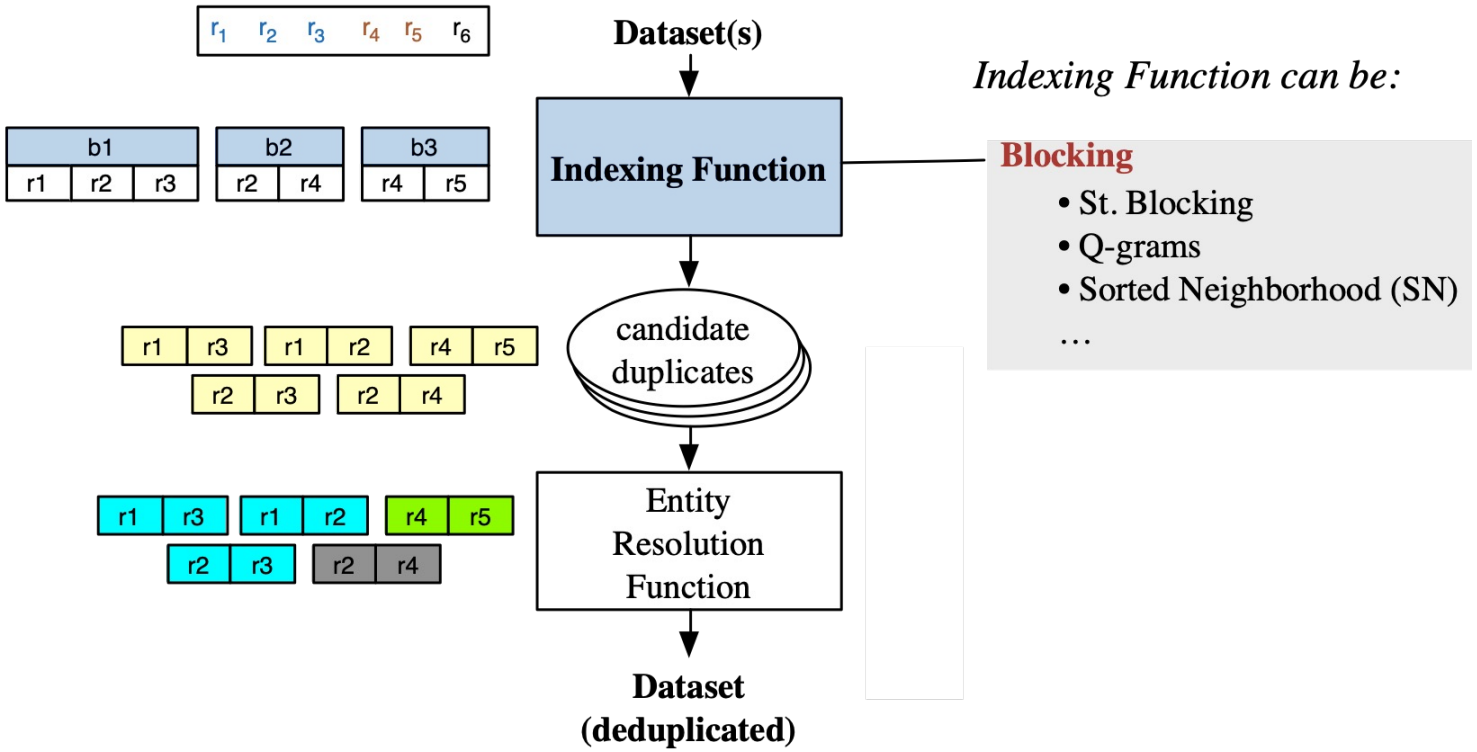
Neme	Profession	Year	Address
John Abram Jr	car seller	1985	main Street

name1	name2	birth yr	job	Loc
Jon Jr	Abram	85	car retail	main st.
Simon	White	75	auto rental	Lincoln LA

full name	b. date	work info	Address
Jon Abrm Jr	31 oct	car seller	main LA
Ellen Smith	21 may	retail	Abrams street 30 NY

name	name_2	year	occupation	mail
Ellen	Smith	85	auto retail	Abram st. 30 NY

r	name	birth date	work info	Address
r_1 r_2 r_3	John Abram Jr.	31/10/1985	car retail	Main Street LA
r_4 r_5	Ellen Smith	21/05/1985	car retail	Abram Street NY
r_6	Simon White	1/1/1975	car rental	Lincoln Street LA



Indexing Function can be:

Blocking

- St. Blocking
- Q-grams
- Sorted Neighborhood (SN)

…

Scaling ER: Final result

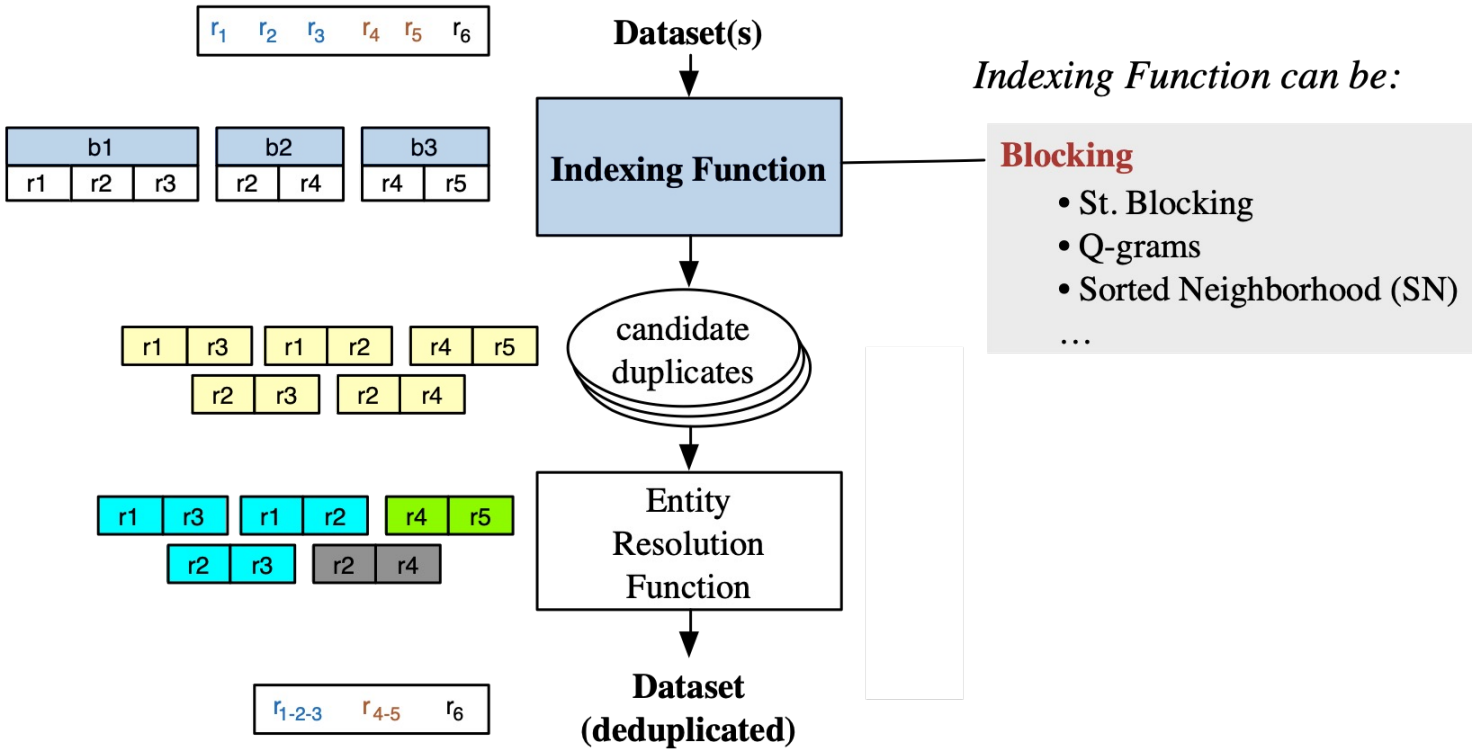
Neme	Profession	Year	Address
John Abram Jr	car seller	1985	main Street

name1	name2	birth yr	job	Loc
Jon Jr	Abram	85	car retail	main st.
Simon	White	75	auto rental	Lincoln LA

full name	b. date	work info	Address
Jon Abrm Jr	31 oct	car seller	main LA
Ellen Smith	21 may	retail	Abrams street 30 NY

name	name_2	year	occupation	mail
Ellen	Smith	85	auto retail	Abram st. 30 NY

r	name	birth date	work info	Address
r_1 r_2 r_3	John Abram Jr.	31/10/1985	car retail	Main Street LA
r_4 r_5	Ellen Smith	21/05/1985	car retail	Abram Street NY
r_6	Simon White	1/1/1975	car rental	Lincoln Street LA



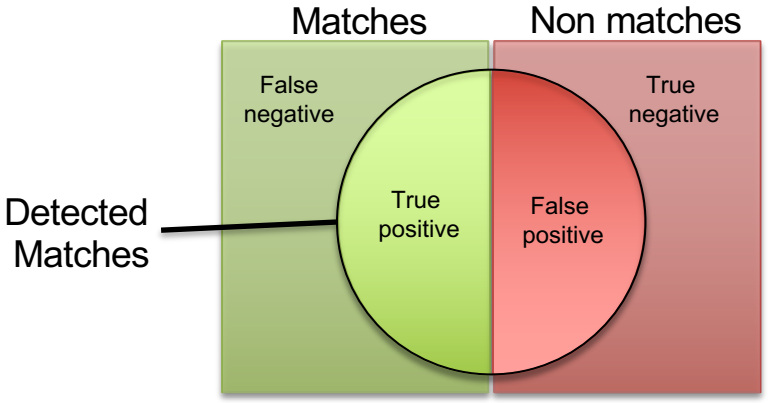
Measuring blocking quality

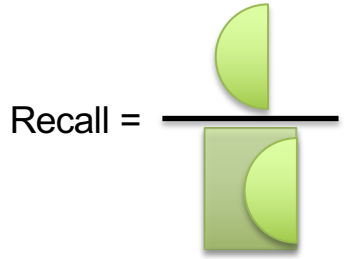
- We employ Recall and Precision, but often we find them with a different name, respecrivelly: Pair Completeness (**PC**) and Pair Quality (**PQ**)
  - To differenciate from Recall and Precision of the final ER pipeline, i.e., after the matching has been applied

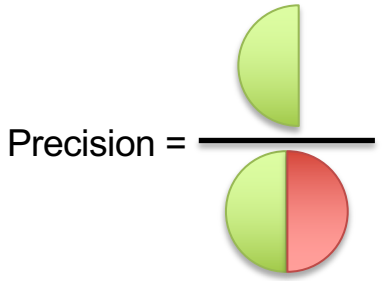
$$PC \equiv Recall_{blocks} = \frac{\boldsymbol{Indexed\ true\ matches}\ in\ the\ blocks}{\boldsymbol{existing\ matches}\ in\ the\ dataset}$$

$$PQ \equiv Precision_{blocks} = \frac{\boldsymbol{Indexed\ true\ matches}\ in\ the\ blocks}{\boldsymbol{Indexed\ pairs}\ in\ the\ blocks}$$

Precision and Recall







Blocking keys (for “small” data)

Neme	Profession	Year	Address
John Abram Jr	car seller	1985	main Street

name1	name2	birth yr	job	Loc
Jon Jr	Abram	85	car retail	main st.
Simon	White	75	auto rental	Lincoln LA

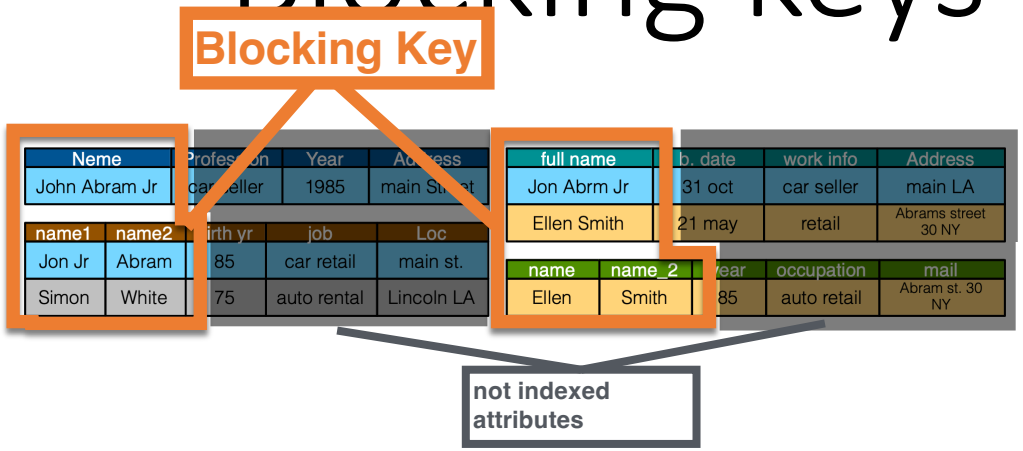
full name	b. date	work info	Address
Jon Abrm Jr	31 oct	car seller	main LA
Ellen Smith	21 may	retail	Abrams street 30 NY

name	name_2	year	occupation	mail
Ellen	Smith	85	auto retail	Abram st. 30 NY

r	name	birth date	work info	Address
r_1 r_2 r_3	John Abram Jr.	31/10/1985	car retail	Main Street LA
r_4 r_5	Ellen Smith	21/05/1985	car retail	Abram Street NY
r_6	Simon White	1/1/1975	car rental	Lincoln Street LA

1. Schema-alignment (mapping among attributes)
2. Indexing criterion
   - Example: **St. Blocking**
     1. **blocking key:** one (or more) attribute(s)
        - **e.g. Name + Surname**

Blocking keys (for “small” data)



r	name	birth date	work info	Address
r_1 r_2 r_3	John Abram Jr.	31/10/1985	car retail	Main Street LA
r_4 r_5	Ellen Smith	21/05/1985	car retail	Abram Street NY
r_6	Simon White	1/1/1975	car rental	Lincoln Street LA

1. Schema-alignment (mapping among attributes)
2. Indexing criterion
   - Example: **St. Blocking**
     1. **blocking key:** one (or more) attribute(s)
     - **e.g. Name + Surname**
     1. index together records that have the same value for that (or those) attribute(s)
     - only r_4 and r_5 are indexed together
     - r_1, r_2, r_3 have spelling errors

Blocking for Big Data?

1. **Blocking keys selection** is a **difficult** and **error-prone** task, it generally requires:
   - **domain experts**
   - **labeled data** and **machine learning**
2. **Schema-alignment** not always achievable
   - Big Data: huge volume, schema heterogeneity, high noise, etc.
     - E.g.: **Google Base** has over **10k entity types** described with **100k unique schemas**

- **Solution: Schema-agnostic Blocking (unsupervised)**

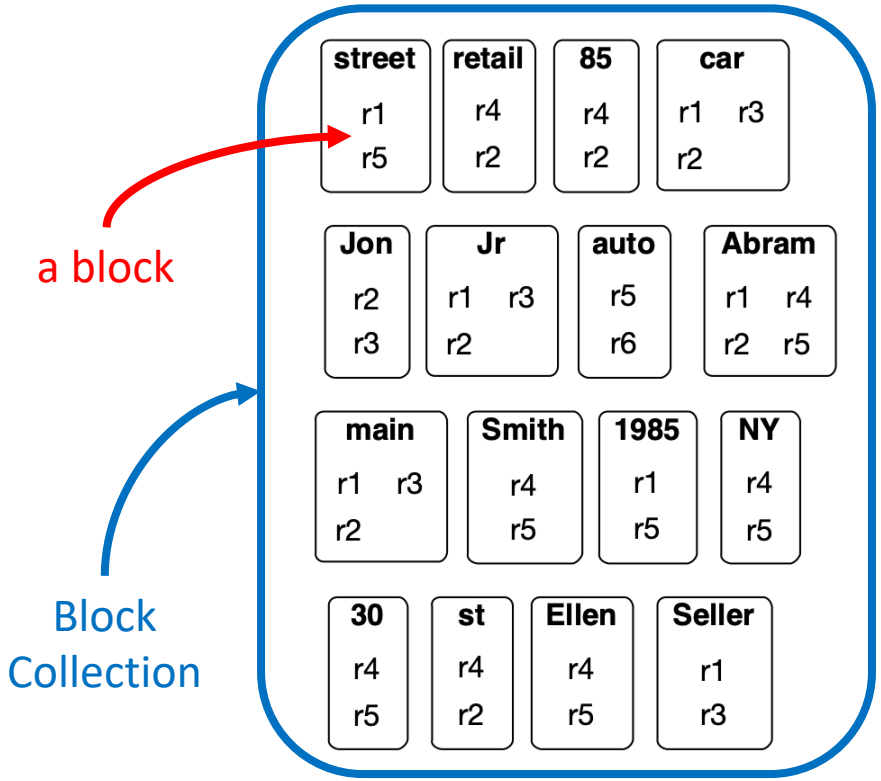
Schema-agnostic (meta-)blocking

Neme	Profession	Year	Address
John Abram Jr	car seller	1985	main Street

name1	name2	birth yr	job	Loc
Jon Jr	Abram	85	car retail	main st.
Simon	White	75	auto rental	Lincoln LA

full name	b. date	work info	Address
Jon Abrm Jr	31 oct	car seller	main LA
Ellen Smith	21 may	retail	Abrams street 30 NY

name	name_2	year	occupation	mail
Ellen	Smith	85	auto retail	Abram st. 30 NY



- Each token is blocking key
- Regardless of the attribute in which it appears
- Each profile is indexed with multiple blocking key
  - High overlapp
    - Very unlikely to miss a match
    - Superfluous candidates

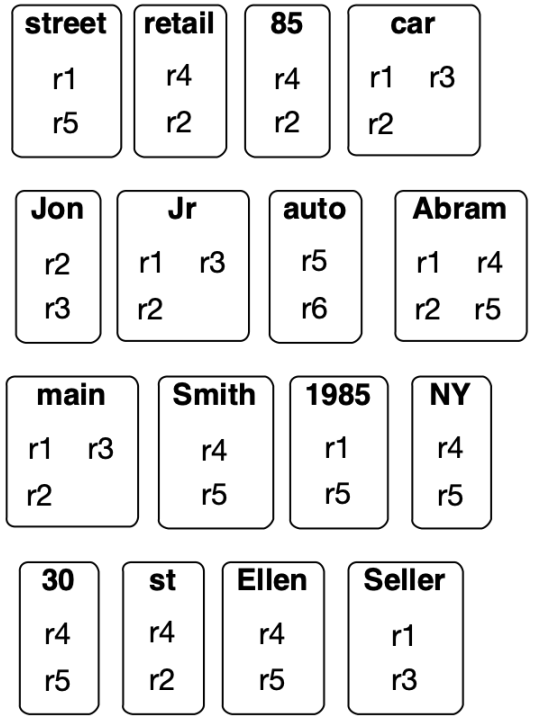
Schema-agnostic Token Blocking (TB)

Neme	Profession	Year	Address
John Abram Jr	car seller	1985	main Street

name1	name2	birth yr	job	Loc
Jon Jr	Abram	85	car retail	main st.
Simon	White	75	auto rental	Lincoln LA

full name	b. date	work info	Address
Jon Abrm Jr	31 oct	car seller	main LA
Ellen Smith	21 may	retail	Abrams street 30 NY

name	name_2	year	occupation	mail
Ellen	Smith	85	auto retail	Abram st. 30 NY

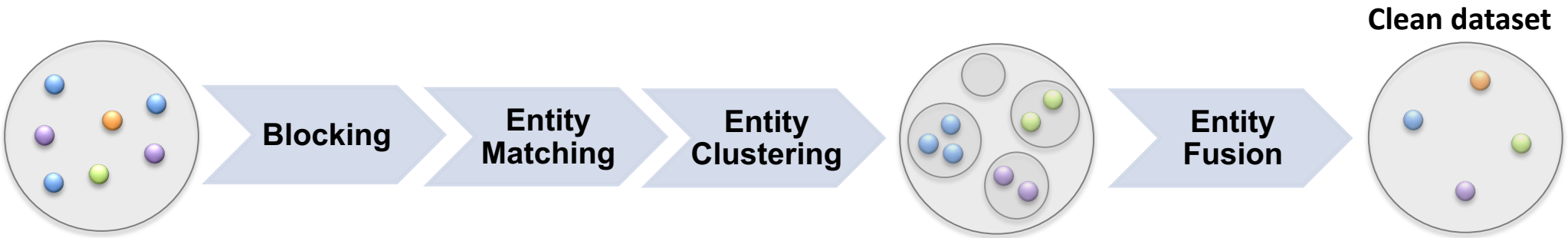


- **PC** is very high
- **PQ** is very low

→ It is OK to trade a bit of PC for improving PQ

Recall	Precision
1.0	0.15

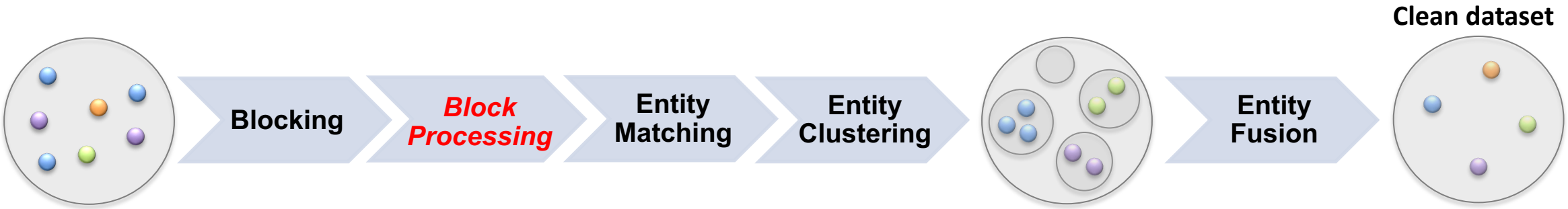
Block processing



- Spend some budget (human labour for labelling/time/computation) to weight the candidate comparison (i.e., the output of blocking)

- To try to keep only most promising comparisons
  - The strategy depends on the application
    - Global/Local Top-k
    - Clobal/Local threshold

Block processing



- Most effective is Meta-blocking:
  - Unsupervised meta-blocking
  - Supervised meta-blocking

- Other strategies to enhance the processing:
  - Block Filtering
  - Plock Purging

Schema-agnostic Meta-blocking (MB): the Blocking Graph

Neme	Profession	Year	Address
John Abram Jr	car seller	1985	main Street

name1	name2	birth yr	job	Loc
Jon Jr	Abram	85	car retail	main st.
Simon	White	75	auto rental	Lincoln LA

full name	b. date	work info	Address
Jon Abrm Jr	31 oct	car seller	main LA
Ellen Smith	21 may	retail	Abrams street 30 NY

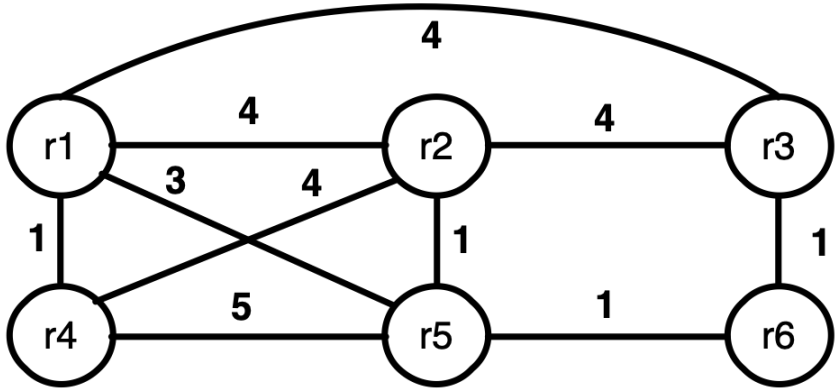
name	name_2	year	occupation	mail
Ellen	Smith	85	auto retail	Abram st. 30 NY

street	retail	85	car
r1	r4	r4	r1 r3
r5	r2	r2	r2

Jon	Jr	auto	Abram
r2	r1 r3	r5	r1 r4
r3	r2	r6	r2 r5

main	Smith	1985	NY
r1 r3	r4	r1	r4
r2	r5	r5	r5

30	st	Ellen	Seller
r4	r4	r4	r1
r5	r2	r5	r3



Schema-agnostic Meta-blocking (MB): the Blocking Graph

Neme	Profession	Year	Address
John Abram Jr	car seller	1985	main Street

name1	name2	birth yr	job	Loc
Jon Jr	Abram	85	car retail	main st.
Simon	White	75	auto rental	Lincoln LA

full name	b. date	work info	Address
Jon Abrm Jr	31 oct	car seller	main LA
Ellen Smith	21 may	retail	Abrams street 30 NY

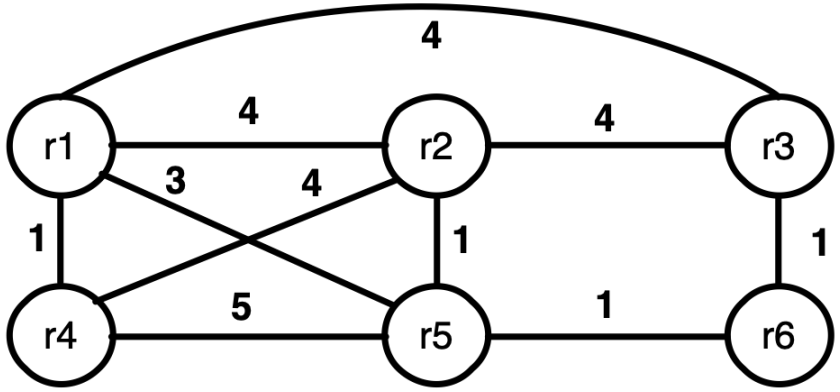
name	name_2	year	occupation	mail
Ellen	Smith	85	auto retail	Abram st. 30 NY

street	retail	85	car
r1 r5	r4 r2	r4 r2	r1 r3 r2

Jon	Jr	auto	Abram
r2 r3	r1 r3 r2	r5 r6	r1 r4 r2 r5

main	Smith	1985	NY
r1 r3 r2	r4 r5	r1 r5	r4 r5

30	st	Ellen	Seller
r4 r5	r4 r2	r4 r5	r1 r3



Recall	Precision
1.0	0.4

Schema-agnostic MB: Blocking graph pruning

Neme	Profession	Year	Address
John Abram Jr	car seller	1985	main Street

name1	name2	birth yr	job	Loc
Jon Jr	Abram	85	car retail	main st.
Simon	White	75	auto rental	Lincoln LA

full name	b. date	work info	Address
Jon Abrm Jr	31 oct	car seller	main LA
Ellen Smith	21 may	retail	Abrams street 30 NY

name	name_2	year	occupation	mail
Ellen	Smith	85	auto retail	Abram st. 30 NY

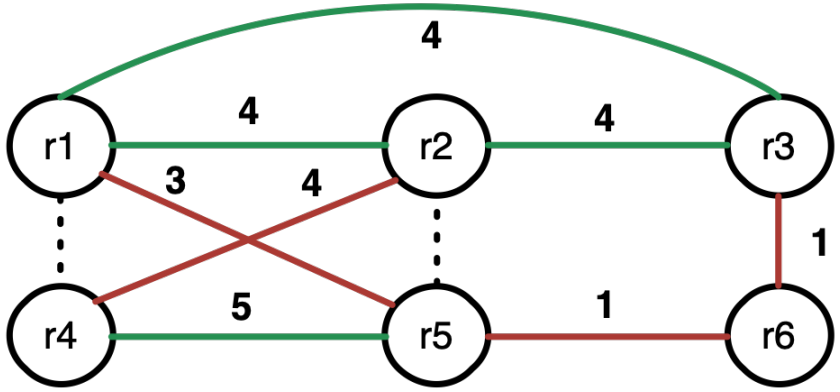
street	retail	85	car
r1 r5	r4 r2	r4 r2	r1 r3 r2

Jon	Jr	auto	Abram
r2 r3	r1 r3 r2	r5 r6	r1 r4 r2 r5

main	Smith	1985	NY
r1 r3 r2	r4 r5	r1 r5	r4 r5

30	st	Ellen	Seller
r4 r5	r4 r2	r4 r5	r1 r3

Apply a weigh-based threshold to prune the graph



Recall	Precision
1.0	0.5

Graph Building

For every block:

- for every entity → add a node
- for every pair of co-occurring entities → add an undirected edge

Blocking graph:

- It eliminates all redundant comparisons → no parallel edges
- Low materialization cost → implicit materialization through inverted indices
- Different from similarity graph!

Edge Weighting

Five generic, attribute-agnostic weighting schemes that rely on the following evidence:

- the number of blocks shared by two entities
- the size of the common blocks
- the number of blocks or comparisons involving each entity.

Computational Cost:

- In theory, equal to executing all pair-wise comparisons in the given block collection.
- In practice, significantly lower because it does not employ string similarity metrics.

Weighting Schemes

1. Aggregate Reciprocal Comparisons Scheme (ARCS)

$$w_{ij} = \sum_{b_k \in B_{ij}} \frac{1}{||b_k||}$$

2. Common Blocks Scheme (CBS)

$$w_{ij} = |B_{ij}|$$

3. Enhanced Common Blocks Scheme (ECBS)

$$w_{ij} = |B_{ij}| \cdot \log \frac{|B|}{|B_i|} \cdot \log \frac{|B|}{|B_j|}$$

4. Jaccard Scheme (JS)

$$w_{ij} = \frac{|B_{ij}|}{|B_i| + |B_j| - |B_{ij}|}$$

5. Enhanced Jaccard Scheme (EJS)

$$w_{ij} = \frac{|B_{ij}|}{|B_i| + |B_j| - |B_{ij}|} \cdot \log \frac{|V_G|}{|v_i|} \cdot \log \frac{|V_G|}{|v_j|}$$

NOTATION

B: *the blocking collection*

$||b_k||$: *#comparisons from a single block*

$|b|$: #profiles in a block b

B_{ij}: *the set of blocks shared by profiles i and j*

Graph Pruning Strategies

- Weighted Edge Pruning (WEP)
  - threshold: average weight across all edges
- Cardinality Edge Pruning (CEP)
  - threshold: $K = BPE \cdot |E|/2$

BPE = AVG number of blocks per entity

- Weighted Node Pruning (WNP)
  - threshold: for each node, the average weight of the adjacent edges
- Cardinality Node Pruning (CNP)
  - threshold: for each node, k=BPE-1

Which strategy is good for the task?

- Local/Global
  - Local strategies are preferable when all entities has the same “value” to the application
    - Each profile has some comparison
    - E.g., for fraud detection all the user profile are “tested” against their most likely match
  - Global strategies are preferable when we discover the highest number of duplicate for a given budget (e.g., time)

- Top-k/Threshold-based
  - Top-k strategies allows to control the number of comparisoins (i.e., the budget)
  - Threshold-based strategies achive in general the best PC

Back to the Example

DBPedia 3.0rc ↔ DBPedia 3.4

Brute-force approach

Comparisons: $2.58 \cdot 10^{12}$
Recall: **100%**
Running time: 1,344 days → **3.7 years**

Token Blocking + Block Filtering + Comparison Propagation

Overhead time: <30 mins
Comparisons: $3.5 \cdot 10^{10}$
Recall: **99%**
Total Running time: **19 days**

Token Blocking + Block Filtering + **Meta-blocking**

Overhead time: 4 hours
Comparisons: $8.95 \cdot 10^{6}$
Recall: **99%**
Total Running time: **10 hours**

Supervised Meta-blocking

- Goal:
  - more accurate and comprehensive methodology for
  - pruning the edges of the blocking graph
- Solution:
  - model edge pruning as a classification task per edge
  - two classes: “likely match”, “unlikely match”
- Similar to a “matcher”, that actually classifies true/false, but has to be lightweight:
  - Classification Features cannot be text similarity
    - Too expensive
  - Classification Algorithms & Configuration have to be fast
    - Low classification cost
  - Training set
    - Should be small, we cannot ask too many labels to the users

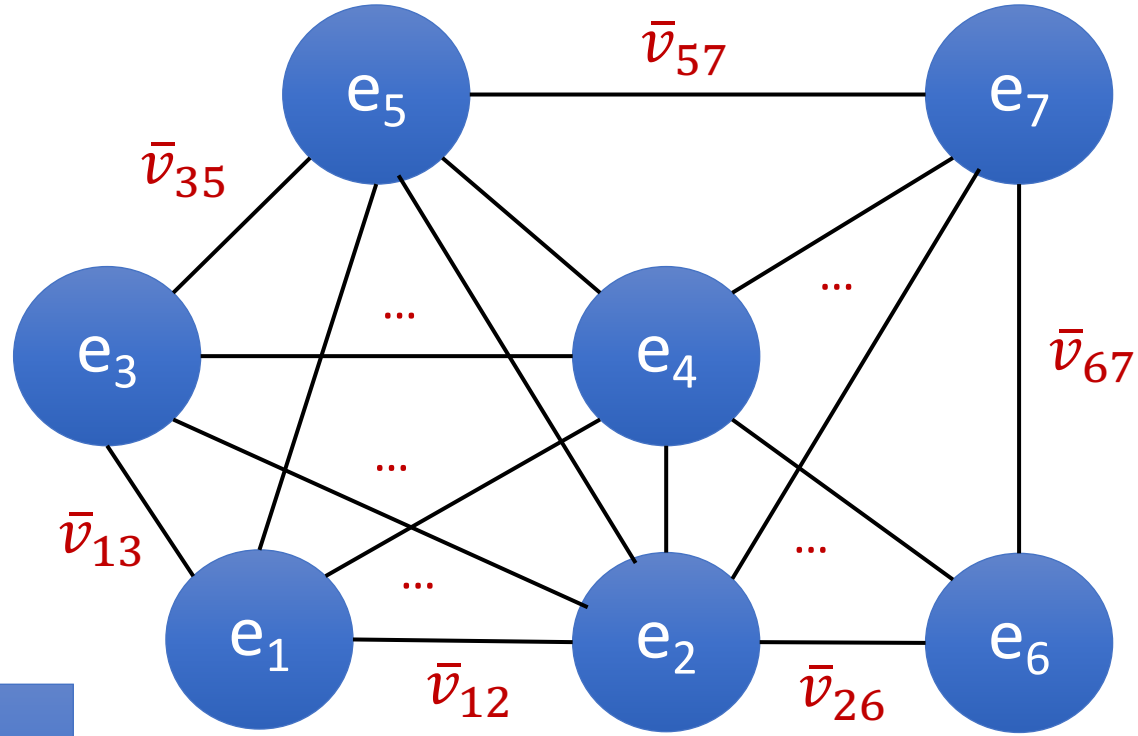
Features

$$EJS(c_{i,j}) = JS(c_{i,j}) \cdot \log \frac{||B||}{||e_i||} \cdot \log \frac{||B||}{||e_j||}$$

$$WJS(c_{i,j}) = \frac{\sum_{b \in B_i \cap B_j} \frac{1}{||b||}}{\sum_{b \in B_i} \frac{1}{||b||} + \sum_{b \in B_j} \frac{1}{||b||} - \sum_{b \in B_i \cap B_j} \frac{1}{||b||}}.$$

$$RS(c_{i,j}) = \sum_{b \in B_i \cap B_j} \frac{1}{|b|}$$

$$NRS(c_{i,j}) = \frac{\sum_{b \in B_i \cap B_j} \frac{1}{|b|}}{\sum_{b \in B_i} \frac{1}{|b|} + \sum_{b \in B_j} \frac{1}{|b|} - \sum_{b \in B_i \cap B_j} \frac{1}{|b|}}.$$



$\bar{v}_{13}$ = [CF-IBF$_{1,2}$, RACCB$_{1,2}$, JS$_{1,2}$, …]

Features

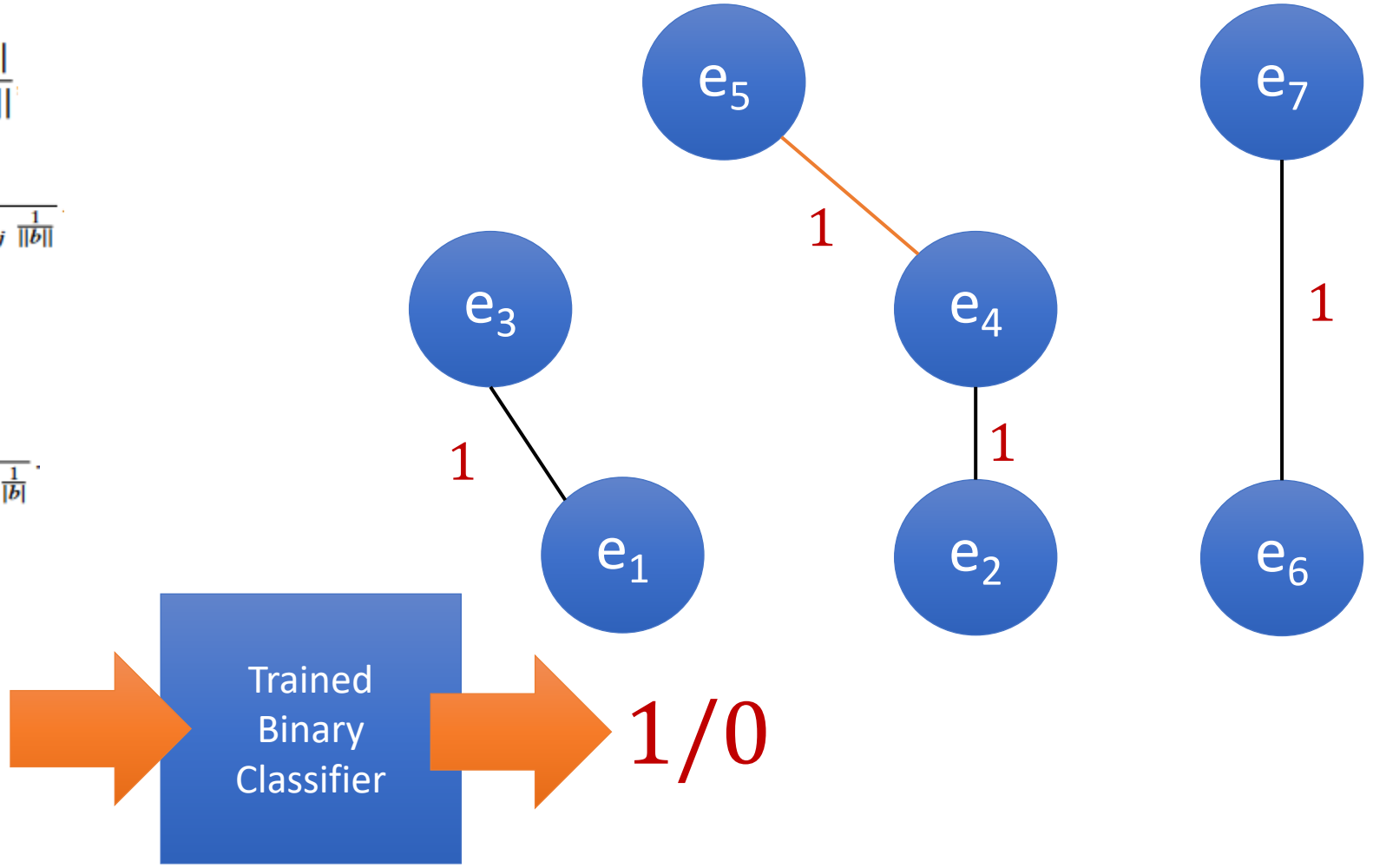
$$EJS(c_{i,j}) = JS(c_{i,j}) \cdot \log \frac{||B||}{||e_i||} \cdot \log \frac{||B||}{||e_j||}$$

$$WJS(c_{i,j}) = \frac{\sum_{b \in B_i \cap B_j} \frac{1}{||b||}}{\sum_{b \in B_i} \frac{1}{||b||} + \sum_{b \in B_j} \frac{1}{||b||} - \sum_{b \in B_i \cap B_j} \frac{1}{||b||}}.$$

$$RS(c_{i,j}) = \sum_{b \in B_i \cap B_j} \frac{1}{|b|}$$

$$NRS(c_{i,j}) = \frac{\sum_{b \in B_i \cap B_j} \frac{1}{|b|}}{\sum_{b \in B_i} \frac{1}{|b|} + \sum_{b \in B_j} \frac{1}{|b|} - \sum_{b \in B_i \cap B_j} \frac{1}{|b|}}.$$

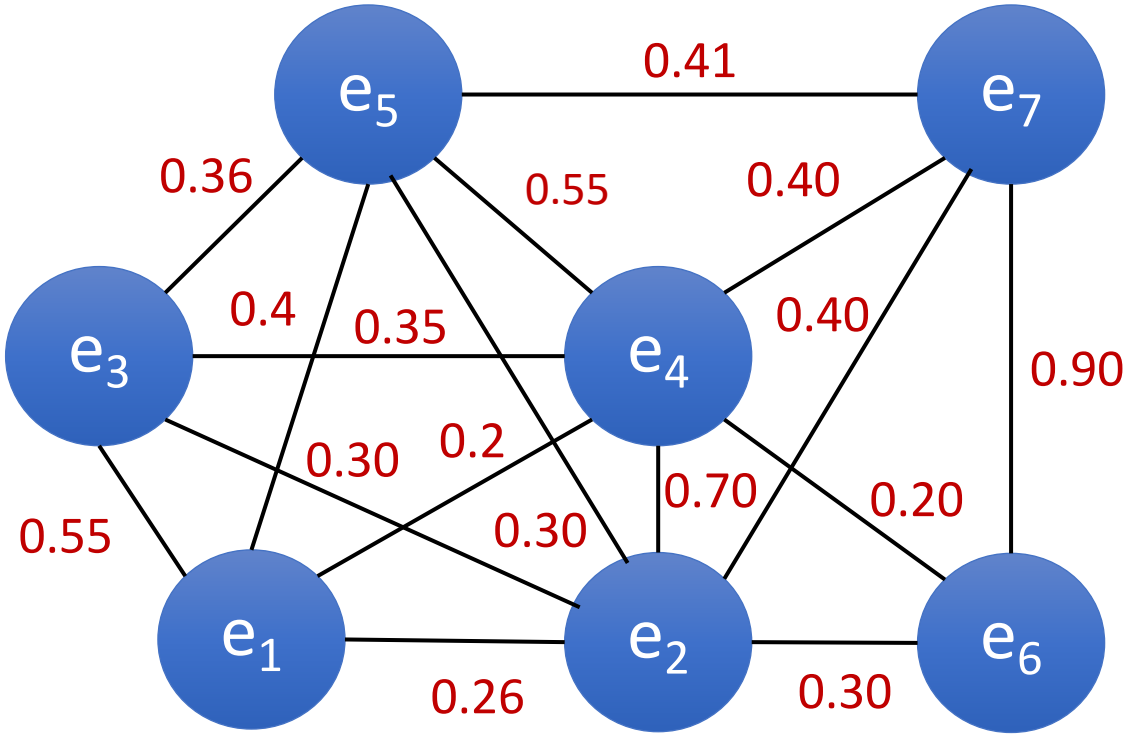
$\bar{v}_{13}$ = [CF-IBF$_{1,2}$, RACCB$_{1,2}$, JS$_{1,2}$, …]



Generalized Supervised Meta-Blocking

Idea

- instead of using the output of the classifier, i.e., 0/1
- Use the “confidence” of the classifier to re-weight the graph



Generalized Supervised Meta-Blocking

Idea: Use meta-blocking over supervised meta-blocking

1. A probabilistic classifier is trained and used to assign each edge with its probability of being a match

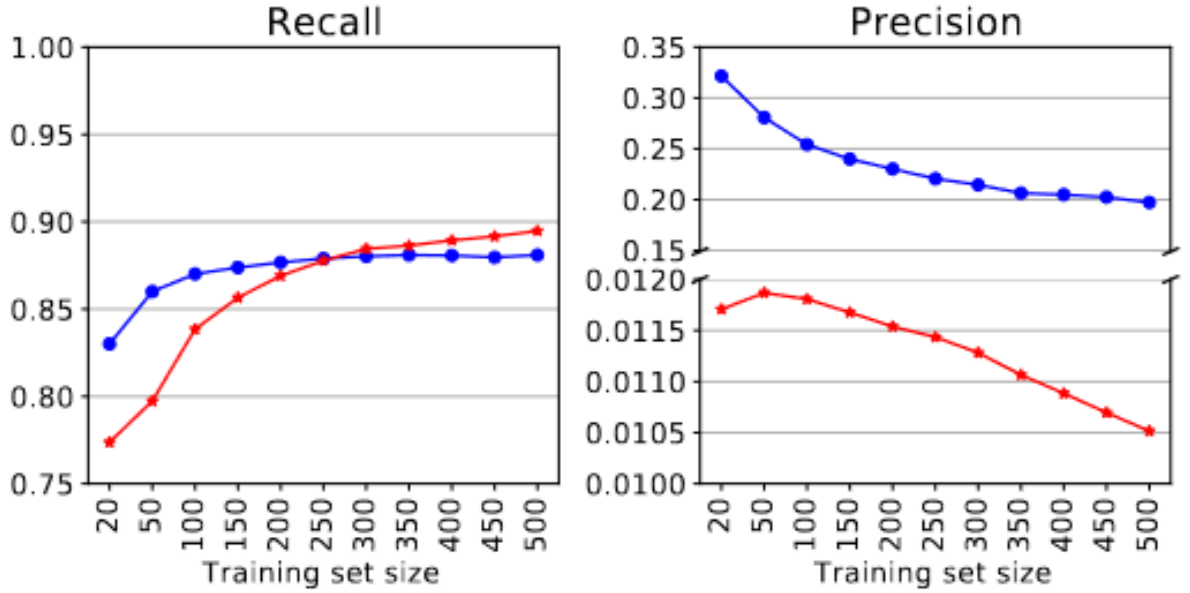
2. The edges with a weight lower than 0.5 are discarded;

3. Unsupervised meta-blocking is applied to the remaining graph

Example on **benchmark datasets**:

- Works well with just 100 labels

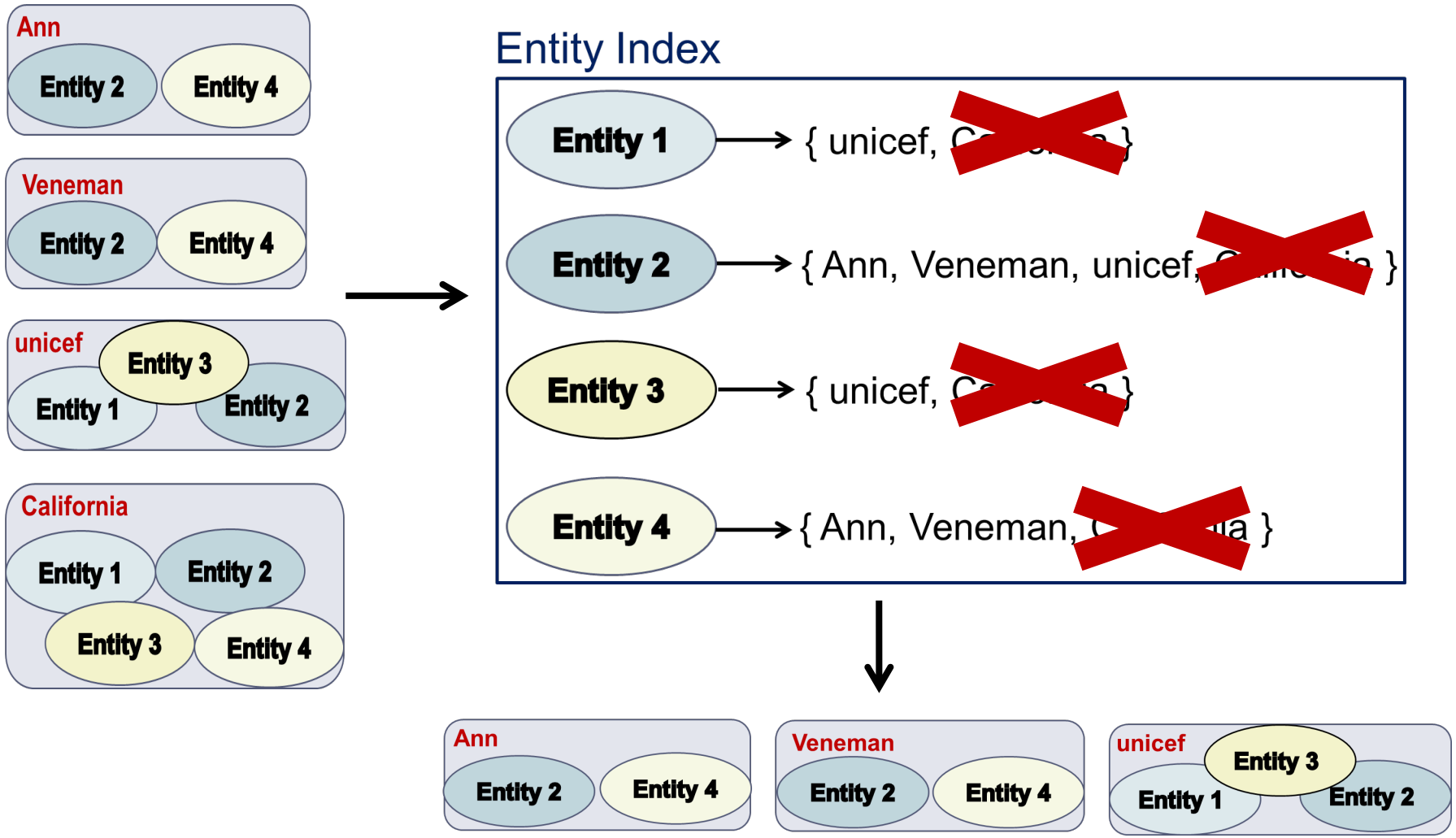
Pruning strategy: BCl BLAST



Other blocking processing techniques

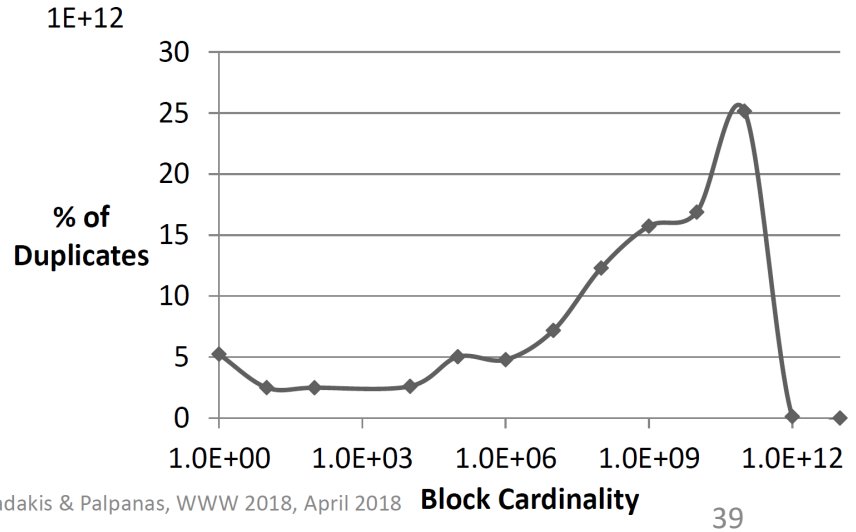
- Block Filtering
  - Idea
    - each block has a different importance for every entity it contains.
    - Larger blocks are less likely to contain unique duplicates and, thus, are less important.
  - Algorithm
    1. sort blocks in ascending cardinality
    2. build an index: profile → blocks
    3. retain every entity in r% of its smallest blocks
    4. reconstruct blocks

Block Filtering Example



Block Purging

- Removes **oversized blocks** (i.e., many comparisons, no duplicates).
- Discards them by setting an upper limit on the **cardinality** (i.e., number of comparisons) of each block
- E.g., remove stop-words



Blocking processing for Privacy-Preserving Record Linkage

- Scenario: a third party have “blind” access to a income database and a medical database
  - Match the patient with its income, without knowing personal information
  - This can be done with Privacy-preserving Record Linkage (PPRL)
- To avoid frequency-based attack:
  - Guarantee k-anonymity, i.e., each block must contain at least k records

- Block Clustering:
  - restrict the size of every block into $[b_{min}, b_{max}]$
  - Algorithm:
    - merge similar blocks with size lower than b_{min}
    - split blocks with size larger than b_{max}
    - until all blocks have the desired size

Blocking Limits

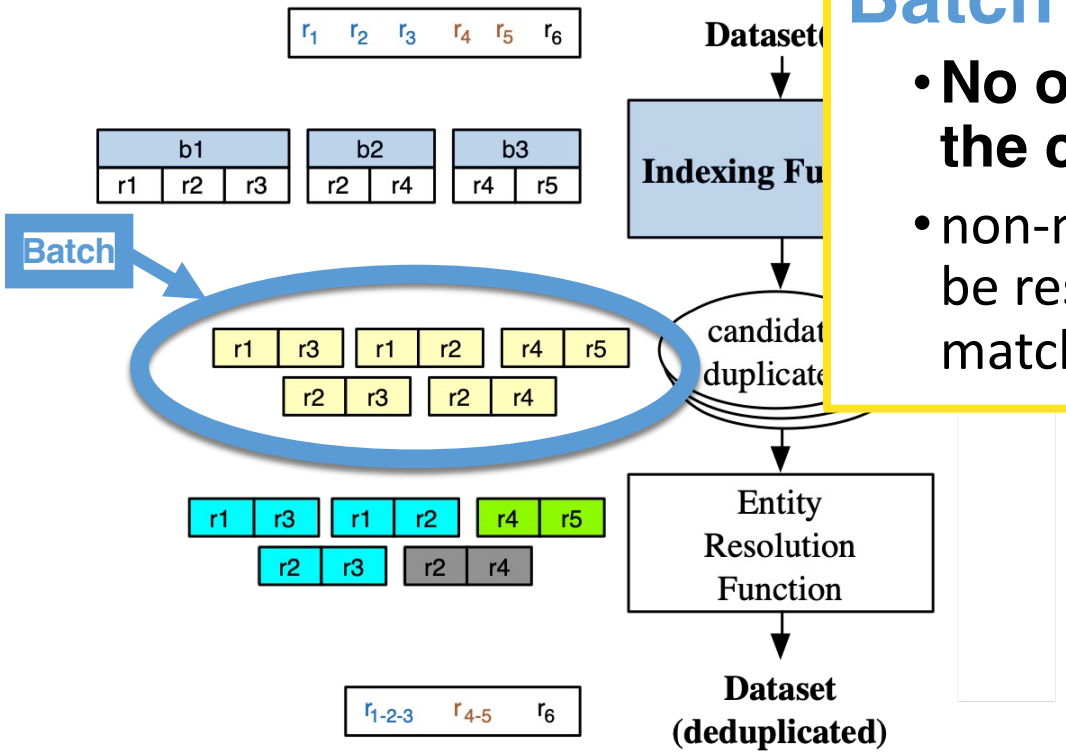
Neme	Profession	Year	Address
John Abram Jr	car seller	1985	main Street

name1	name2	birth yr	job	Loc
Jon Jr	Abram	85	car retail	main st.
Simon	White	75	auto rental	Lincoln LA

full name	b. date	work info	Address
Jon Abrm Jr	31 oct	car seller	main LA
Ellen Smith	21 may	retail	Abrams street 30 NY

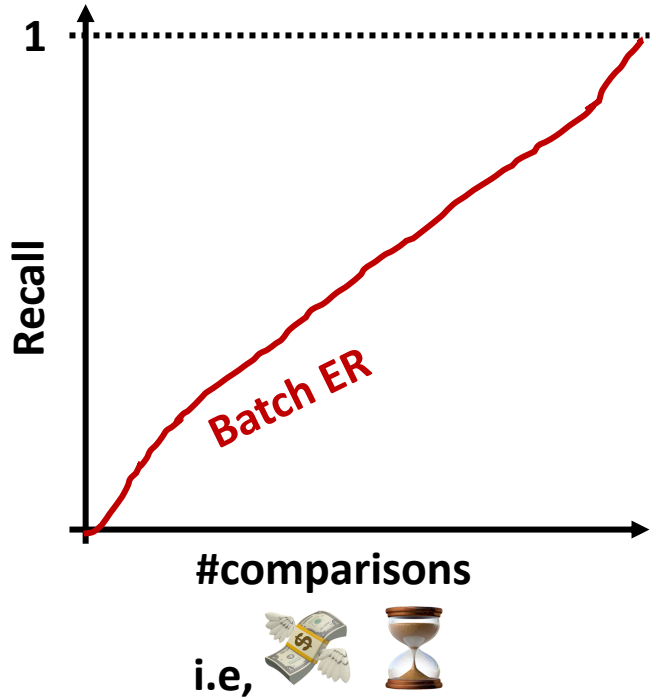
name	name_2	year	occupation	mail
Ellen	Smith	85	auto retail	Abram st. 30 NY

r	name	birth date	work info	Address
r_1 r_2 r_3	John Abram Jr.	31/10/1985	car retail	Main Street LA
r_4 r_5	Ellen Smith	21/05/1985	car retail	Abram Street NY
r_6	Simon White	1/1/1975	car rental	Lincoln Street LA



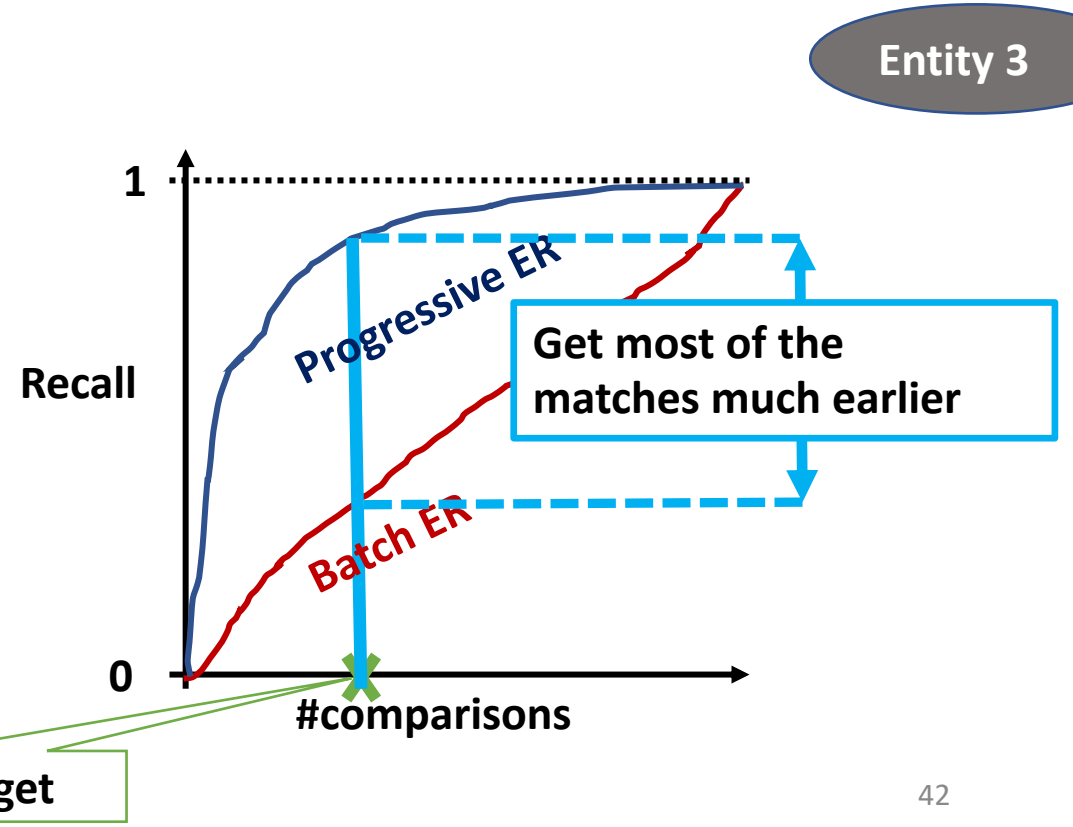
Batch Execution

- **No order for reso the comparisons**
- non-matching pairs be resolved before t matching ones



Pay-as-you-go ER

- aka “Progressive ER” or “on-line”
- Generate candidate pairs with and order
- Assumes there is a budget
  - 💰
  - ⏳
- Tries to maximize the progressive recall
  - Area Under the Curve



Pay-as-you-go ER

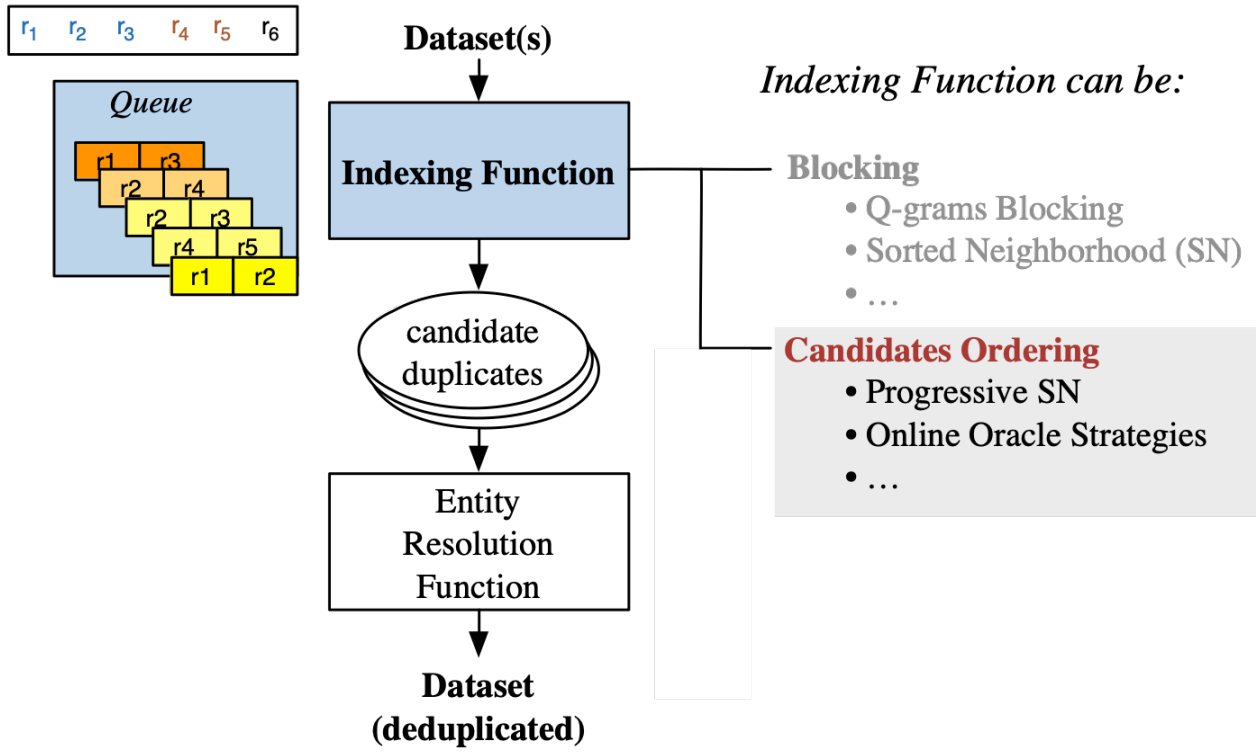
Neme	Profession	Year	Address
John Abram Jr	car seller	1985	main Street

name1	name2	birth yr	job	Loc
Jon Jr	Abram	85	car retail	main st.
Simon	White	75	auto rental	Lincoln LA

full name	b. date	work info	Address
Jon Abrm Jr	31 oct	car seller	main LA
Ellen Smith	21 may	retail	Abrams street 30 NY

name	name_2	year	occupation	mail
Ellen	Smith	85	auto retail	Abram st. 30 NY

r	name	birth date	work info	Address
r_1 r_2 r_3	John Abram Jr.	31/10/1985	car retail	Main Street LA
r_4 r_5	Ellen Smith	21/05/1985	car retail	Abram Street NY
r_6	Simon White	1/1/1975	car rental	Lincoln Street LA



Pay-as-you-go ER

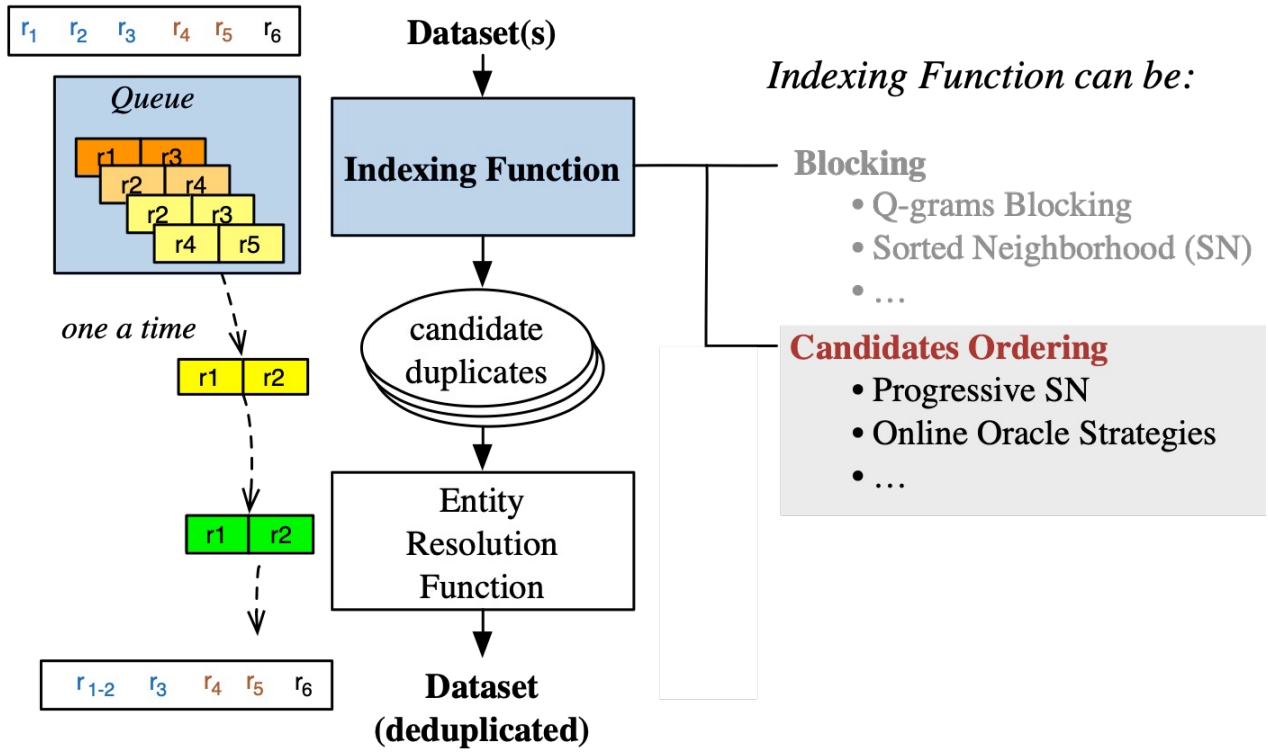
Neme	Profession	Year	Address
John Abram Jr	car seller	1985	main Street

name1	name2	birth yr	job	Loc
Jon Jr	Abram	85	car retail	main st.
Simon	White	75	auto rental	Lincoln LA

full name	b. date	work info	Address
Jon Abrm Jr	31 oct	car seller	main LA
Ellen Smith	21 may	retail	Abrams street 30 NY

name	name_2	year	occupation	mail
Ellen	Smith	85	auto retail	Abram st. 30 NY

r	name	birth date	work info	Address
r_1 r_2 r_3	John Abram Jr.	31/10/1985	car retail	Main Street LA
r_4 r_5	Ellen Smith	21/05/1985	car retail	Abram Street NY
r_6	Simon White	1/1/1975	car rental	Lincoln Street LA



Pay-as-you-go ER

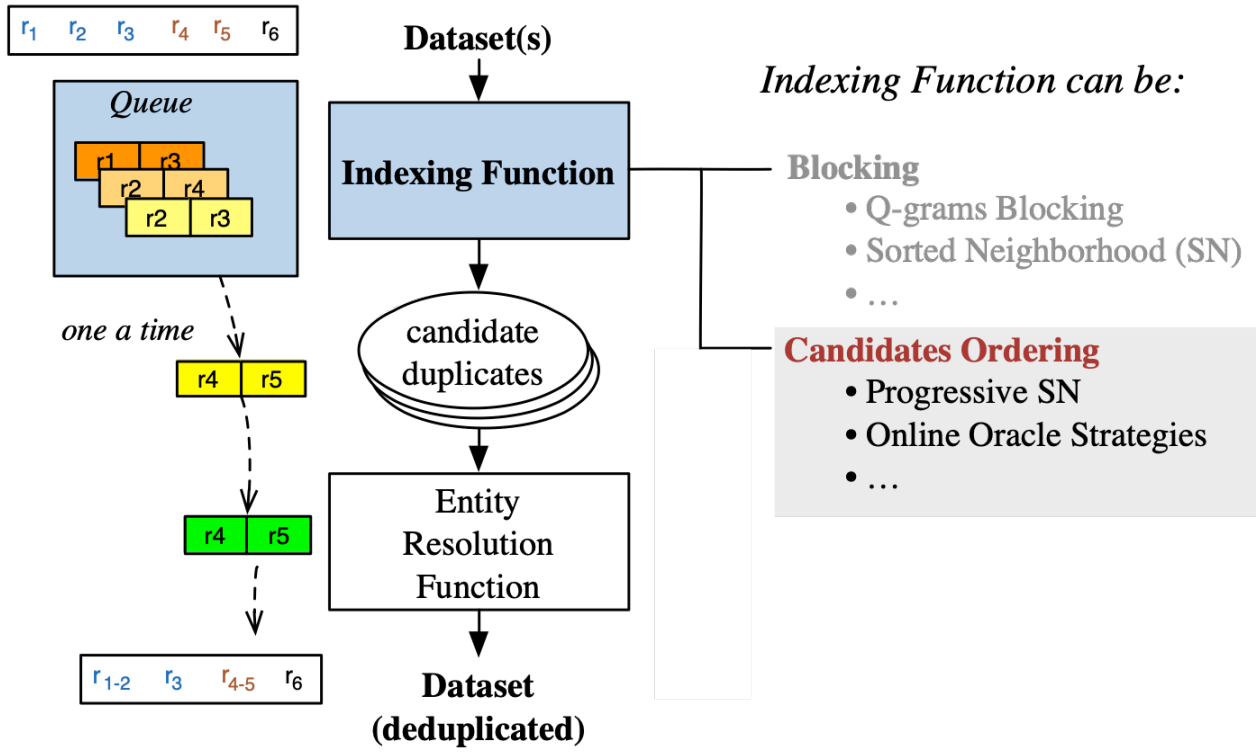
Neme	Profession	Year	Address
John Abram Jr	car seller	1985	main Street

name1	name2	birth yr	job	Loc
Jon Jr	Abram	85	car retail	main st.
Simon	White	75	auto rental	Lincoln LA

full name	b. date	work info	Address
Jon Abrm Jr	31 oct	car seller	main LA
Ellen Smith	21 may	retail	Abrams street 30 NY

name	name_2	year	occupation	mail
Ellen	Smith	85	auto retail	Abram st. 30 NY

r	name	birth date	work info	Address
r_1 r_2 r_3	John Abram Jr.	31/10/1985	car retail	Main Street LA
r_4 r_5	Ellen Smith	21/05/1985	car retail	Abram Street NY
r_6	Simon White	1/1/1975	car rental	Lincoln Street LA



Pay-as-you-go ER

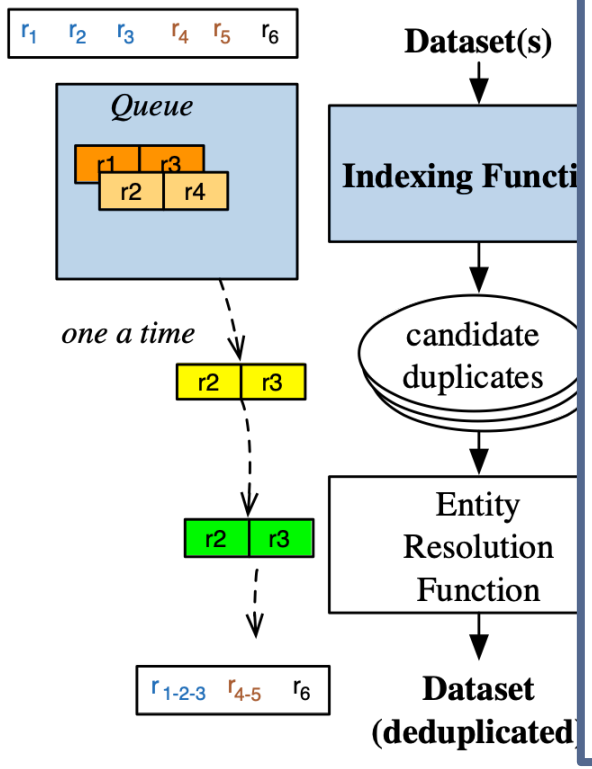
Neme	Profession	Year	Address
John Abram Jr	car seller	1985	main Street

name1	name2	birth yr	job	Loc
Jon Jr	Abram	85	car retail	main st.
Simon	White	75	auto rental	Lincoln LA

full name	b. date	work info	Address
Jon Abrm Jr	31 oct	car seller	main LA
Ellen Smith	21 may	retail	Abrams street 30 NY

name	name_2	year	occupation	mail
Ellen	Smith	85	auto retail	Abram st. 30 NY

r	name	birth date	work info	Address
r_1 r_2 r_3	John Abram Jr.	31/10/1985	car retail	Main Street LA
r_4 r_5	Ellen Smith	21/05/1985	car retail	Abram Street NY
r_6	Simon White	1/1/1975	car rental	Lincoln Street LA



Pro

- same final result of the batch blocking

Challenges

1. how to generate "order"
2. how to generate it efficiently

3 types of ordering

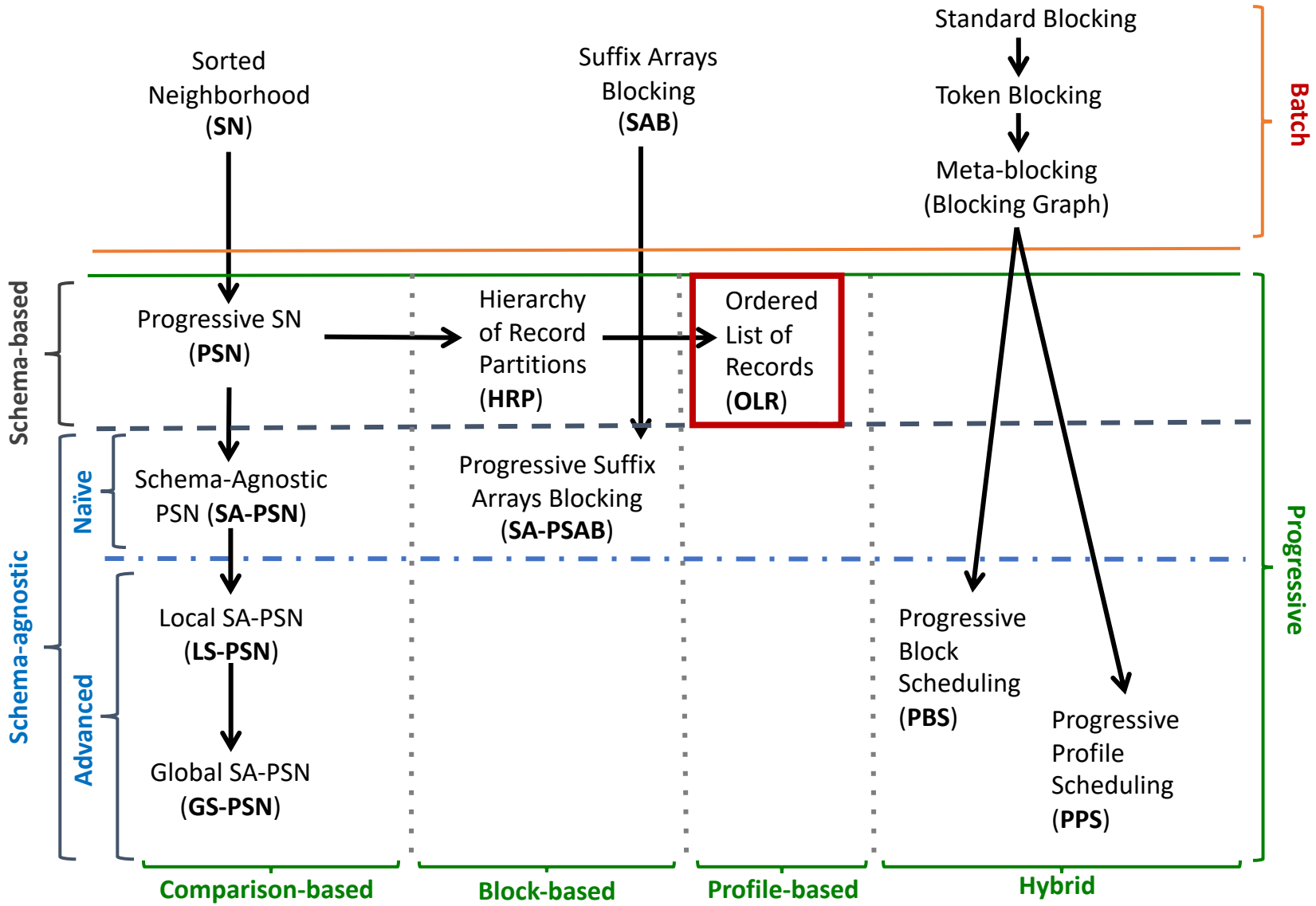
Starting from a blocking schema

1. Ordering at the **Comparisons level**
   - **generates a sorted list of all possible comparisons**
   + high control on what to compare (best progressiveness)
   - may be expensive to achieve

2. Ordering at the **Block level**
   - **generates a sorted list of all the blocks (then resolve each block)**
   + good control on what to compare
   - may be worse than comparisons level

3. Ordering at the **Entity (record) level**
   - **generates a sorted list of all the entities (then resolve each entity)**
   + cheapest to manage (for scalability)
   - not really good for Clean-Clean ER

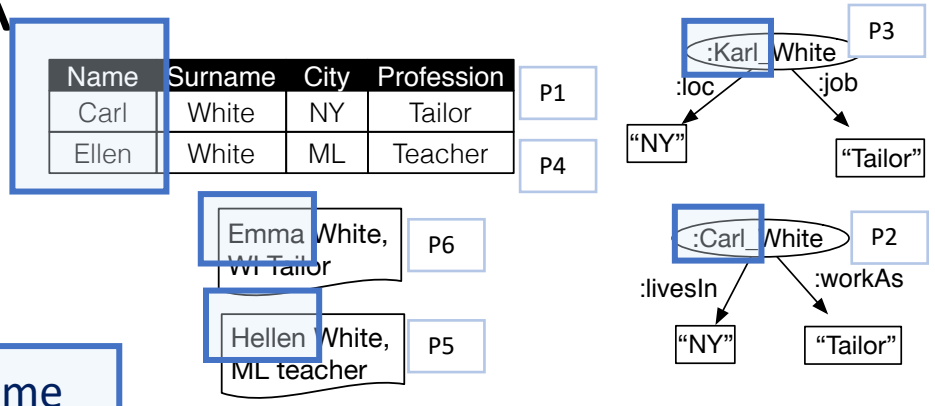
Taxonomy of Progressive Methods



Simonini G, Papadakis G, Palpanas T, Bergamaschi S. Schema-agnostic progressive entity resolution. IEEE Transactions on Knowledge and Data Engineering. 2018 Jul 4;31(6):1208-21.

Progressive **Schema-based** ER

- Progressive Sorted Neighborhood (**PSN**)
  [1][2] **State-of-the-art**



1. Defines a sorting key

Sorting Key = Name

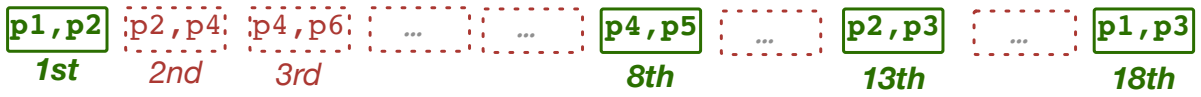
2. Orders the dataset accordingly
   - *Neighbor List*

```
['carl','ellen','emma','hellen','karl']
```

3. For **w** in = 2..size(*Neighbor List*)
   i. Slides a window of size w over the Neighbor List
   ii. Emits each pair of profiles at distance **w-1**

w = 6

p1, p2, p4, p6, p5, p3



[1] Whang et al. "Pay-as-you-go entity resolution." IEEE TKDE 2013
[2] Papenbrock et al. "Progressive duplicate detection." IEEE TKDE 2015

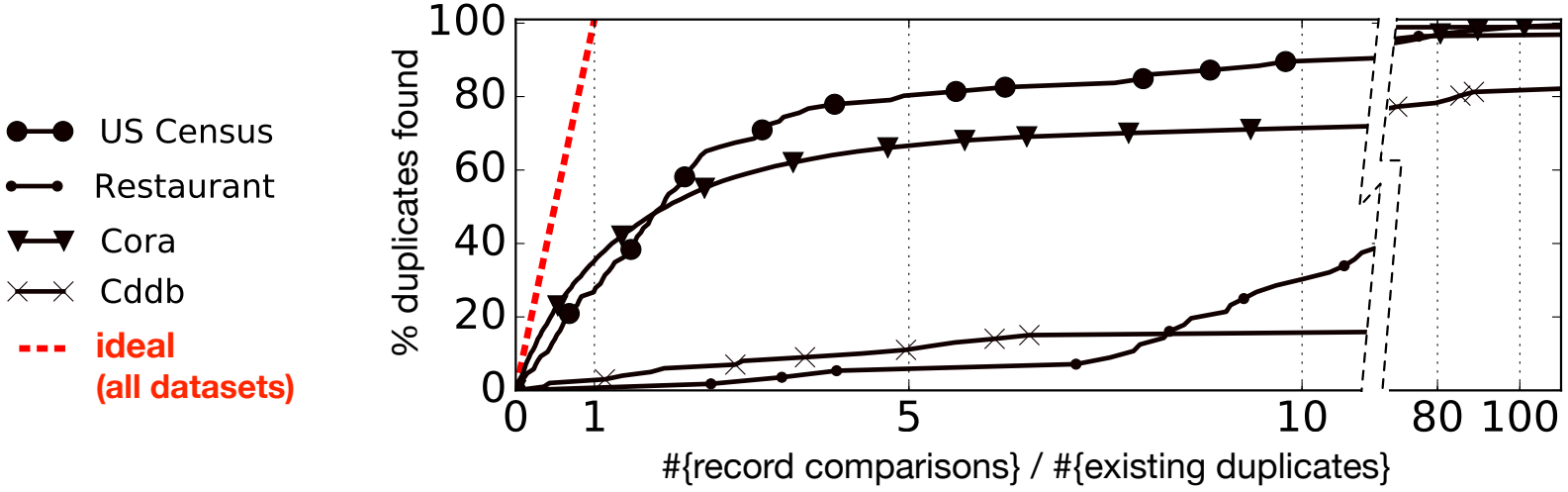
Progressive ER: Limits of the Schema-based methods

1. **Schema-alignment** for multiple, heterogeneous data sources
   - Expensive task to perform **before** the ER (Prohibitive in a pay-as-you-go context)

2. Either **domain experts or labelled data** (and an ML algorithm) to define the "*keys*" to generate the candidate pairs are required

- Even when these requirements are satisfied:
  - There is plenty of **room for improvement**

State-of-the-art
PSN
On benchmark datasets



The Schema-agnostic Approach (our)

- **Define order w/o using the schema information**
  - No need for the schema-alignment
  - No need for attribute-based keys definition
  - **Inherently addresses the Variety of Big Data**

- **2 efficient schema-agnostic methods:**
  - **LS/GS-PSN**
  - **PBS**
  - **address also the Volume of Big Data**

Naïve solution does not work

Sorting Key = Every Token

```
['carl','ellen','emma',
'hellen','karl','ml','ny',
'tailor','teacher','white']
```

Coincidental

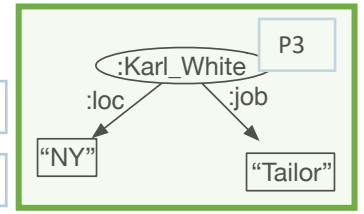
Name	Surname	City	Profession
Carl	White	NY	Tailor
Ellen	White	ML	Teacher

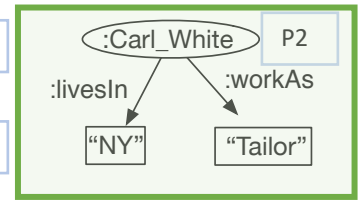
P1

P4

Emma White, WI Tailor — P6

Hellen White, ML teacher — P5





w = 2

Repeated

Neighbor List

p1, p2, p4, p6, p5, p3, p4, p5, p2, p1, p3, p6, p2, p3, p1, p5, p4, p1, p2, p3, p4, p5, p6, p6

Each profile appears in the list as many times as the number of its tokens

Idea:

- Consider every token as a sorting key
- Apply PSN (as before)

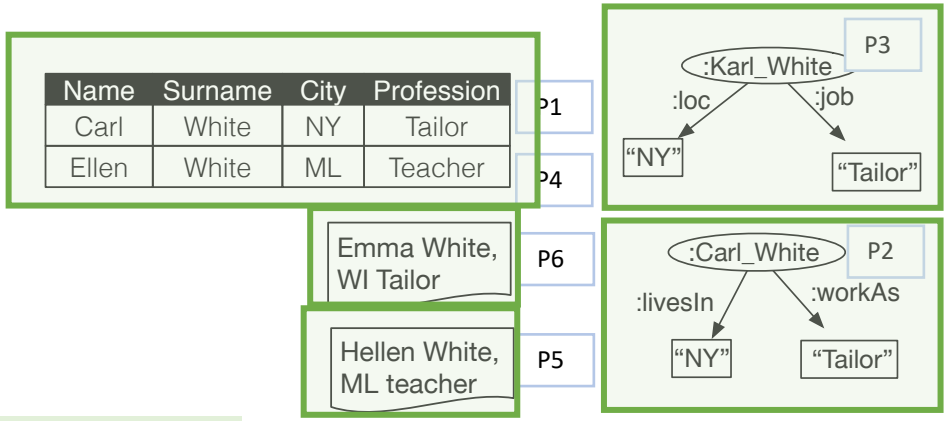
Low Progressive Recall:

- **Repeated Emissions (*redundancy*)**
- **Coincidental Proximity**

Local Schema-agnostic PSN (**LS-PSN**)

Sorting Key = Every Token

```
['carl','ellen','emma',
'hellen','karl','ml','ny',
'tailor','teacher','white']
```



(for w = 2..#NeighborList)

w = 2

Neighbor List

p1, p2, p4, p6, p5, p3, p4, p5, p2, p1, p3, p6, p2, p3, p1, p5, p4, p1, p2, p3, p4, p5, p6,p6

Position Index

p	Positions
p1	1, 10, 15,…
p2	2, 9…
…	…
p6	…20, 21

Pair	Count
p1,p2	1+1+1
p2,p4	1
p4,p6	1
p5,p6	1+1
…	…
p2,p3	1+1+1

Pair Weighting

	p2	p3	p4	p5	p6
p1	.6	.3	.1	.1	.1
p2		.6	.1	.1	.1
p3			.3	.1	.1
p4				.6	.1
p5					.3

Sorting

p1,p2	*1st*
p2,p3	*2nd*
p4,p5	*3rd*
p1,p3	*4th*

w = 3..

Idea:

- To exploit the redundancy of the schema-agnostic Neighbor List

Local/Global Schema-agnostic PSN

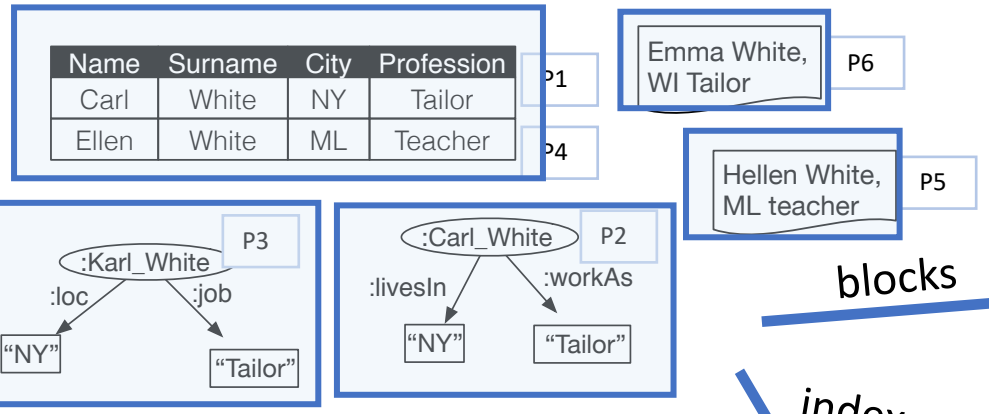
LS-PSN

- **PRO:**
  - **Remove redundancy within the same window**
  - **Progressive Recall**
- **Cons:**
  - **Redundant Comparisons among windows**

GS-PSN

- Similar to LS-PSN, but defines a **global** execution order for all comparisons in a range of window sizes up to w_{max}
- **PRO:**
  - **Progressive Recall**
  - **No redundancy**
- **Cons:**
  - **Higher overhead**
  - **W_{max} Parameter**

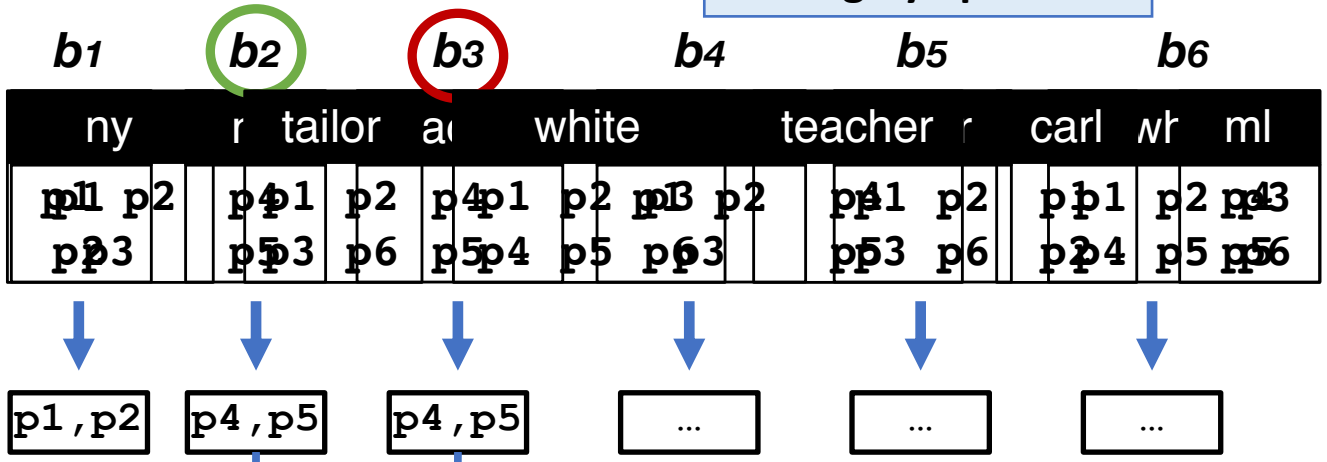
Progressive Block Scheduling (PBS)



blocks

index

Sorting by #pairs



Profile Index

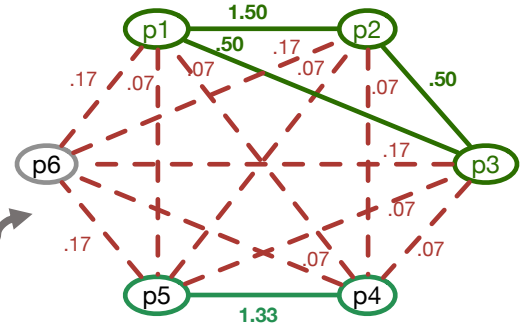
p	Blocks
p1	b1, b4, b5, b6
...	...
p4	b2, b3, b6
p5	b2, b3, b6
...	...

LeCoBI

LeCoBI

Order within each block exploiting the **Blocking graph**

Efficiently built using the Profile Index



Std. Blocking

Block key = Every Token

Idea:

- the **smaller** a block is
- the more **distinctive** information it encapsulates
- the more **likely** it is to **contain duplicate** profiles

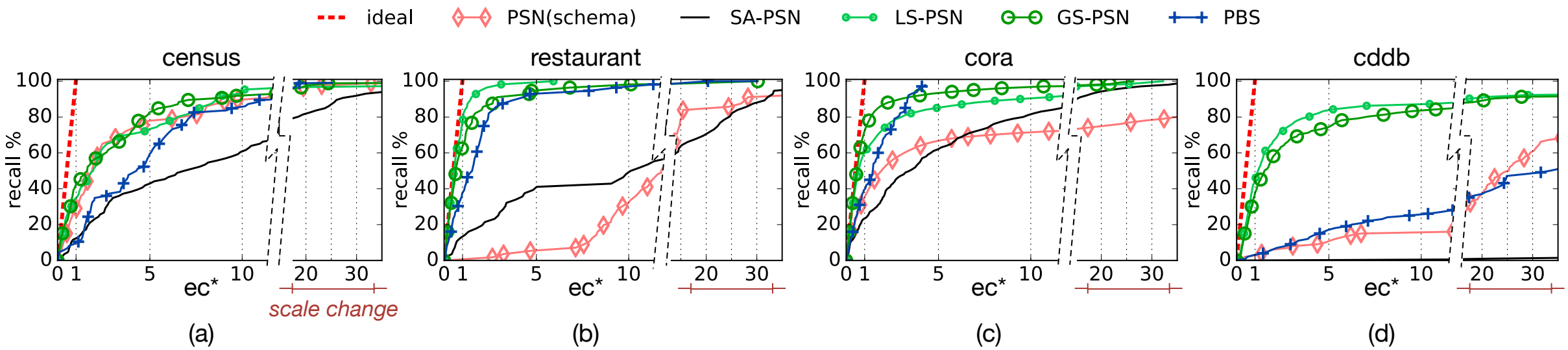
Experiments

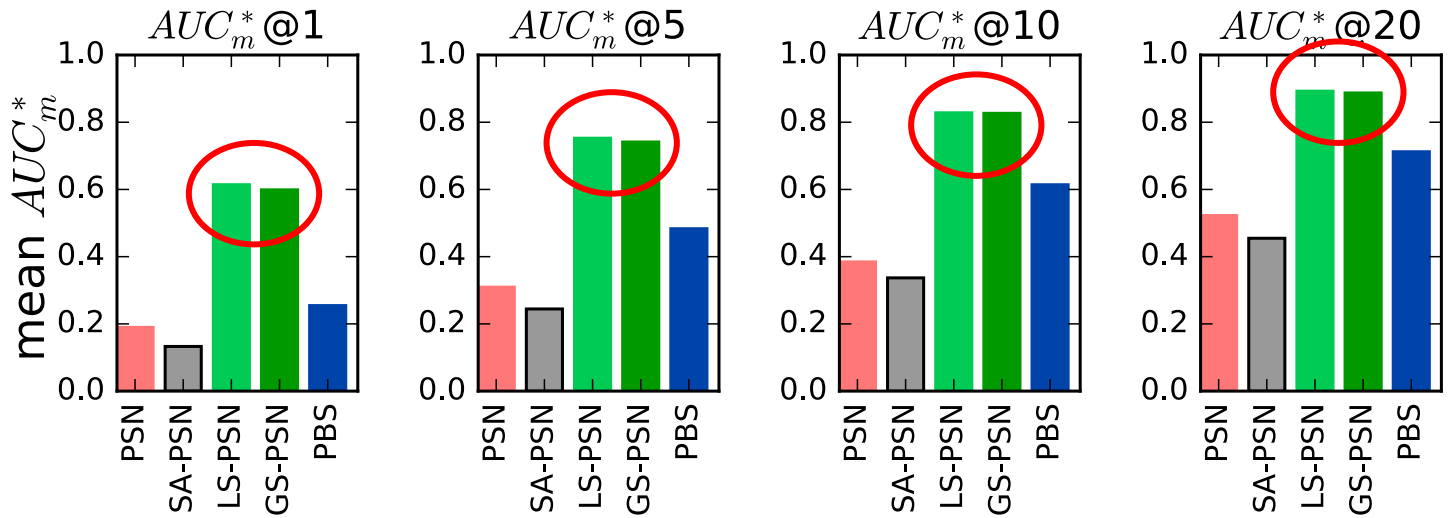
TABLE I

DATASET CHARACTERISTICS: ER TYPE, NUMBER OF ENTITY PROFILES, NUMBER OF *attribute names*, AND NUMBER OF EXISTING MATCHES.

	ER type	$\lvert P \rvert$	#attr.	$\lvert \mathcal{D}_P \rvert$
Structured Datasets				
census	Dirty ER	841	5	344
restaurant	Dirty ER	864	5	112
cora	Dirty ER	1.3k	12	17k
cddb	Dirty ER	9.8k	106	300
Large, Heterogeneous Datasets				
movies	Clean-clean ER	28k—23k	4—7	23k
dbpedia	Clean-clean ER	1.2M—2.2M	30k—50k	893k
freebase	Clean-clean ER	4.2M—3.7M	37k—11k	1.5M

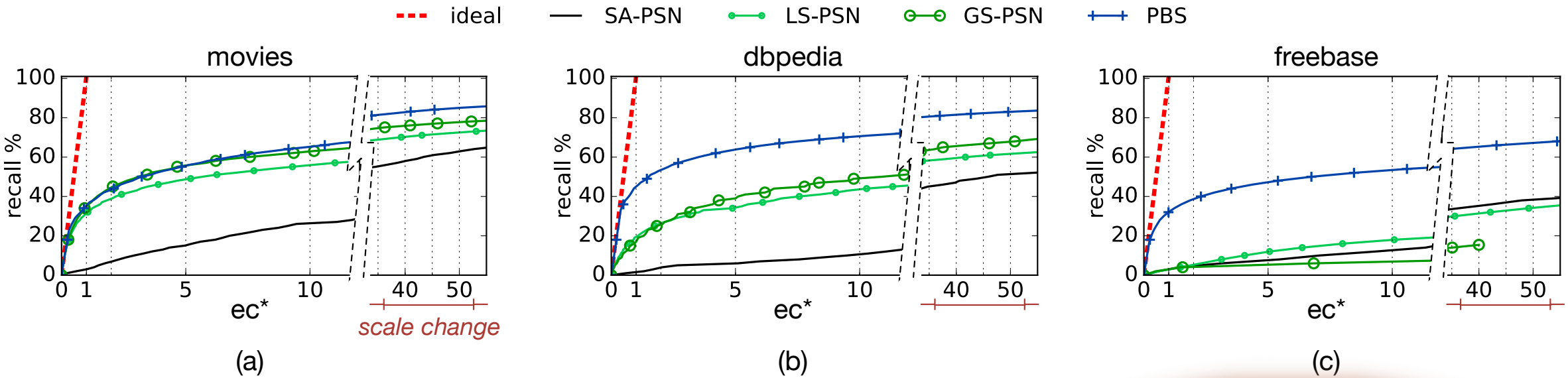
Structured Datasets

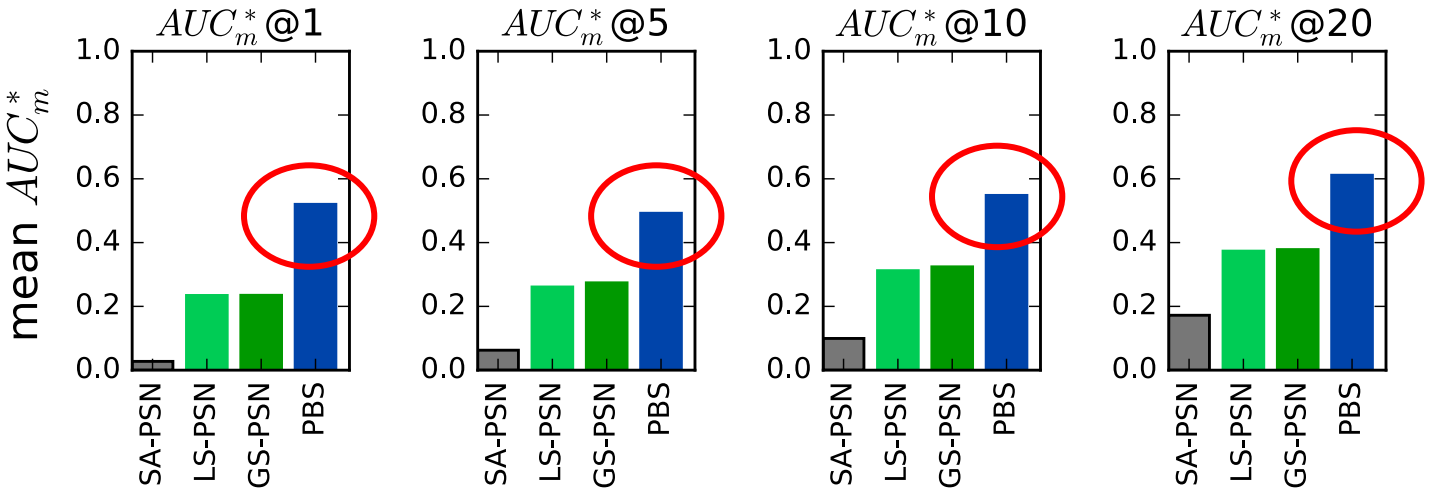




Structured data: with small number of attributes and unique schema for all data, **LS/GS-PSN methods work best**

Large, Heterogeneous Datasets

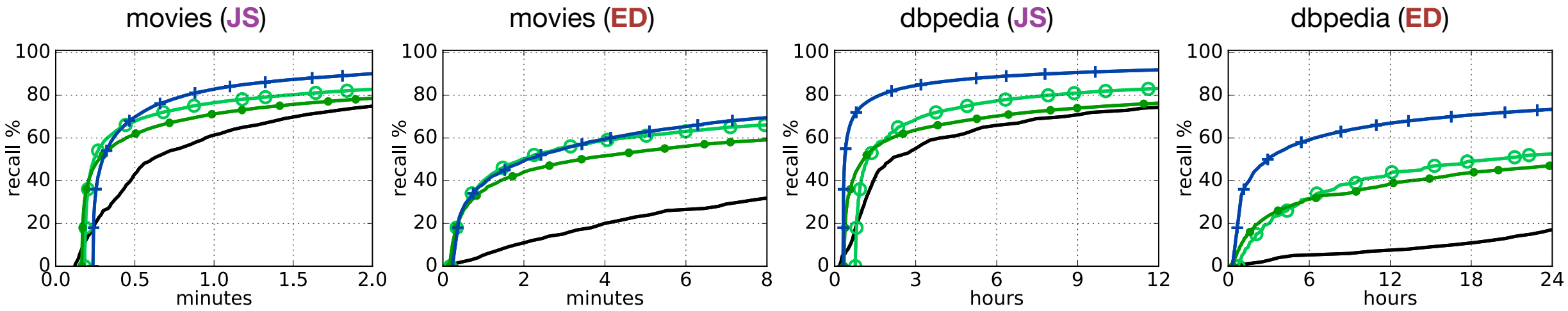




Heterogeneous data: with high number of attributes and heterogeneous schemata, **PBS method works best**

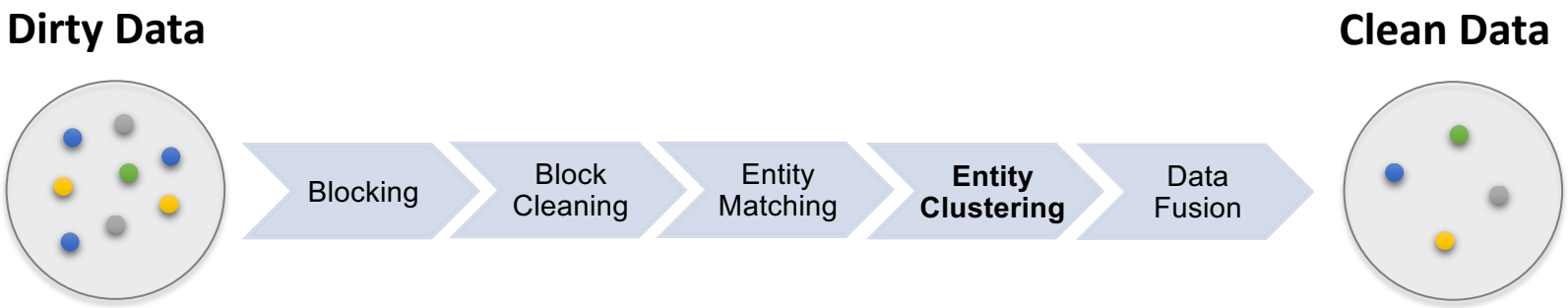
In combination with a Match Function

- **JS**: Match function based on Jaccard Similarity (“cheap”)
- **ED**: Match function based on Edit Distance (“expensive”)



ENTITY CLUSTERING

From pairs of matching profiles to consistent clusters of matches, each one referring to a different entity



Entity Clustering

- Entity Matching provides **pairs of profiles that are identified as true matches**
- These pairs may refer to the same entity
- Entity Clustering partitions matched pairs (correspondences) into equivalence clusters

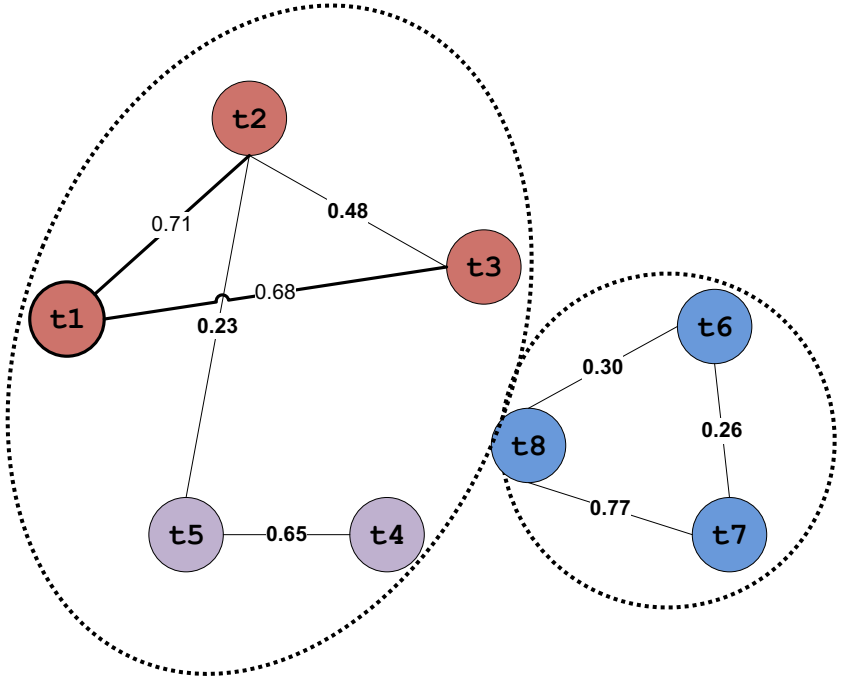
- **Input**
  - Matched pairs of profiles = Similarity Graph:
    - Nodes → Profiles, Edges → Candidate matches, Edge weights → similarity
- **Output**
  - Equivalence Clusters

Entity Clustering Algorithms

- **Connected Components (Transitive Closure)**
- CENTER
- MERGE-CENTER

The standard approach
But

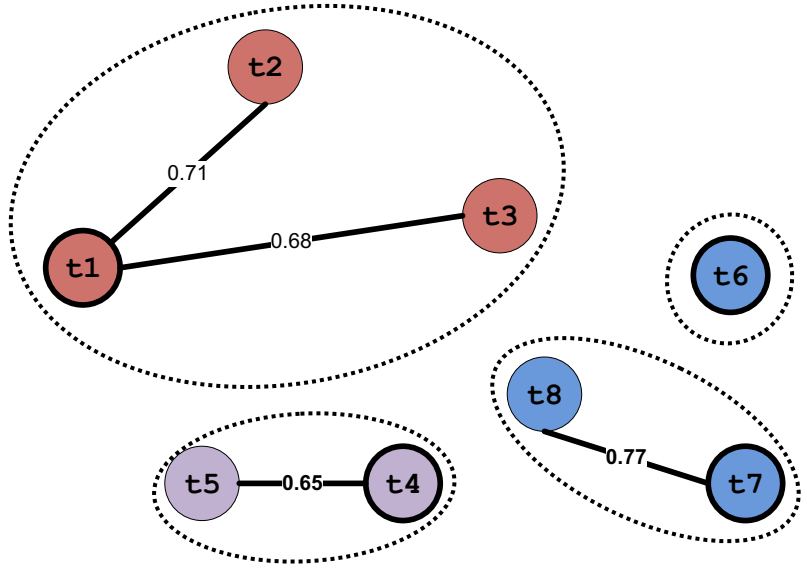
- May put together many dissimilar profiles (low threshold)
- May split many similar profiles (high thresholds)
- Very sensitive to the value of the threshold used for the similarity join



Entity Clustering Algorithms

- Connected Components (Transitive Closure)
- **CENTER**
- MERGE-CENTER

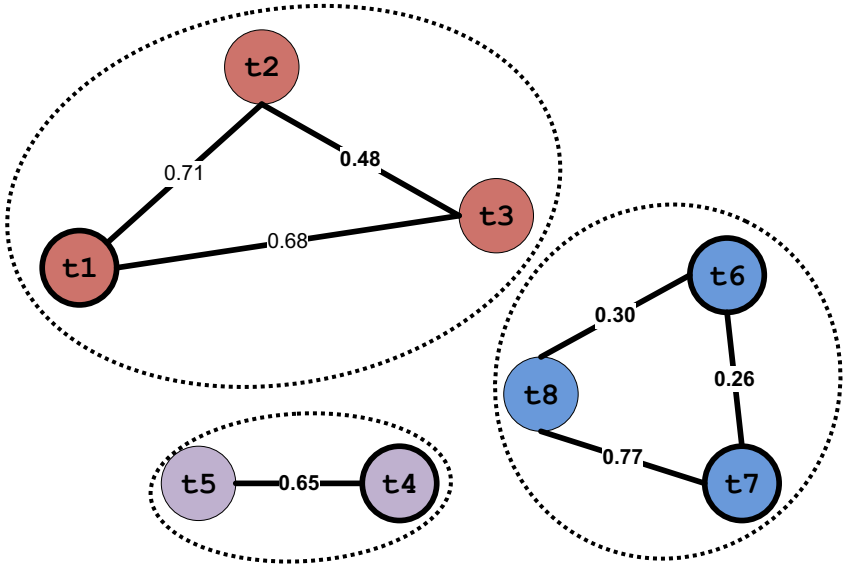
The CENTER algorithm performs clustering
by partitioning the similarity graph
into clusters that have a center,
and all profiles in each cluster are similar
to the center of the cluster.



Entity Clustering Algorithms

- Connected Components (Transitive Closure)
- CENTER
- **MERGE-CENTER**

It performs similar to CENTER, but merges two clusters c_i and c_j whenever a profile similar to the center node of c_j is in the cluster c_i, i.e., a profile that is similar to the center of the cluster c_i is similar to the center of c_j.



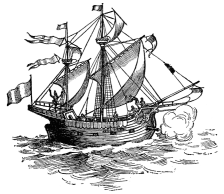
JedAI Toolkit

What is the JedAI Toolkit?

JedAI can be used in three ways:

- As an open source library that implements numerous state-of-the-art methods for all steps of an established end-to-end ER workflow
  - Batch/Progressive
  - Single machine / parallel and distributed (Apache Spark)
  - Complete ER workflow

JedAI vs other tools



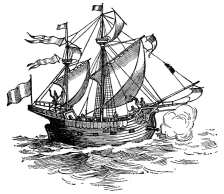
Magellan

× limited variety of (blocking) methods

JedAI

✓rich variety available methods for every step in the end-to-end workflow

JedAI vs other tools



Magellan

× limited variety of (blocking) methods

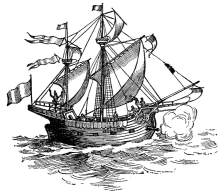
× restricted to relational data only

JedAI

✓ rich variety available methods for every step in the end-to-end workflow

✓ applies to both structured and non-structured data

JedAI vs other tools



Magellan

× limited variety of (blocking) methods

× restricted to relational data only

× targeted to expert users, focusing on development of tailor-made methods

JedAI

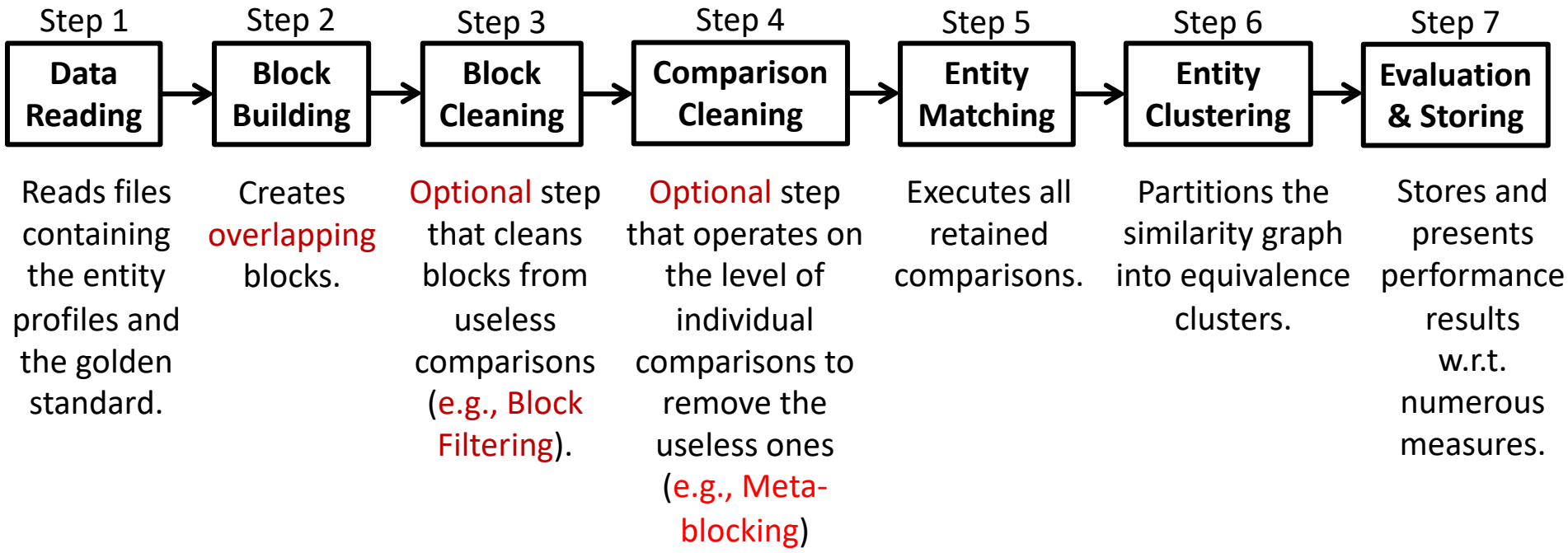
✓ rich variety available methods for every step in the end-to-end workflow

✓ applies to both structured and non-structured data

✓ hands-off functionality through default configuration of every method, but also extensible

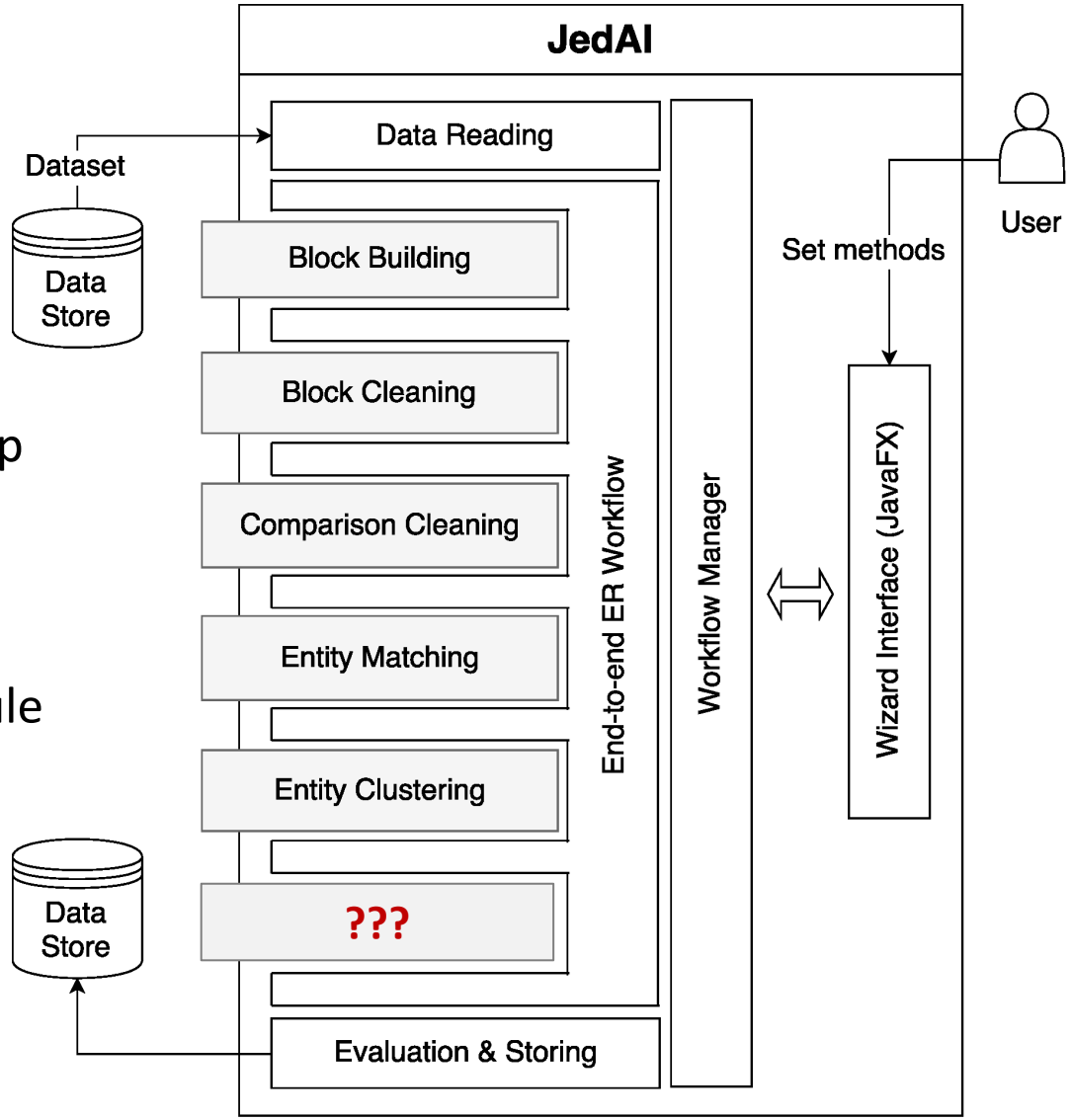
How does the JedAI Toolkit work?

JedAI implements the following schema-agnostic, end-to-end workflow for both Clean-Clean and Dirty ER:



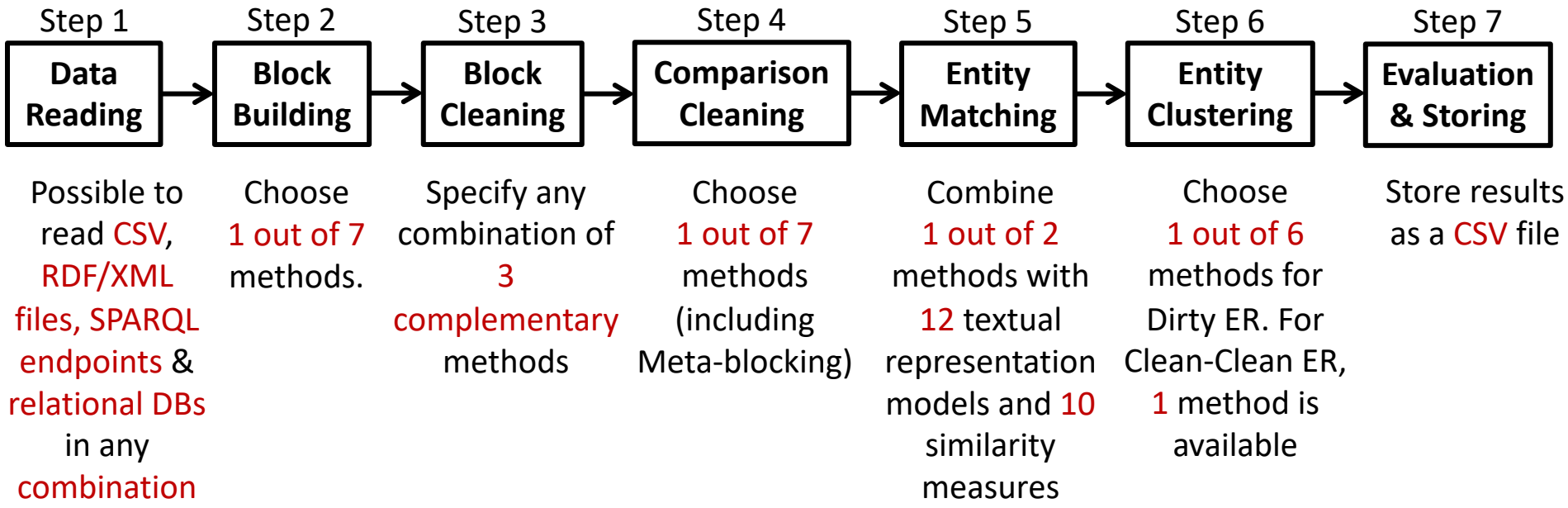
How is the JedAI Toolkit structured?

- Modular architecture
  - one module per workflow step
- Extensible architecture
  - e.g., ontology matching module



How can I build an ER workflow?

JedAI supports several established methods for each workflow step:



Bibliography

1. Christen P. A survey of indexing technigues for scalable record linkage and deduplication. IEEE transactions on knowledge and data engineering. 2011 Jun 16;24(9):1537-55.
2. Papadakis G, Koutrika G, Palpanas T, Nejdl W. Meta-blocking: Taking entity resolutionto the next level. IEEE Transactions on Knowledge and Data Engineering. 2013 Mar 27;26(8):1946-60.
3. Christophides V, Efthymiou V, Palpanas T, Papadakis G, Stefanidis K. End-to-end entity resolution for big data: A survey. arXiv preprint arXiv:1905.06397. 2019 May 15.
4. Elmagarmid AK, Ipeirotis PG, Verykios VS. Duplicate record detection: A survey. IEEE Transactions on knowledge and data engineering. 2006 Nov 30;19(1):1-6.
5. Simonini G, Papadakis G, Palpanas T, Bergamaschi S. Schema-agnostic progressive entity resolution. IEEE Transactions on Knowledge and Data Engineering. 2018 Jul 4;31(6):1208-21.
6. Nauman F, Herschel M. An introduction to duplicate detection. Springer Nature; 2022 Jun 1.
7. Konda PV. Magellan: Toward building entity matching management systems. The University of Wisconsin-Madison; 2018.
8. Getoor L, Machanavajjhala A. Entity resolution: theory, practice & open challenges. Proceedings of the VLDB Endowment. 2012 Aug 1;5(12):2018-9.
9. Ebraheem M, Thirumuruganathan S, Joty S, Ouzzani M, Tang N. DeepER--Deep Entity Resolution. arXiv preprint arXiv:1710.00597. 2017 Oct 2.
10. Fellegi IP, Sunter AB. A theory for record linkage. Journal of the American Statistical Association. 1969 Dec 1;64(328):1183-210.
11. Papadakis G, Mandilaras G, Gagliardelli L, Simonini G, Thanos E, Giannakopoulos G, Bergamaschi S, Palpanas T, Koubarakis M. Three-dimensional entity resolution with jedai. Information Systems. 2020 Nov 1;93:101565.