The five generations of Entity Resolution

OM 2023, Athens
November 7, 2023

George Papadakis

gpapadis@di.uoa.gr

National and Kapodistrian
UNIVERSITY OF ATHENS

Structure Outline

- Introduction
- The First Four Generations
- The Fifth Generation: Leveraging External Knowledge
- Challenges and Final Remarks

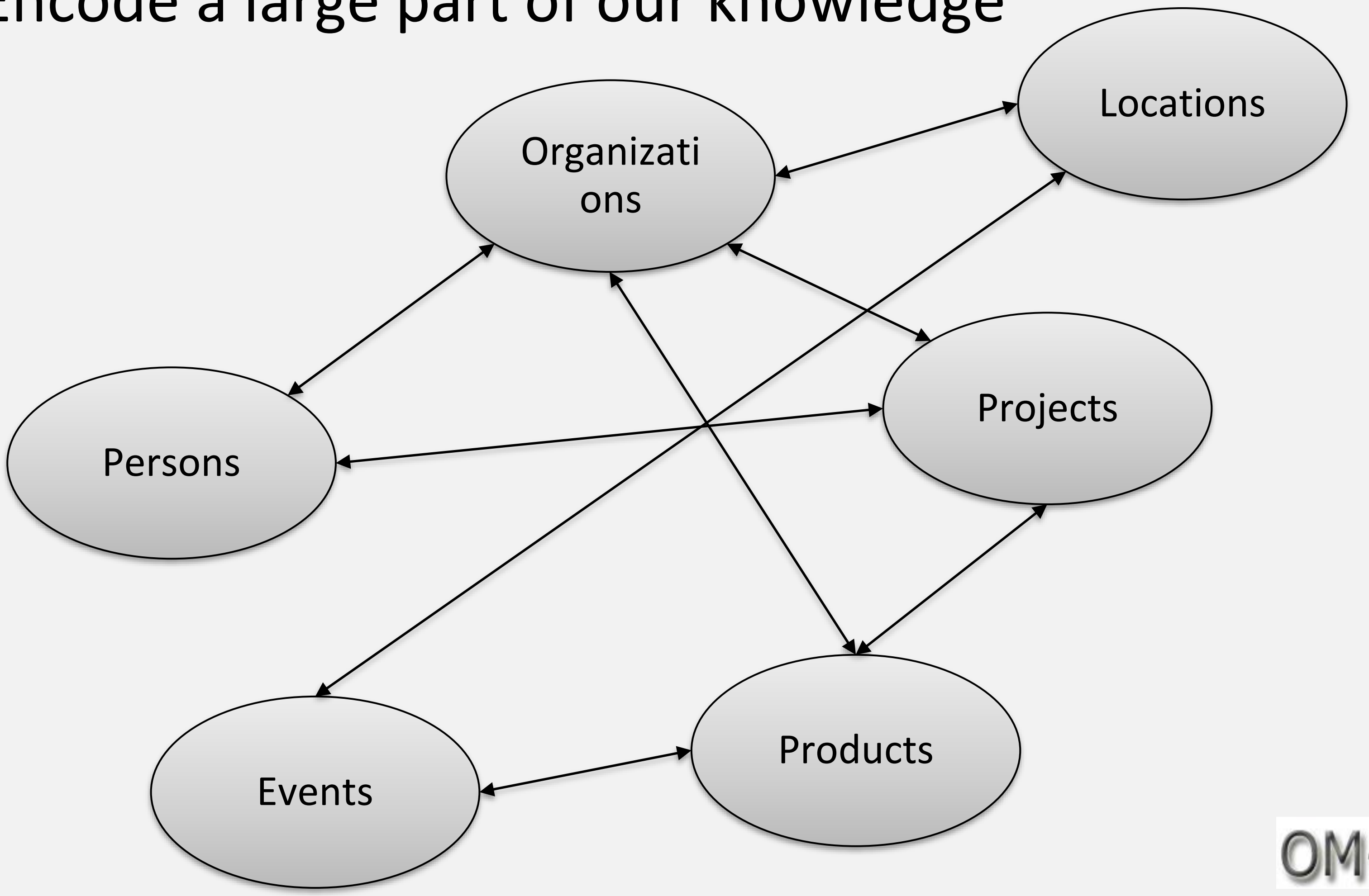
Part A – Introduction

  – Motivation
  – Preliminaries

- The First Four Generations
- The Fifth Generation: Leveraging External Knowledge
- Challenges and Final Remarks

Motivation

- Entities: an invaluable asset for numerous current applications and systems
- Encode a large part of our knowledge



Matching, Linkage, Reconciliation, etc.

- Many names, descriptions, or IDs (URIs) are used for the same real-world "entity"
- Example:

Matching, Linkage, Reconciliation, etc.

- Many names, descriptions, or IDs (URIs) are used for the same real-world “entity”
- Example:

London 런던 ލަންޑަން लंडन लंदन લંડન ለንደን ロンドン লন্ডন ລອນດອນ இலண்டன் ლონდონი Llundain Londain Londe Londen Londen Londen Londinium London Londona Londonas Londoni Londono Londra Londres Londrez Londyn Lontoo Loundres Luân Đôn Lunden Lundúnir Lunnainn Lunnon لندن لندن لندن لوندون לונדון לאנדאןΛονδίνο Лёндан Лондан Лондон Лондон Лондон Լոնդոն 伦敦 …

Matching, Linkage, Reconciliation, etc.

- Many names, descriptions, or IDs (URIs) are used for the same real-world "entity"
- Example:

London 런던 ܠܘܢܕܘܢ लंडन लंदन લંડન ለንደን ロンドン
লন্ডন ລອນດອນ இலண்டன் ლონდონი Llundain
Londain Londe Londen Londen Londen Londinium
London Londona Londonas Londoni Londono Londra
Londres Londrez Londyn Lontoo Loundres Luân Đôn
Lunden Lundúnir Lunnainn Lunnon لندن لندن لندن لوندون
לונדון לאנדאן Λονδίνο Лёндан Лондан Лондон Лондон
Лондон Լոնդոն 伦敦 …

capital of UK, host city of the IV Olympic Games, host city of the XIV Olympic Games, future host of the XXX Olympic Games, city of the Westminster Abbey, city of the London Eye, the city described by Charles Dickens in his novels, …

Matching, Linkage, Reconciliation, etc.

- Many names, descriptions, or IDs (URIs) are used for the same real-world “entity”
- Example:

London 런던 ᓚᓐᑕᓐ लंडन लंदन લંડન ለንደን ロンドン লন্ডন ลอนดอน இலண்டன் ლონდონი Llundain Londain Londe Londen Londen Londen Londinium London Londona Londonas Londoni Londono Londra Londres Londrez Londyn Lontoo Loundres Luân Đôn Lunden Lundúnir Lunnainn Lunnon لندن لندن لندن لوندون לאנדאן לונדון Λονδίνο Лёндан Лондан Лондон Лондон Лондон Լոնդոն 伦敦 …

capital of UK, host city of the IV Olympic Games, host city of the XIV Olympic Games, future host of the XXX Olympic Games, city of the Westminster Abbey, city of the London Eye, the city described by Charles Dickens in his novels, …

http://sws.geonames.org/2643743/
http://en.wikipedia.org/wiki/London
http://dbpedia.org/resource/Category:London
…

Disambiguation, Deduplication, etc.

- Plethora of different “entities” have the same name
- Example:
  - London, KY
  - London, Laurel, KY
  - London, OH
  - London, Madison, OH
  - London, AR
  - London, Pope, AR
  - London, TX
  - London, Kimble, TX
  - London, MO
  - London, London, MI
  - London, London, Monroe, MI
  - London, Uninc Conecuh County, AL
  - London, Uninc Conecuh County, Conecuh, AL
  - London, Uninc Shelby County, IN
  - London, Uninc Shelby County, Shelby, IN
  - London, Deerfield, WI
  - London, Deerfield, Dane, WI
  - London, Uninc Freeborn County, MN
  - ...

Disambiguation, Deduplication, etc.

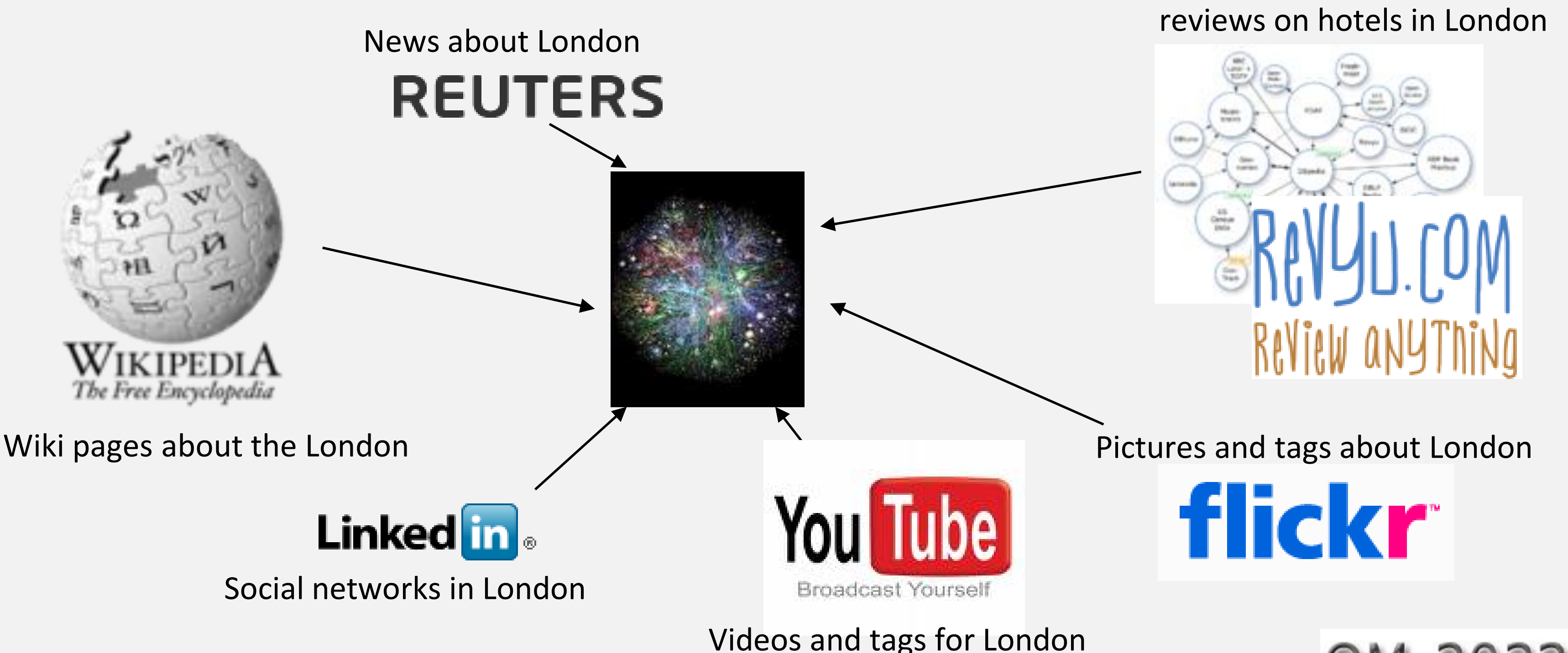
- Plethora of different "entities" have the same name
- Example:
  - London, KY
  - London, Laurel, KY
  - London, OH
  - London, Madison, OH
  - London, AR
  - London, Pope, AR
  - London, TX
  - London, Kimble, TX
  - London, MO
  - London, Lo
  - London, Lo
  - London, Ur
  - London, Ur
  - London, Ur
  - London, Ur
  - London, De
  - London, De
  - London, Ur
  - ...

AL

- London, Jack
  2612 Almes Dr
  Montgomery, AL
  (334) 272-7005
- London, Jack R
  2511 Winchester Rd
  Montgomery, AL 36106-3327
  (334) 272-7005
- London, Jack
  1222 Whitetail Trl
  Van Buren, AR 72956-7368
  (479) 474-4136
- London, Jack
  7400 Vista Del Mar Ave
  La Jolla, CA 92037-4954
  (858) 456-1850
- ...

Entities in today's settings

- Content providers provide valuable information describing (part of) real-world "entities"
- ER is required for data integration, link discovery, query answering, Web / object-oriented searching, etc.



Entity Resolution

- Identifies and aggregates the different entity profiles that describe the same objects [1,2,3,4]
- Primary usefulness:
  - Improves data quality and integrity
  - Fosters re-use of existing data sources
- Example application domains:
  - Linked Data
  - Building Knowledge Graphs
  - Census data
  - Price comparison portals

Types of Entity Resolution

- The given entity collections can be of two types:

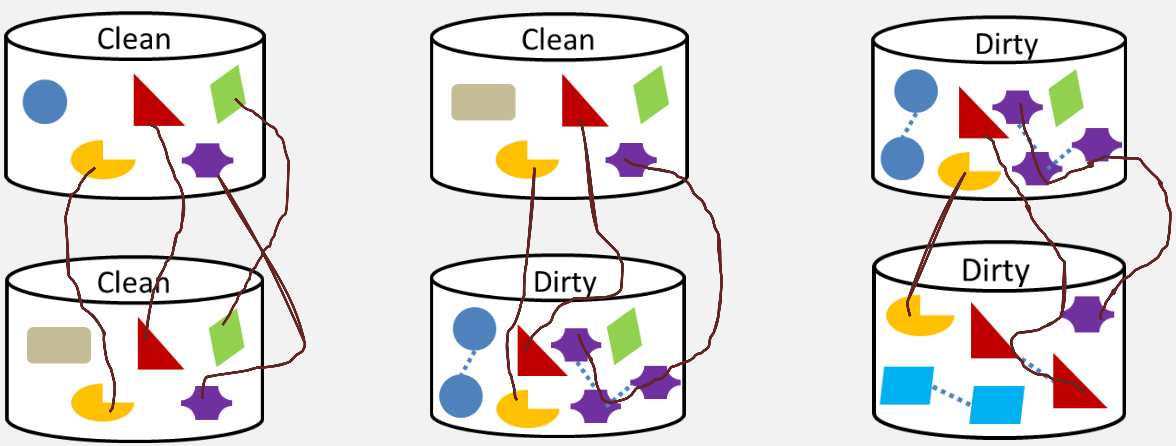
  clean + dirty [3,5]

- Clean:
  - Duplicate-free data
  - E.g., DBLP, ACM Digital Library, Wikipedia, Freebase
- Dirty:
  - Contain duplicate entity profiles
  - E.g., Google Scholar, CiteseerX

Types of Entity Resolution

- Based on the quality of input, we distinguish entity resolution into 3 sub-tasks:
  1. Clean-Clean ER (a.k.a. ***Record Linkage*** in databases)
  2. Dirty-Clean ER
  3. Dirty-Dirty ER

  Equivalent to **Dirty ER** (a.k.a. ***Deduplication*** in databases)



References

1. X. L. Dong, D. Srivastava. Big Data Integration. Synthesis Lectures on Data Management, Morgan & Claypool Publishers 2015, pp. 1-198.
2. A. K. Elmagarmid, P. G. Ipeirotis, V. S. Verykios. Duplicate Record Detection: A Survey. IEEE Trans. Knowl. Data Eng. 19(1): 1-16 (2007).
3. V. Christophides, V. Efthymiou, K. Stefanidis. Entity Resolution in the Web of Data. Synthesis Lectures on the Semantic Web: Theory and Technology, Morgan & Claypool Publishers 2015.
4. P. Christen. Data Matching - Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection. Data-Centric Systems and Applications, Springer 2012, ISBN 978-3-642-31163-5, pp. I-XIX, 1-270.
5. P. Christen. A Survey of Indexing Techniques for Scalable Record Linkage and Deduplication. IEEE Trans. Knowl. Data Eng. 24(9): 1537-1555 (2012).

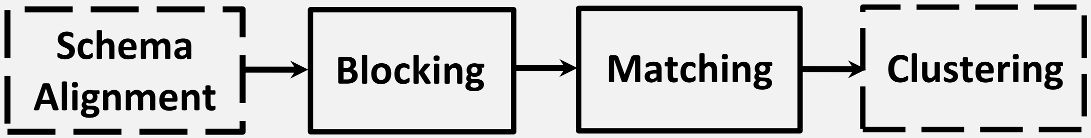
- Introduction

Part B – Four Generations

  - Generation 1: tackling Veracity
  - Generation 2: tackling Volume and Veracity
  - Generation 3: tackling Variety, Volume and Veracity
  - Generation 4: tackling Velocity, Variety, Volume and Veracity

- The Fifth Generation: Leveraging External Knowledge
- Challenges and Final Remarks

Generation 1: Tackling Veracity



- Earliest approach
- Scope:
  - Structured data
- Goal:
  - Achieve high accuracy despite inconsistencies, noise, or errors in entity profiles
- Assumptions:
  - Known schema → custom, schema-based solutions

Step 1: Schema Alignment / Matching

- Scope:
  - Record Linkage
- Goal:
  - Create mappings between equivalent attributes of the two schemata, e.g., *profession ≡ job*
- Types of Solutions:
  - Structure-based
  - Instance-based
  - Hybrid

Step 1: Schema Alignment / Matching

- Taxonomy of Main Schema Matching Methods (in chronological order)

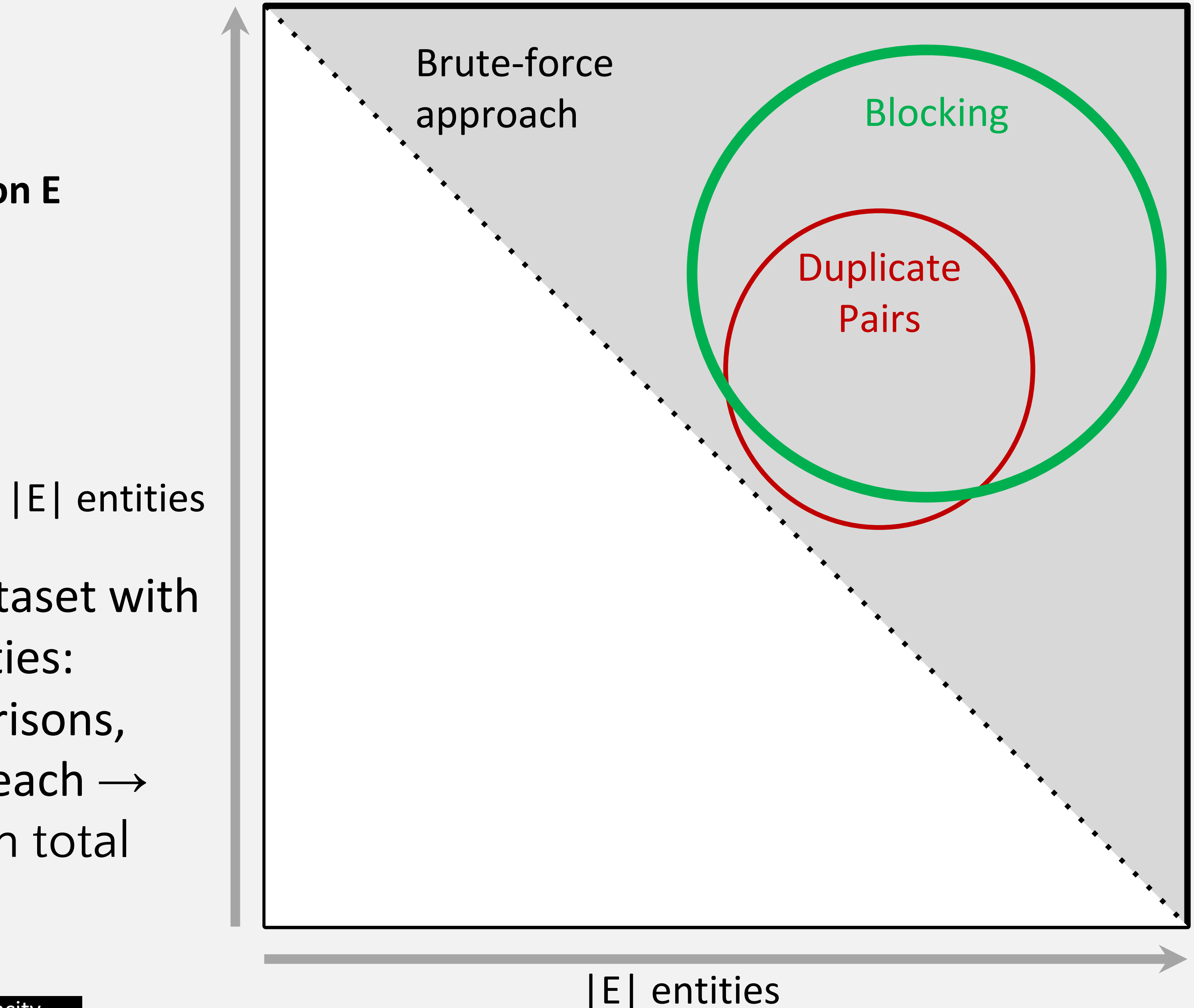
Method	Category	Type of Evidence
Cupid [1]	Structure-based	Name similarity, Constraints, Contextual similarity
Similarity Flooding [2]	Structure-based	Name similarity, Contextual similarity
COMA [3]	Hybrid	Name similarity, Constraints, Contextual similarity
Distribution-based [4]	Instance-based	Value distribution

Step 2: Blocking

- Scope:
  - Both Deduplication and Record Linkage
- Goal:
  - ER is an inherently quadratic problem, $O(n^2)$: every entity has to be compared with all others
  - Blocking groups similar entities into blocks
    - Comparisons are executed only inside each block
    - Complexity is now quadratic to the size of the block (much smaller than dataset size!)

Computational cost

Input:
Entity Collection E



E.g.: For a dataset with 100,000 entities:
~10^{10} comparisons,
If 0.05 msec each ⟶
>100 hours in total

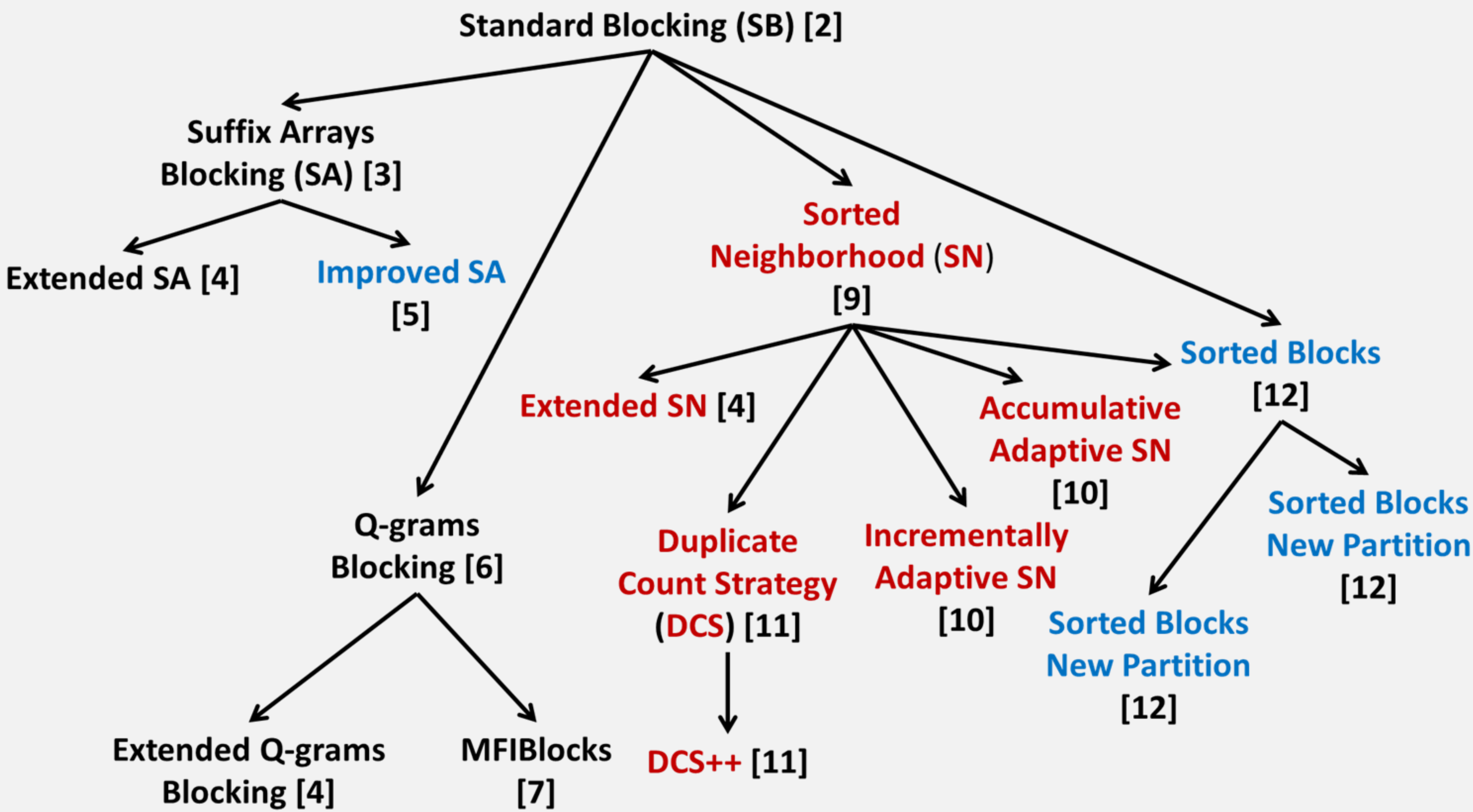
General Principles of Blocking

1. Represent each entity by *one or more* signatures called blocking keys
   – Focus on string values
2. Place into blocks all entities having the *same or similar* blocking key
3. Two matching profiles can be detected as long as they co-occur in at least one block
   – Trade-off between recall and precision!

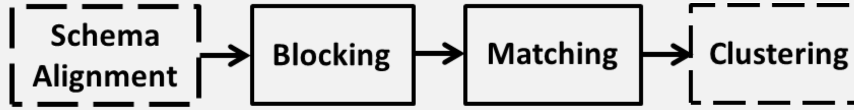
Taxonomy of Blocking Methods [1]

Method	Key Type	Redundancy awareness	Matching awareness	Key selection
Standard Blocking [2]	Hash-based	Red.-free	Static	Non-learning
Suffix Arrays [3] + [4,5]	Hash-based	Red.-positive	Static	Non-learning
Q-grams Blocking [6] + [4]	Hash-based	Red.-positive	Static	Non-learning
MFIBlocks [7]	Hash-based	Red.-positive	Static	Non-learning
Sorted Neighborhood [9] + [4,10]	Sort-based	Red.-neutral	Static	Non-learning
Duplicate Count Strategy [11]	Sort-based	Red.-neutral	Dynamic	Non-learning
Sorted Blocks [12]	Hybrid	Red.-neutral	Static	Non-learning
ApproxDNF [13]	Hash-based	Red.-positive	Static	Learning-based
Blocking Scheme Learner [14]	Hash-based	Red.-positive	Static	Learning-based
CBlock [15]	Hash-based	Red.-positive	Static	Learning-based
FisherDisjunctive [16]	Hash-based	Red.-positive	Static	Learning-based

Genealogy Tree of Non-learning Blocking Methods [1]

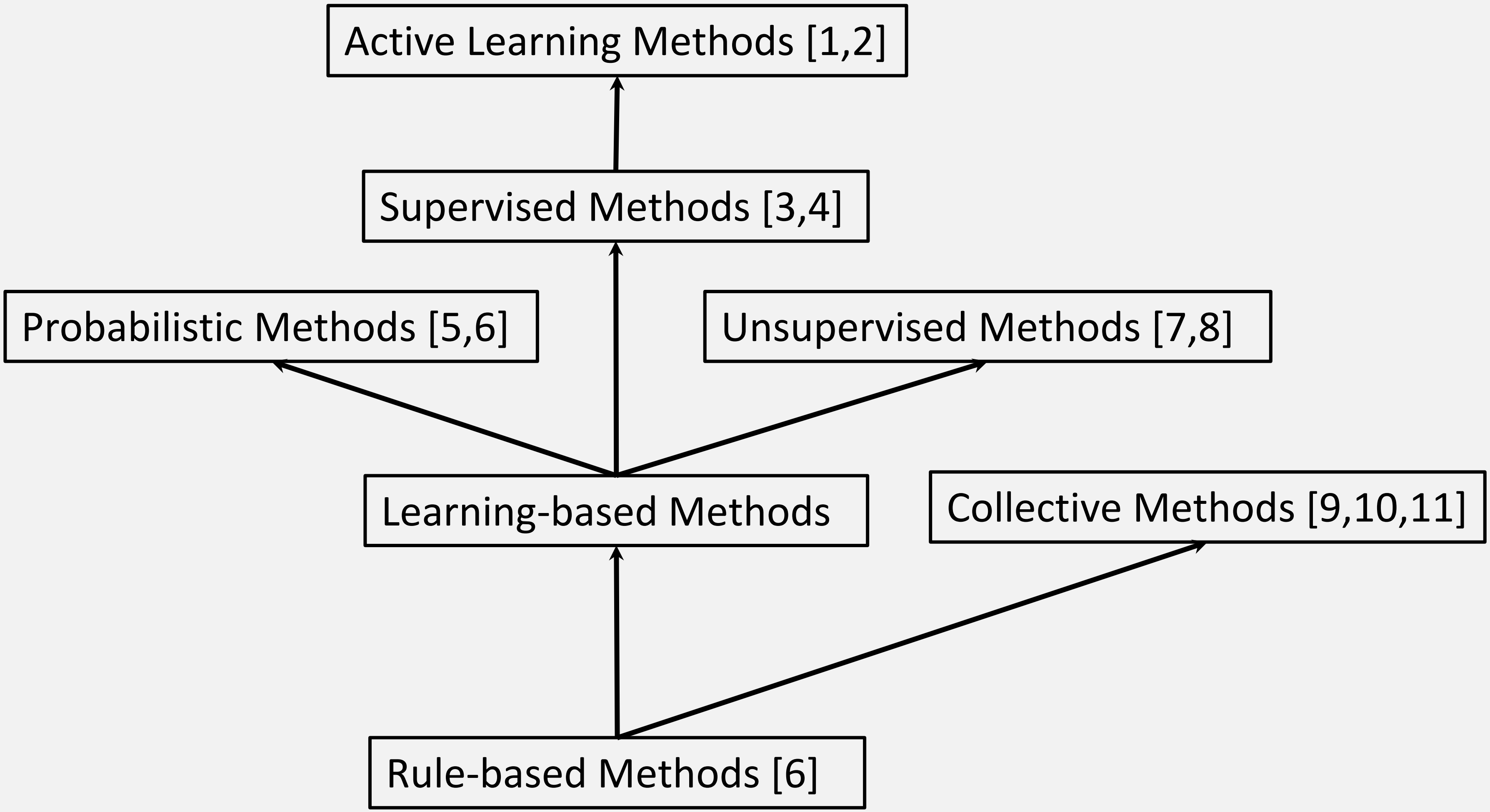


Step 3: Matching



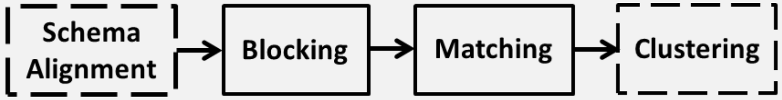
- Estimates the similarity of candidate matches.
- Input
  - A set of blocks
    - Every distinct comparison in any block is a candidate match
- Output
  - Similarity Graph
    - Nodes → entities
    - Edges → candidate matches
    - Edge weights → matching likelihood (based on similarity score)

Evolution of Matching

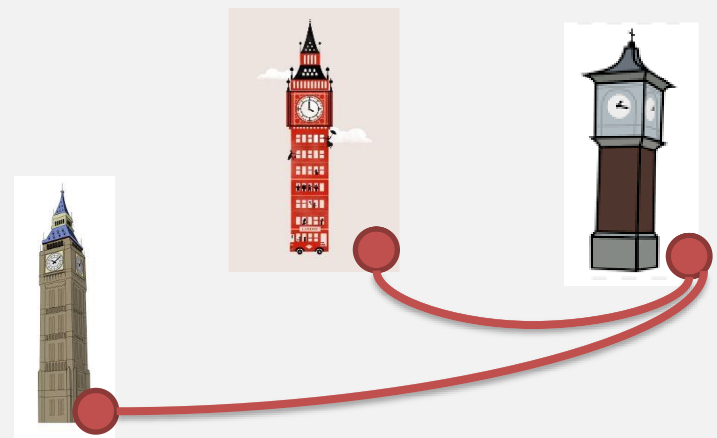


All are heavily based on string similarity measures [6].

Step 4: Clustering



- Partitions the matched pairs into equivalence clusters i.e., groups of entity profiles describing the same real-world object



- Input
  - Similarity Graph:
    - Nodes → entities
    - Edges → candidate matches
    - Edge weights → matching likelihood (based on similarity score)

- Output
  - Equivalence Clusters

Clustering Algorithms

- For Clean-Clean ER [1][2][4]:
  - They rely on the 1-1 constraint:
    - every entity from the source dataset matches with at most one entity from the target dataset
- For Dirty ER [3]:
  - "Unconstrained algorithms": able to predict the correct number of clusters
  - Goal: Sets of clusters that
    - maximize the intra-cluster weights
    - minimize the inter-cluster edge weights
- For both tasks:
  - Need to scale well
    - Time complexity < $O(n^2)$
  - Need to be robust with respect to characteristics of the data
    - E.g., distribution of the duplicates

Schema Matching References

1. J. Madhavan, P. A. Bernstein, and E. Rahm. Generic schema matching with cupid. In VLDB, pages 49–58, 2001.
2. S. Melnik, H. Garcia-Molina, and E. Rahm. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In ICDE, pages 117–128, 2002.
3. H.-H. Do and E. Rahm. COMA: a system for flexible combination of schema matching approaches. In VLDB, pages 610–621, 2002.
4. M. Zhang, M. Hadjieleftheriou, B. C. Ooi, C. M. Procopiuc, D. Srivastava. Automatic discovery of attributes in relational databases. In SIGMOD, pages 109–120, 2011.
5. H. W. Kuhn. The hungarian method for the assignment problem. Naval research logistics quarterly, 2(1-2):83–97, 1955.
6. L. Ramshaw and R. E. Tarjan. On minimum-cost assignments in unbalanced bipartite graphs. HP Labs, Palo Alto, CA, USA, Tech. Rep. HPL-2012-40R1, 2012.

Blocking References – Part I

1. George Papadakis, Dimitrios Skoutas, Emmanouil Thanos, Themis Palpanas: A Survey of Blocking and Filtering Techniques for Entity Resolution. CoRR abs/1905.06167 (2019)
2. I. P. Fellegi and A. B. Sunter. A theory for record linkage. Journal of the American Statistical Association, 64(328):1183–1210, 1969.
3. A. N. Aizawa and K. Oyama. A fast linkage detection scheme for multi-source information integration. In WIRI, pages 30–39, 2005.
4. P. Christen. A survey of indexing techniques for scalable record linkage and deduplication. IEEE TKDE, 24(9):1537–1555, 2012.
5. T. de Vries, H. Ke, S. Chawla, and P. Christen. Robust record linkage blocking using suffix arrays. In CIKM, pages 305–314, 2009
6. R. Baxter, P. Christen, and T. Churches. A comparison of fast blocking methods for record linkage. In KDD Workshops, 2003.
7. B. Kenig and A. Gal. Mfiblocks: An effective blocking algorithm for entity resolution. Inf. Syst., 38(6):908–926, 2013.
8. M. A. Hernández and S. J. Stolfo. The merge/purge problem for large databases. In SIGMOD, pages 127–138, 1995.
9. S. Yan, D. Lee, M. Kan, and C. L. Giles. Adaptive sorted neighborhood methods for efficient record linkage. In JCDL, pages 185–194, 2007.

Blocking References – Part II

11. U. Draisbach, F. Naumann, S. Szott, and O. Wonneberg. Adaptive windows for duplicate detection. In ICDE, pages 1073–1083, 2012.
12. U. Draisbach and F. Naumann. A generalization of blocking and windowing algorithms for duplicate detection. In ICDKE, pages 18–24, 2011
13. M. Bilenko, B. Kamath, and R. J. Mooney. Adaptive blocking: Learning to scale up record linkage. In ICDM, pages 87–96, 2006
14. M. Michelson and C. A. Knoblock. Learning blocking schemes for record linkage. In AAAI, pages 440–445, 2006
15. A. D. Sarma, A. Jain, A. Machanavajjhala, and P. Bohannon. An automatic blocking mechanism for large-scale de-duplication tasks. In CIKM, pages 1055–1064, 2012.
16. M. Kejriwal and D. P. Miranker. An unsupervised algorithm for learning blocking schemes. In ICDM, pages 340–349, 2013.

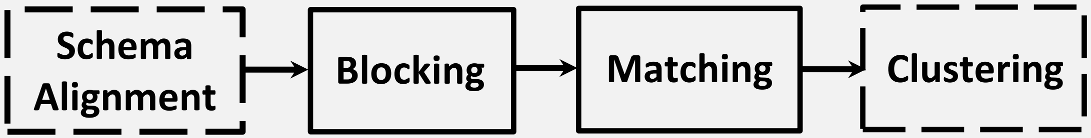
Matching References

1. K. Qian, L. Popa, P. Sen. Active Learning for Large-Scale Entity Resolution. CIKM 2017: 1379-1388
2. .J. Fisher, P. Christen, Q. Wangl. Active Learning Based Entity Resolution Using Markov Logic. PAKDD (2) 2016: 338-349
3. Reyes-Galaviz, O.F., Pedrycz, W., He, Z., Pizzi, N.J. A supervised gradient-based learning algorithm for optimized entity resolution. Data Knowl. Eng. 112, 106–129 (2017)
4. P. Christen. Automatic record linkage using seeded nearest neighbour and support vector machine classification. KDD 2008: 151-159.
5. A. Rasch, R. Schulze, W. Gorus, J. Hiller, S. Bartholomäus, S. Gorlatch. High-performance probabilistic record linkage via multi-dimensional homomorphisms. SAC 2019: 526-533.
6. A. K. Elmagarmid, P. G. Ipeirotis, V. S. Verykios. Duplicate Record Detection: A Survey. IEEE Trans. Knowl. Data Eng. 19(1): 1-16 (2007)
7. A. Jurek, J. Hong, Y. Chi, W. Liu. A novel ensemble learning approach to unsupervised record linkage. Inf. Syst. 71: 40-54 (2017)
8. A. Jurek, Deepak P. It Pays to Be Certain: Unsupervised Record Linkage via Ambiguity Minimization. PAKDD (3) 2018: 177-190.X
9. . Dong, A. Y. Halevy, J. Madhavan. Reference Reconciliation in Complex Information Spaces. SIGMOD Conference 2005: 85-96.O
10. . Benjelloun, H. Garcia-Molina, D. Menestrina, Q. Su, S. E. Whang, J. Widom. Swoosh: a generic approach to entity resolution. VLDB J. 18(1): 255-276 (2009).
11. I. Bhattacharya, L. Getoor. Collective entity resolution in relational data. TKDD 1(1): 5 (2007).

Clustering References

1. S. Lacoste-Julien, K. Palla, A. Davies, G. Kasneci, T. Graepel, Z. Ghahramani. SIGMa: simple greedy matching for aligning large knowledge bases. KDD 2013: 572-580
2. F. M. Suchanek, S. Abiteboul, P. Senellart. PARIS: Probabilistic Alignment of Relations, Instances, and Schema. PVLDB 5(3): 157-168 (2011)
3. O. Hassanzadeh, F. Chiang, R. J. Miller, H. C. Lee. Framework for Evaluating Clustering Algorithms in Duplicate Detection. PVLDB 2(1): 1282-1293 (2009)
4. H. W. Kuhn. The hungarian method for the assignment problem. Naval research logistics quarterly, 2(1-2):83–97, 1955.

Generation 2: Tackling **Volume** and Veracity



- Same workflow as Generation 1
- Scope:
  - (tens of) millions of structured entity profiles
- Goals:
  - High accuracy despite noise
  - High time efficiency despite the size of data
- Assumptions:
  - Known schema → custom, schema-based solutions

Solution: Parallelization

Two types:

- Multi-core parallelization
  - Single system → shared memory
  - Distribute processing among available CPUs
- Massive parallelization
  - Cluster of independent systems
  - **Map-Reduce** paradigm [1]
    - Data partitioned across the nodes of a cluster
    - Fault-tolerant, optimized execution
    - Map Phase: transforms a data partition into (key, value) pairs
    - Reduce Phase: processes pairs with the same key

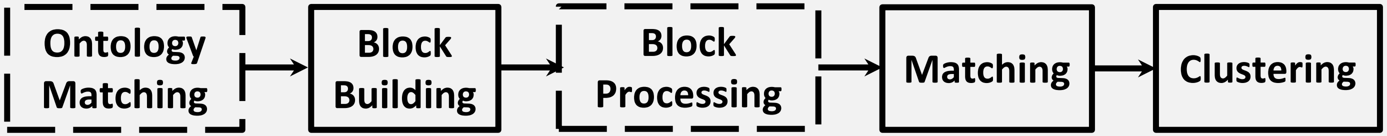
Parallelization Methods per Step

- Blocking
    - Dedoop [2]
    - MapReduce-based Sorted Neighborhood [3]
- Matching
  – Multi-core approaches [7][8]
  – MapReduce-based: Emphasis on **load balancing**
    - BlockSplit & PairRange [4][5]
    - Dis-Dedup [6]
    - Message-passing framework [9]
- Clustering
    - Fast Multi-source Entity Resolution (FAMER) framework [10][11]

Generation 2 References

1. J. Dean and S. Ghemawat. Mapreduce: simplified data processing on large clusters. Commun. ACM, 51(1):107–113, 2008.
2. L. Kolb, A. Thor, and E. Rahm. Dedoop: Efficient deduplication with hadoop. PVLDB, 5(12):1878–1881, 2012.
3. L. Kolb, A. Thor, and E. Rahm. Multi-pass sorted neighborhood blocking with mapreduce. Computer Science - R&D, 27(1):45–63, 2012.
4. L. Kolb, A. Thor, and E. Rahm. Load balancing for mapreduce-based entity resolution. In ICDE, pages 618–629, 2012.
5. W. Yan, Y. Xue, and B. Malin. Scalable load balancing for mapreduce-based record linkage. In IPCCC, pages 1–10, 2013.
6. X. Chu, I. F. Ilyas, and P. Koutris. Distributed data deduplication. PVLDB, 9(11):864–875, 2016.
7. O. Benjelloun, H. Garcia-Molina, H. Gong, H. Kawai, T. E. Larson, D. Menestrina, and S. Thavisomboon. D-swoosh: A family of algorithms for generic, distributed entity resolution. In ICDCS, page 37, 2007.
8. Hung-sik Kim and Dongwon Lee. Parallel linkage. In CIKM, pages 283–292, 2007.
9. V. Rastogi, N. N. Dalvi, and M. N. Garofalakis. Large-scale collective entity matching. PVLDB, 4(4):208–218, 2011.
10. A. Saeedi, E. Peukert, and E. Rahm. Comparative evaluation of distributed clustering schemes for multi-source entity resolution. In ADBIS, pages 278–293, 2017.
11. A. Saeedi, M. Nentwig, E. Peukert, and E. Rahm. Scalable matching and clustering of entities with FAMER. CSIMQ, 16:61–83, 2018.

G3: Tackling **Variety**, Volume and Veracity



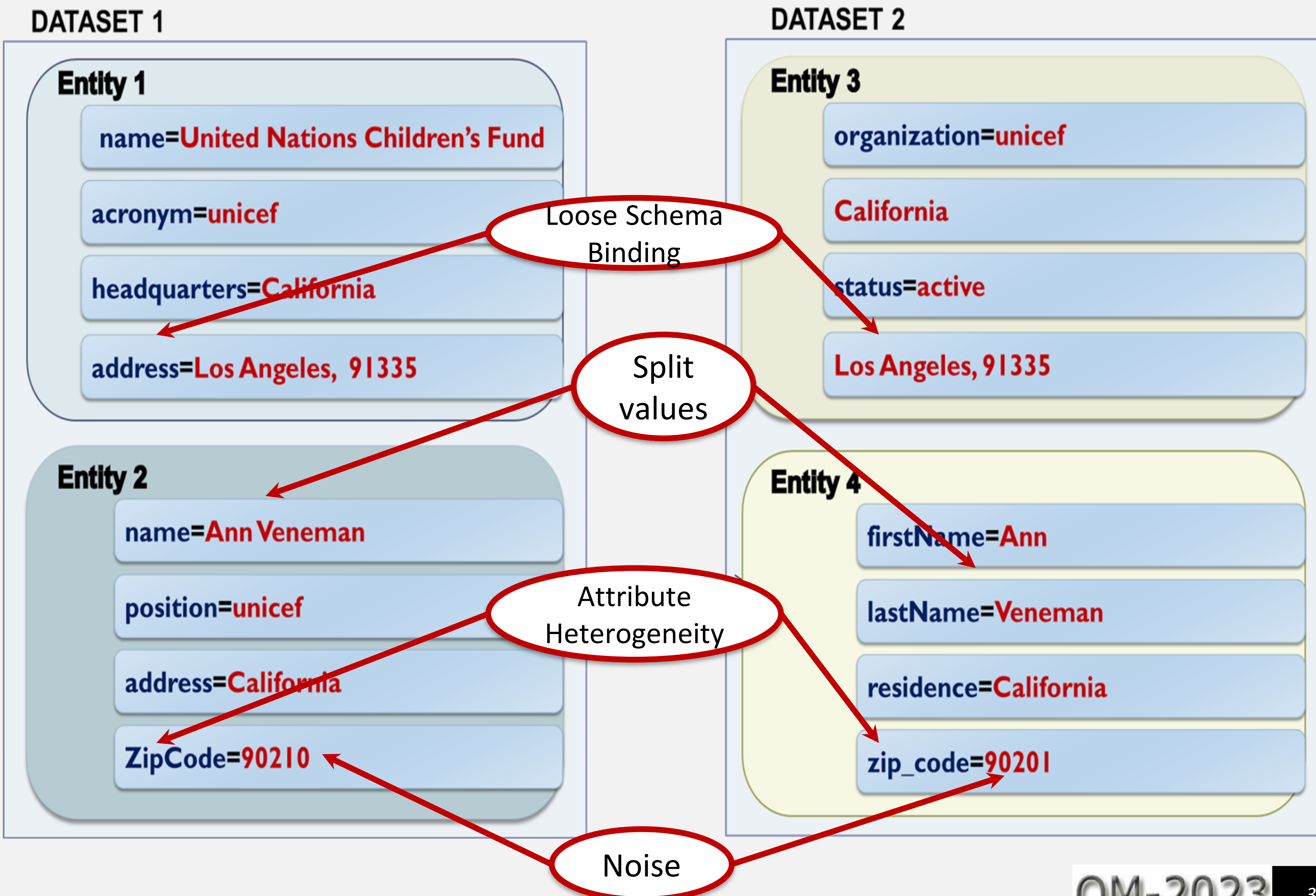
- Scope:
  - User-generated Web Data

    Voluminous, (semi-)structured datasets.
    - BTC09: 1.15 billion triples, 182 million entities.

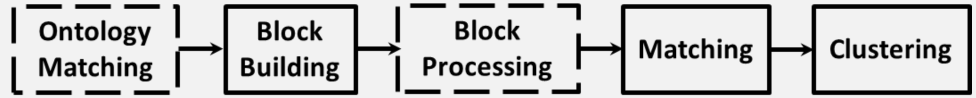
    Users are free to add attribute values and/or attribute names
    → unprecedented levels of schema heterogeneity.
    - Google Base: 100,000 schemata for 10,000 entity types
    - BTC09: 136,000 predicates

    Several datasets produced by automatic information extraction techniques → noise, tag-style values.

Example of Web Data



Ontology Matching



- Schema Matching → not scalable
  → not effective (more complex tasks)

- For details, see the series of the “International Workshop on Ontology Matching”

Block Building

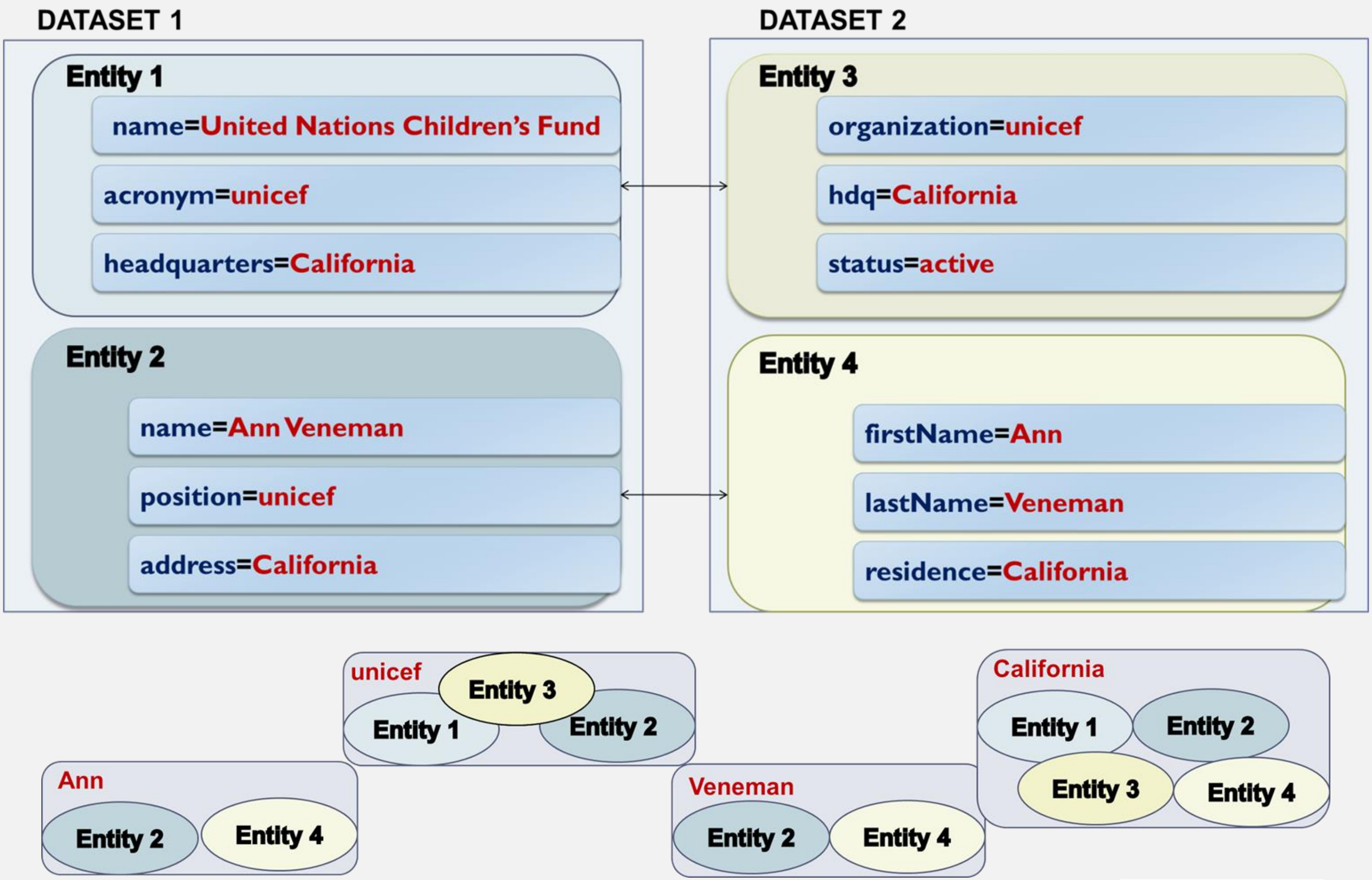
Ontology Matching → Block Building → Block Processing → Matching → Clustering

- Unlike Blocking in G1/G2, it considers all attribute values and completely ignores all attribute names
  → **schema-agnostic functionality**

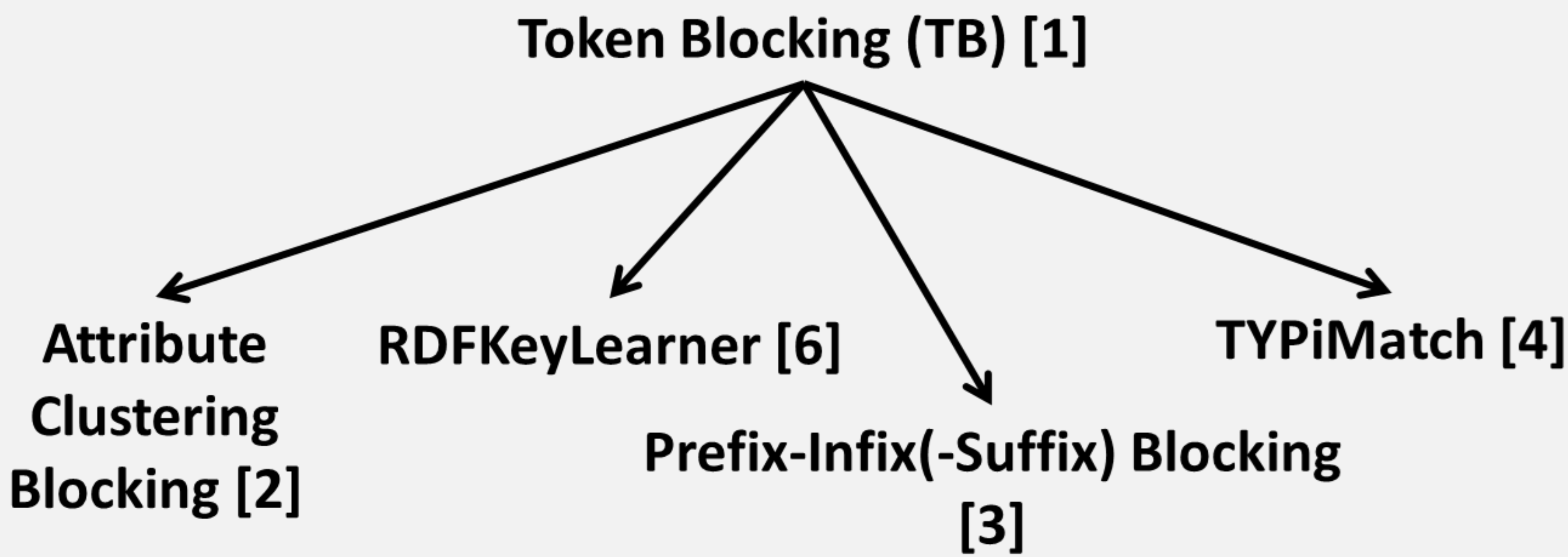
- Core approach: Token Blocking [1]
  1. Given an entity profile, extract all tokens that are contained in its attribute values.
  2. Create one block for every distinct token with frequency > 2 → each block contains all entities with the corresponding token.

     Pros:
     - Parameter-free
     - Efficient
     - Unsupervised

Example of Token Blocking



Genealogy of Block Building Techniques [8]



Semantic Graph Blocking [5]

MapReduce-based parallelizations in [7]

Block Processing

- High Recall due to redundancy
- Low Precision due to:
  1. the blocks are overlapping → **redundant comparisons**
  2. high number of comparisons between irrelevant entities → **superfluous comparisons**

Solution:

restructure the original blocks so as to increase precision at no significant cost in recall

Block Processing Techniques

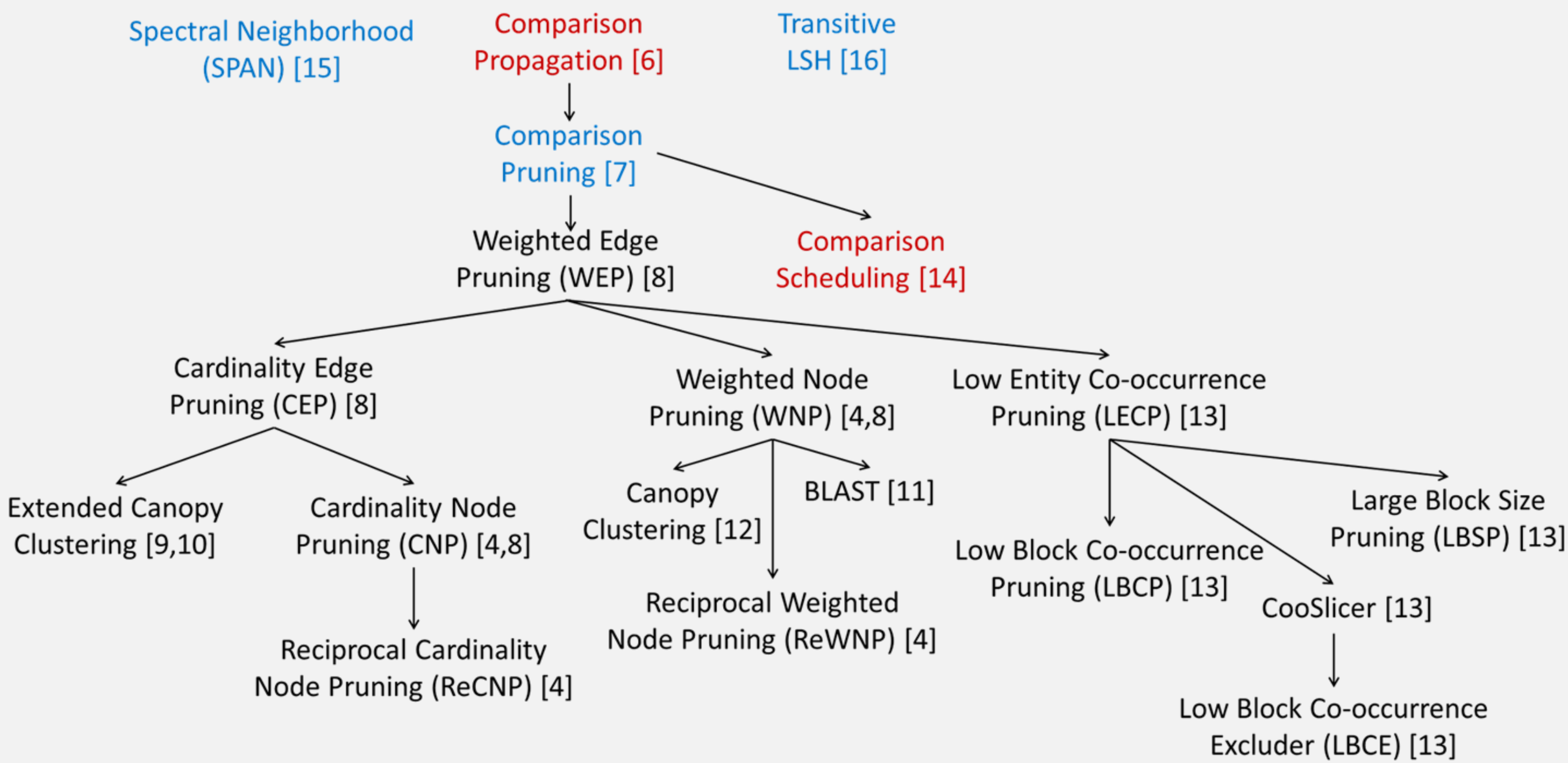
Generic approach

- Assign a matching likelihood score to each item
- Discard items with low costs

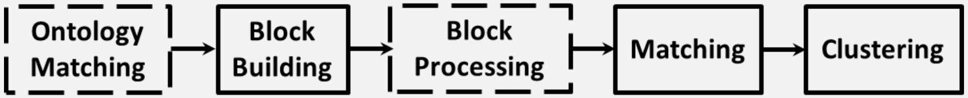
Block-centric methods

- Block Purging [1,2,3]
- Block Filtering [4]
- Block Clustering [5]

Comparison Cleaning Methods [17]

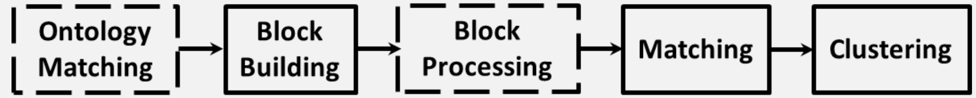


Entity Matching



- Collective approaches to tackle Variety
- Most methods are crafted for Clean-Clean ER
- General outline of
  SiGMa [1], PARIS [2], LINDA [3], RiMOM-IM [4,5]
  - Bootstrap with a few reliable seed matches.
  - Using value and neighbor similarity, propagate initial matches to neighbors.
  - Order candidate matches in descending overall similarity
  - Iteratively mark the top pair as a match if it satisfies a constraint
  - Recompute the similarity of the neighbors
  - Update candidate matches order
- MinoanER [6] performs a specific number of steps, rather than iterating until convergence

Entity Clustering



- Methods of G1 & G2 are still applicable
  - Only difference: similarity scores extracted in a schema-agnostic fashion, not from specific predicates

Block Building References

1. G. Papadakis, E. Ioannou, C. Niederée, P. Fankhauser. Efficient entity resolution for large heterogeneous information spaces. WSDM 2011: 535-544
2. G. Papadakis, E. Ioannou, T. Palpanas, C. Niederée, W. Nejdl. A Blocking Framework for Entity Resolution in Highly Heterogeneous Information Spaces. IEEE Trans. Knowl. Data Eng. 25(12): 2665-2682 (2013)
3. G. Papadakis, E. Ioannou, C. Niederée, T. Palpanas, W. Nejdl. Beyond 100 million entities: large-scale blocking-based resolution for heterogeneous data. WSDM 2012: 53-62
4. Y. Ma, T. Tran. TYPiMatch: type-specific unsupervised learning of keys and key values for heterogeneous web data integration. WSDM 2013: 325-334
5. J. Nin, V. Muntés-Mulero, N. Martínez-Bazan, and J. Larriba-Pey. On the use of semantic blocking techniques for data cleansing and integration. In IDEAS, pages 190–198, 2007.
6. D. Song and J. Heflin. Automatically generating data linkages using a domain-independent candidate selection approach. In ISWC, pages 649–664, 2011.
7. V. Christophides, V. Efthymiou, K. Stefanidis. Entity Resolution in the Web of Data. Synthesis Lectures on the Semantic Web: Theory and Technology, Morgan & Claypool Publishers 2015.
8. George Papadakis, Dimitrios Skoutas, Emmanouil Thanos, Themis Palpanas: A Survey of Blocking and Filtering Techniques for Entity Resolution. CoRR abs/1905.06167 (2019)

Block Processing References – Part I

1. G. Papadakis, E. Ioannou, C. Niederée, P. Fankhauser. Efficient entity resolution for large heterogeneous information spaces. WSDM 2011: 535-544
2. G. Papadakis, E. Ioannou, T. Palpanas, C. Niederée, W. Nejdl. A Blocking Framework for Entity Resolution in Highly Heterogeneous Information Spaces. IEEE Trans. Knowl. Data Eng. 25(12): 2665-2682 (2013)
3. G. Papadakis, E. Ioannou, C. Niederée, T. Palpanas, W. Nejdl. Beyond 100 million entities: large-scale blocking-based resolution for heterogeneous data. WSDM 2012: 53-62
4. G. Papadakis, G. Papastefanatos, T. Palpanas, M. Koubarakis. Scaling Entity Resolution to Large, Heterogeneous Data with Enhanced Meta-blocking. EDBT 2016: 221-232
5. J. Fisher, P. Christen, Q. Wang, E. Rahm. A Clustering-Based Framework to Control Block Sizes for Entity Resolution. KDD 2015: 279-288
6. G. Papadakis, E. Ioannou, C. Niederée, T. Palpanas, W. Nejdl. Eliminating the redundancy in blocking-based entity resolution methods. JCDL 2011: 85-94.
7. G. Papadakis, E. Ioannou, C. Niederée, T. Palpanas, W. Nejdl. To compare or not to compare: making entity resolution more efficient. SWIM 2011: 3.
8. G. Papadakis, G. Koutrika, T. Palpanas, W. Nejdl. Meta-Blocking: Taking Entity Resolutionto the Next Level. IEEE Trans. Knowl. Data Eng. 26(8): 1946-1960 (2014).
9. P. Christen. A Survey of Indexing Techniques for Scalable Record Linkage and Deduplication. IEEE Trans. Knowl. Data Eng. 24(9): 1537-1555 (2012).
10. G. Papadakis, G. Alexiou, G. Papastefanatos, G. Koutrika. Schema-agnostic vs Schema-based Configurations for Blocking Methods on Homogeneous Data. PVLDB 9(4): 312-323 (2015).

Block Processing References – Part II

11. G. Simonini, S. Bergamaschi, H. V. Jagadish. BLAST: a Loosely Schema-aware Meta-blocking Approach for Entity Resolution. PVLDB 9(12): 1173-1184 (2016)
12. A. McCallum, K. Nigam, L. H. Ungar. Efficient clustering of high-dimensional data sets with application to reference matching. KDD 2000: 169-178.
13. D. C. Nascimento, C. E. S. Pires, and D. G. Mestre. Exploiting block co-occurrence to control block sizes for entity resolution. Knowledge and Information Systems, pages 1–42, 2019.
14. G. Papadakis, E. Ioannou, T. Palpanas, C. Niederée, and W. Nejdl. A blocking framework for entity resolution in highly heterogeneous information spaces. IEEE TKDE, 25(12):2665–2682, 2013.
15. L. Shu, A. Chen, M. Xiong, and W. Meng. Efficient spectral neighborhood blocking for entity resolution. In ICDE, pages 1067–1078, 2011.
16. R. C. Steorts, S. L. Ventura, M. Sadinle, and S. E. Fienberg. A comparison of blocking methods for record linkage. In Privacy in Statistical Databases, pages 253–268, 2014.
17. George Papadakis, Dimitrios Skoutas, Emmanouil Thanos, Themis Palpanas: A Survey of Blocking and Filtering Techniques for Entity Resolution. CoRR abs/1905.06167 (2019)

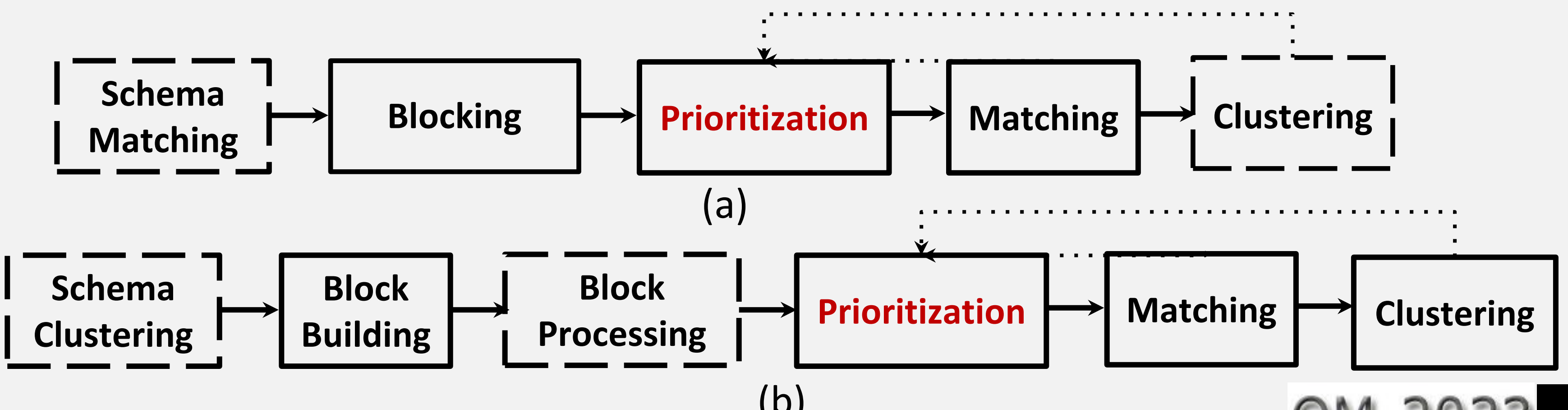
Entity Matching References

1. S. Lacoste-Julien, K. Palla, A. Davies, G. Kasneci, T. Graepel, Z. Ghahramani. SIGMa: simple greedy matching for aligning large knowledge bases. KDD 2013: 572-580
2. F. M. Suchanek, S. Abiteboul, P. Senellart. PARIS: Probabilistic Alignment of Relations, Instances, and Schema. PVLDB 5(3): 157-168 (2011)
3. C. Böhm, G. de Melo, F. Naumann, and G. Weikum. LINDA: distributed web-of-data-scale entity matching. In CIKM, pages 2104–2108, 2012.
4. J. Li, J. Tang, Y. Li, and Q. Luo. Rimom: A dynamic multistrategy ontology alignment framework. TKDE, 21(8):1218–1232, 2009.
5. C. Shao, L. Hu, J. Li, Z. Wang, T. L. Chung, and J.-B. Xia. Rimom-im: A novel iterative framework for instance matching. J. Comput. Sci. Technol., 31(1):185–197, 2016.
6. V. Efthymiou, G. Papadakis, K. Stefanidis, and V. Christophides. MinoanER: Schema-agnostic, non-iterative, massively parallel resolution of web entities. In EDBT, pages 373–384, 2019.

G4: Tackling **Velocity**, Variety, Volume and Veracity

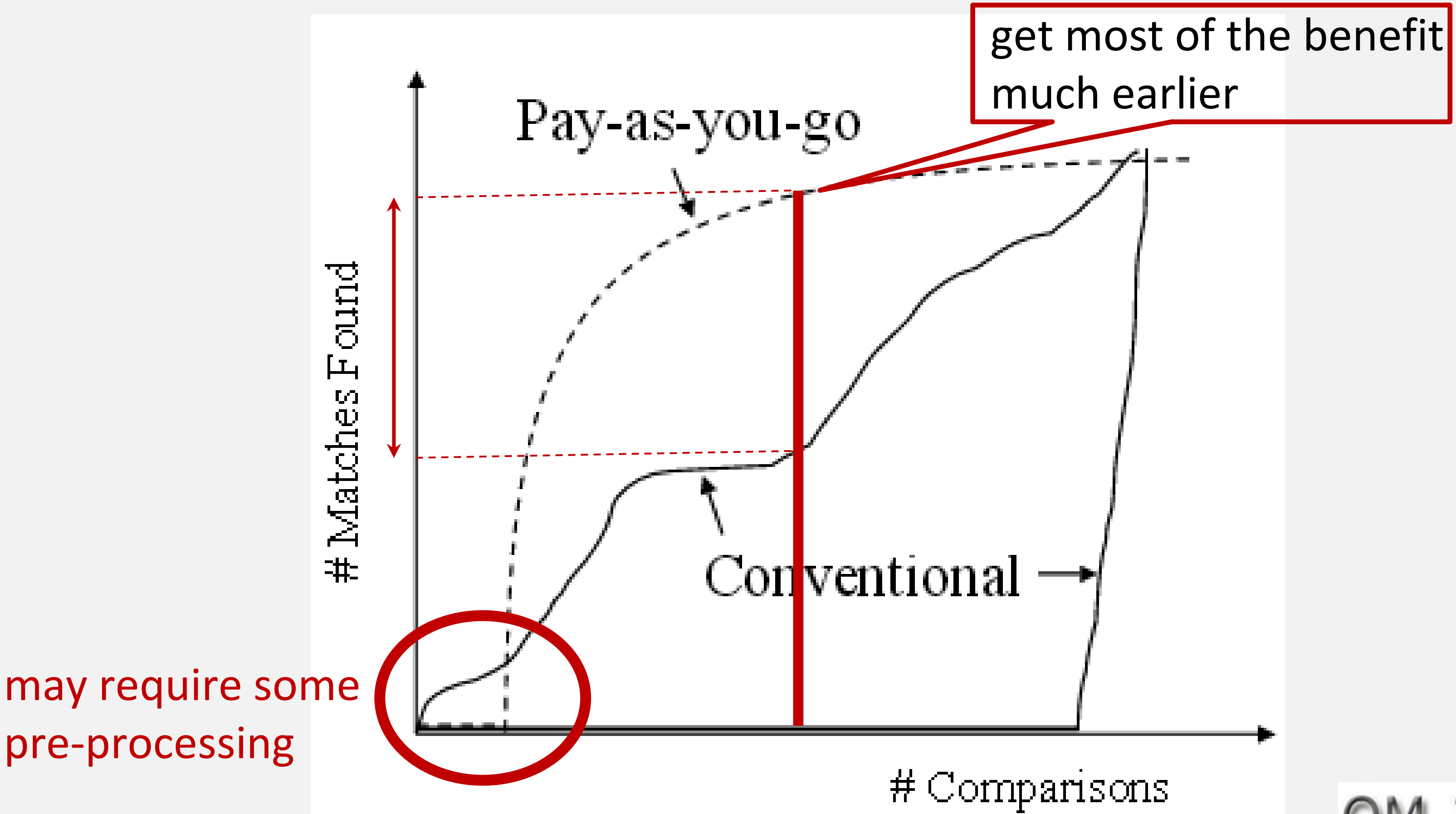
- Scope:
  - Applications with increasing data volume and time constraints
    - Loose ones (e.g., minutes, hours) → Progressive ER
    - Strict ones (i.e., seconds) → Real-time (On-line) ER

- End-to-end workflows for Progressive ER



Progressive Entity Resolution

Unprecedented, increasing volume of data → applications requiring partial solutions to produce useful results

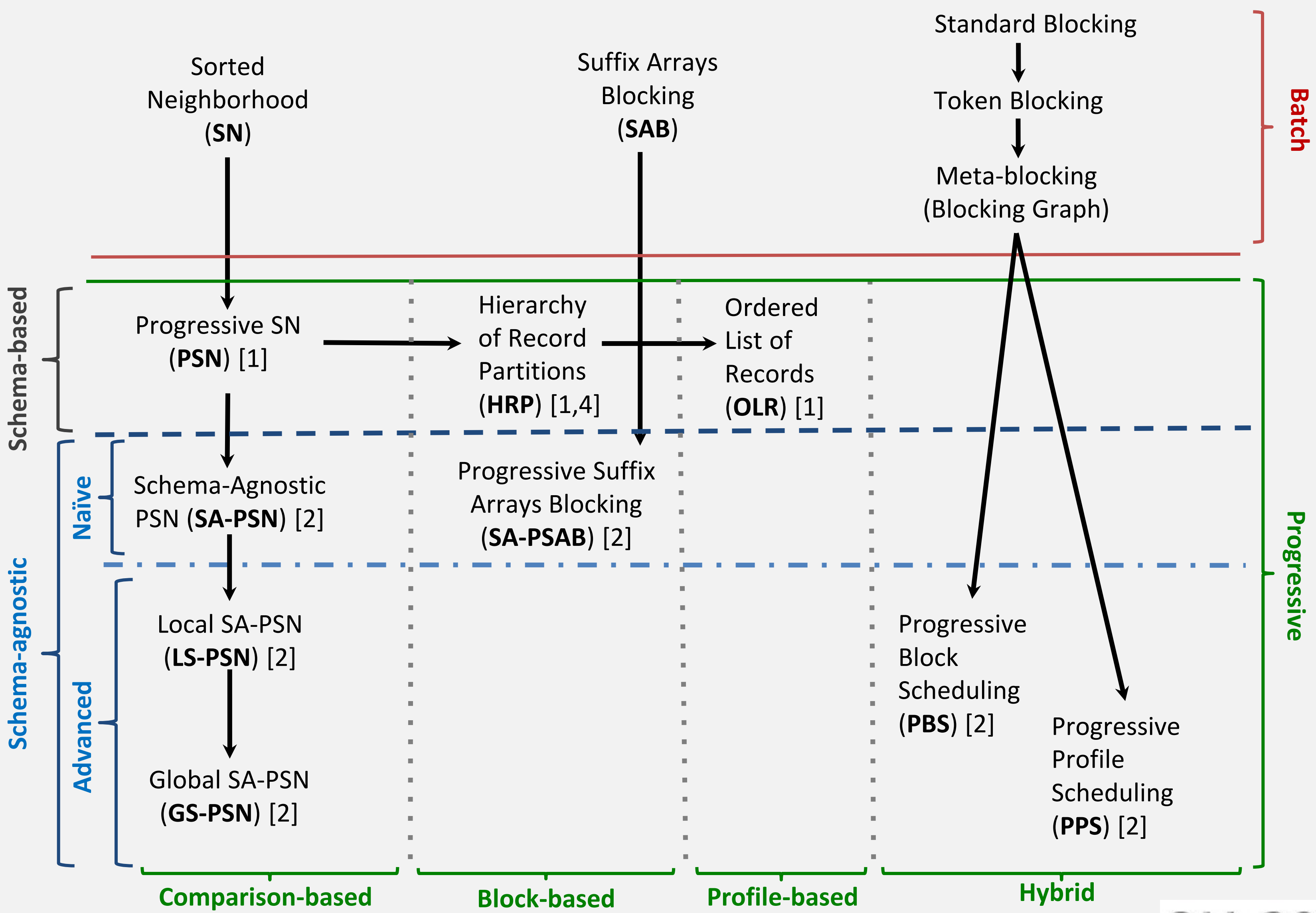


Outline Progressive ER

- Requires:
  - Improved Early Quality
  - Same Eventual Quality

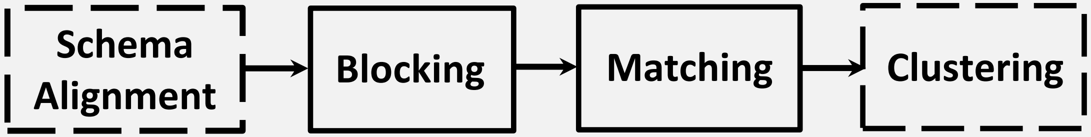
- Prioritization
  - Defines **optimal processing order** for a set of entities
  - Static Methods [1,2]:
    - Guide which records to compare first, **independently** of Entity Matching results
  - Dynamic Methods [3]:
    - If $c_{i,j}$ is a duplicate, then check $c_{i+1,j}$ and $c_{i,j+1}$ as well.
    - Assumption:
      - Oracle for Entity Matching

Taxonomy of Static Prioritization Methods



Real-time Entity Resolution

Same workflow as Generations 1 and 2:



Same scope (so far):

- Structured data

Different input:

- stream of query entity profiles

Different goal:

- resolve each query over a large dataset in the shorted possible time (& with the minimum memory footprint)

Techniques per workflow step

Incremental Blocking

- **DySimII** [1] - extends Standard Blocking
- **F-DySNI** [2,3] - extends Sorted Neighborhood
- **(S)BlockSketch** [4] - bounded matching time, constant memory footprint

Incremental Matching

- **QDA** [5] - SQL-like selection queries over a single dataset
- **QuERy** [6] - complex join queries over multiple, overlapping, dirty DSs
- **EAQP** [7] - queries under data
- Evolving matching rules [8]

Incremental Clustering

- Incremental Correlation Clustering [9]

Progressive ER References

1. S. E. Whang, D. Marmaros, and H. Garcia-Molina. Pay-as-you-go entity resolution. TKDE, 25(5):1111–1124, 2013.
2. T. Papenbrock, A. Heise, and F. Naumann. Progressive duplicate detection. TKDE, 27(5):1316–1329, 2015.
3. G. Simonini, G. Papadakis, T. Palpanas, S. Bergamaschi. Schema-Agnostic Progressive Entity Resolution. IEEE Trans. Knowl. Data Eng. 31(6): 1208-1221 (2019)
4. Y. Altowim and S. Mehrotra. Parallel progressive approach to entity resolution using mapreduce. In ICDE, pages 909–920, 2017.

Incremental ER References

1. B. Ramadan and P. Christen, H. Liang, and R. W. Gayler, and D. Hawking. Dynamic similarity-aware inverted indexing for real-time entity resolution. In PAKDD Workshops, pages 47–58, 2013.
2. B. Ramadan and P. Christen. Forest-based dynamic sorted neighborhood indexing for real-time entity resolution. In CIKM, pages 1787–1790, 2014.
3. B. Ramadan and P. Christen, H. Liang, and R. W. Gayler. Dynamic sorted neighborhood indexing for real-time entity resolution. J. Data and Information Quality, 6(4):15:1–15:29, 2015.
4. D. Karapiperis, A. Gkoulalas-Divanis, V. S. Verykios. Summarization Algorithms for Record Linkage. EDBT 2018: 73-84.
5. H. Altwaijry, D. V. Kalashnikov, and S. Mehrotra. QDA: A query-driven approach to entity resolution. TKDE, 29(2):402–417, 2017.
6. H. Altwaijry, S. Mehrotra, and D. V. Kalashnikov. Query: A framework for integrating entity resolution with query processing. PVLDB, 9(3):120–131, 2015.
7. E. Ioannou, W. Nejdl, C. Niederée, and Y. Velegrakis. On-the-fly entity-aware query processing in the presence of linkage. PVLDB, 3(1): 429–438, 2010.
8. S. E. Whang and H. Garcia-Molina. Entity resolution with evolving rules. PVLDB, 3(1):1326–1337, 2010.
9. A. Gruenheid, X. L. Dong, and D. Srivastava. Incremental record linkage. Proc. VLDB Endow., 7(9):697–708, May 2014. ISSN 2150-8097.

- Introduction
- The First Four Generations

Part C – The Fifth Generation: Leveraging External Knowledge

  - Types of External Knowledge
  - Blocking
  - Matching
  - Clustering
- Challenges and Final Remarks

Crowd-sourcing for Entity Resolution

- Key Idea:
  tasks complex for computers, but simple for human intelligence are divided among many people, called **workers** (e.g., from Amazon Mechanical Turk)

- Adaptation to ER:
  Delegate the **entity matching decisions** to the workers, i.e., transform pairwise comparisons into Human Intelligence Tasks (**HITS**)

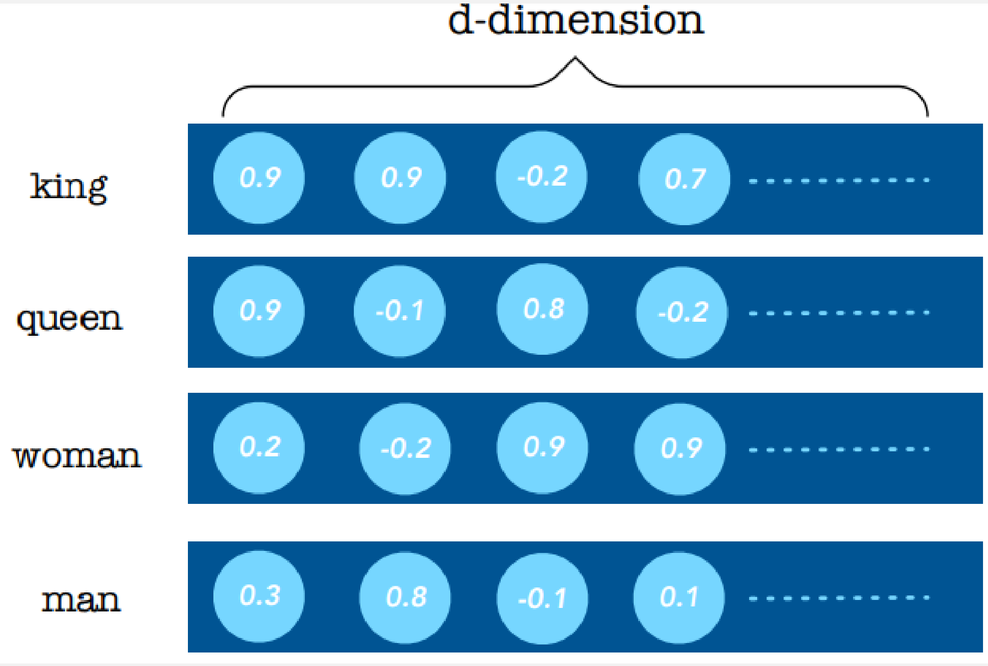
- Challenges:
  1. Generating HITs
  2. Formulating HITs
  3. Balancing accuracy and monetary cost
  4. Restricting the labor cost

Language Models

- Based on the distributional hypothesis

  i.e., *words appearing in the same context share meaning*
- Each word is represented as a distribution of weights (positive or negative) across specific dimensions

- Goal: capture semantic string similarities based on the contextual information from huge textual corpora
- Note: it applies to both **blocking** and **matching**, unlike crowd-sourcing



Performance of Language Models

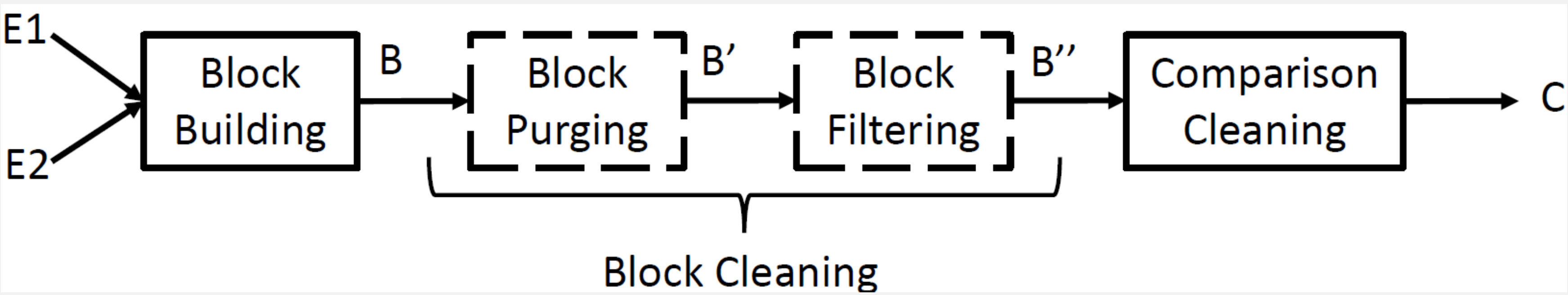
- Experimental analyses for:
  1. Blocking
  2. Matching

using the 10 established real-world datasets from various domains.

	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}
Dataset_1	Rest.1	Abt	Amazon	DBLP	IMDb	IMDb	TMDb	Walmart	DBLP	IMDb
Dataset_2	Rest.2	Buy	Google Pr.	ACM	TMDb	TVDB	TVDB	Amazon	Scholar	DBpedia
$\|V_1\|$	339	1,076	1,354	2,616	5,118	5,118	6,056	2,554	2,516	27,615
$\|V_2\|$	2,256	1,076	3,039	2,294	6,056	7,810	7,810	22,074	61,353	23,182
NVP_1	1,130	2,568	5,302	10,464	21,294	21,294	23,761	14,143	10,064	$1.6 \cdot 10^5$
NVP_2	7,519	2,308	9,110	9,162	23,761	20,902	20,902	$1.14 \cdot 10^5$	$1.98 \cdot 10^5$	$8.2 \cdot 10^5$
$\|A_1\|$	7	3	4	4	13	13	30	6	4	4
$\|A_2\|$	7	3	4	4	30	9	9	6	4	7
$\|\bar{p}_1\|$	3.33	2.39	3.92	4.00	4.16	4.16	3.92	5.54	4.00	5.63
$\|\bar{p}_2\|$	3.33	2.14	3.00	3.99	3.92	2.68	2.68	5.18	3.24	35.20
$\|D(V_1 \cap V_2)\|$	89	1,076	1,104	2,224	1,968	1,072	1,095	853	2,308	22,863
$\|\|V_1 \times V_2\|\|$	$7.65 \cdot 10^5$	$1.16 \cdot 10^6$	$4.11 \cdot 10^6$	$6.00 \cdot 10^6$	$3.10 \cdot 10^7$	$4.00 \cdot 10^7$	$4.73 \cdot 10^7$	$5.64 \cdot 10^7$	$1.54 \cdot 10^8$	$6.40 \cdot 10^8$

Filtering techniques [1]

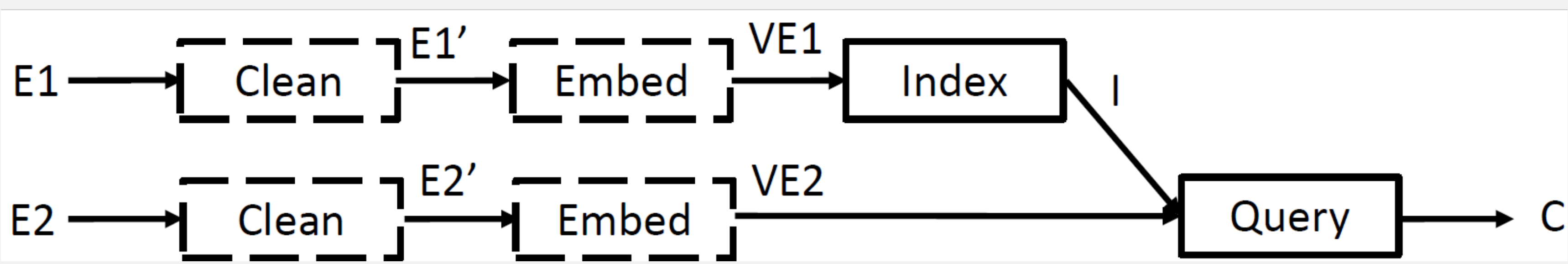
- Blocking workflow



- Nearest Neighbor workflow

i) Sparse Vectors → Similarity of token sets (Jaccard, Cosine, etc)

ii) Dense Vectors → Similarity of Vector Embeddings



[1] George Papadakis, Marco Fisichella, Franziska Schoger, George Mandilaras, Nikolaus Augsten, Wolfgang Nejdl. Benchmarking Filtering Techniques for Entity Resolution. ICDE 2023: 653-666

Evaluated approaches

Non-trivial comparison → Solution:

Configuration Optimization, i.e., maximize precision for a recall > τ, where τ = 0.85, 0.90, 0.95, …

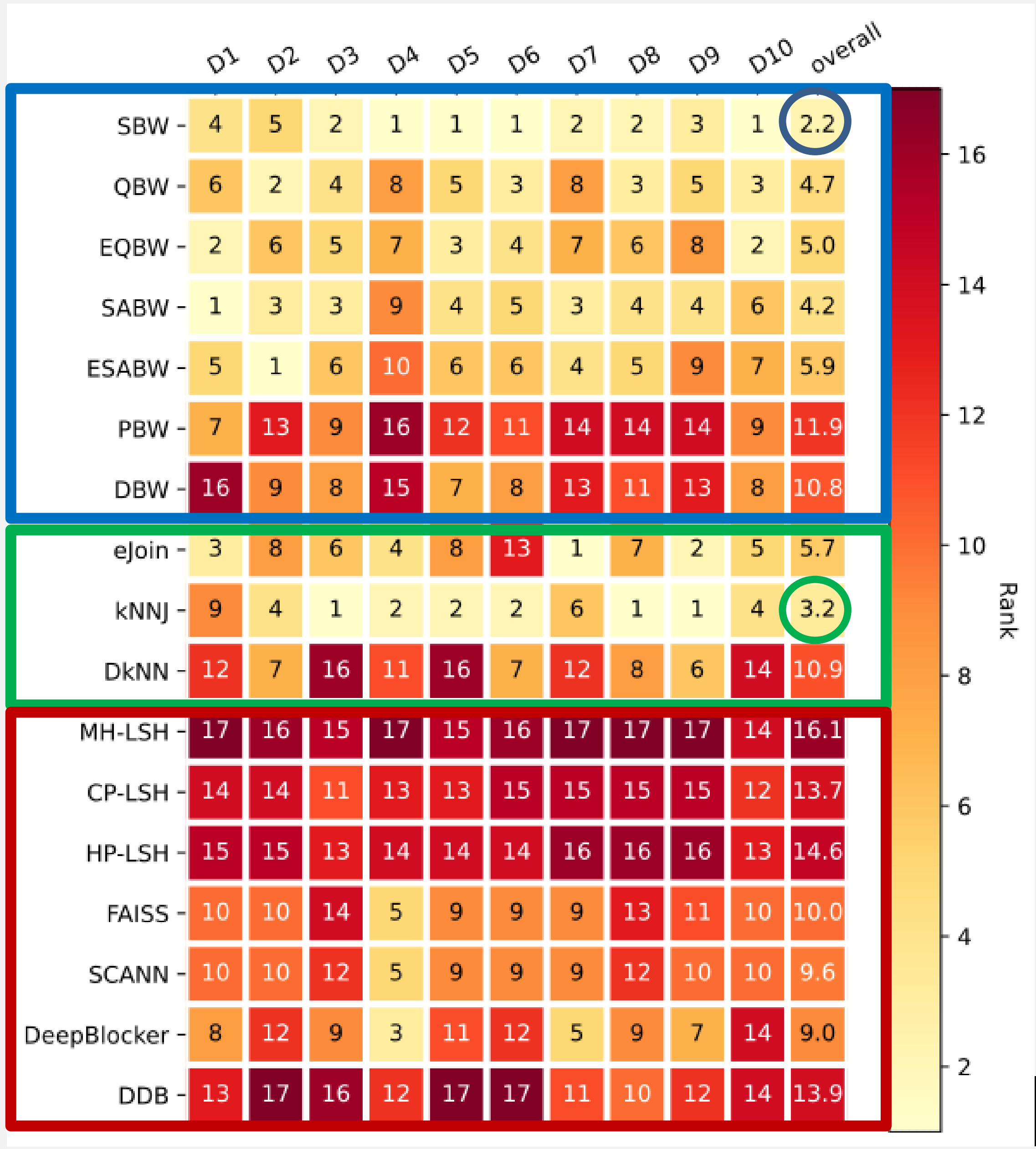
FastText embeddings

	Method	Number of Configurations
Blocking Methods	Standard Blocking	3,440
	Q-Grams Blocking	17,200
	Extended Q-Grams Blocking	68,800
	(Ex.) Suffix Arrays Blocking	21,285
Sparse NN Methods	ε-Join	6,000
	kNN-Join	12,000
Dense NN Methods	MH-LSH	168
	HP-LSH	400
	CP-LSH	2,000
	FAISS	2,720
	SCANN	10,880
	DeepBlocker	2,720

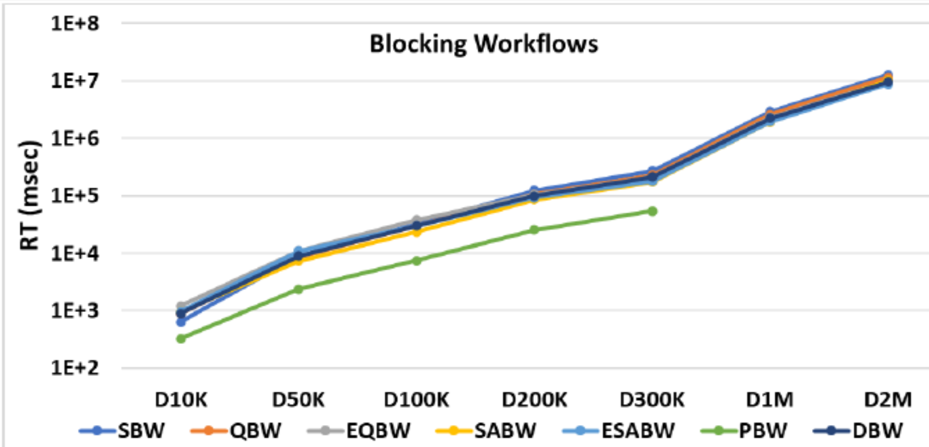
Schema-agnostic results

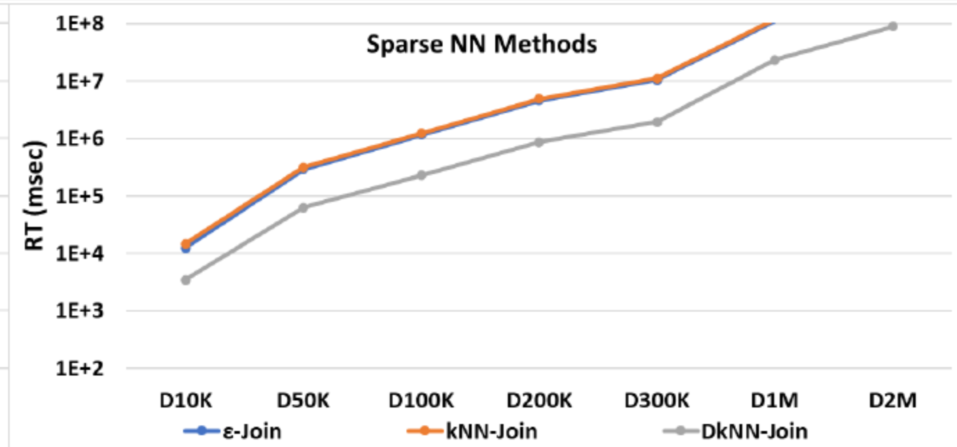
Conclusions:

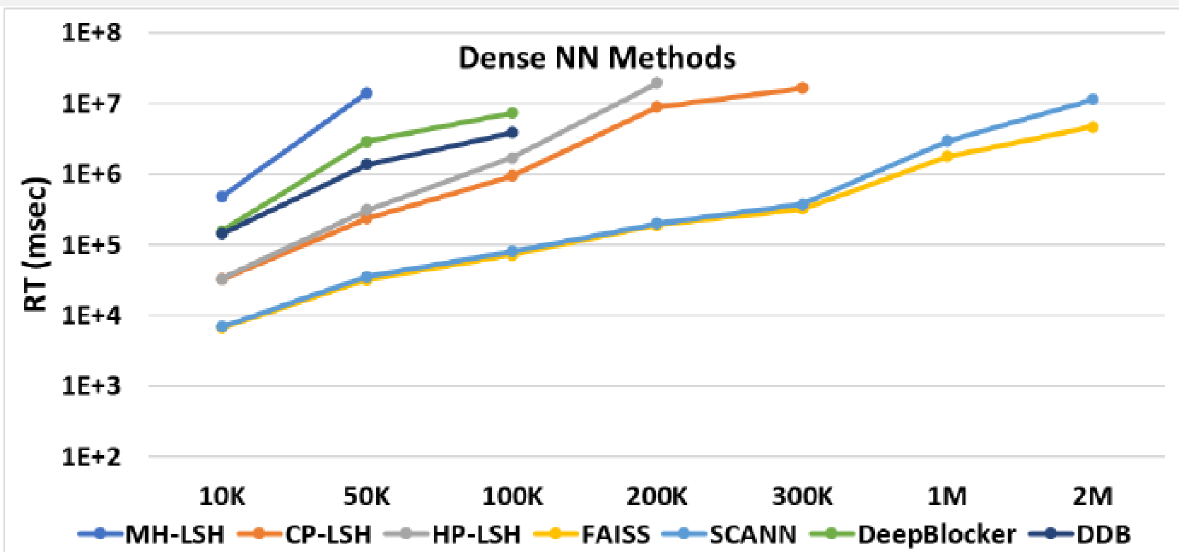
- Blocking workflows: Attribute value tokens yield better results than substrings of tokens.
- Sparse NN: Cardinality thresholds are more effective than similarity thresholds
- Dense NN: Similarity-based methods achieve high PC only with low precision. Learning-based tuple embedding raises the PQ, but does not scale.



Scalability







Conclusions

We used 10 real, established datasets for Record Linkage.
We compared 13 methods, all fine-tuned to each dataset.
We included 4 baseline methods with default configurations.

- PQ of all methods is highly correlated → performance heavily depends on dataset characteristics
- Parameter fine-tuning significantly increases blocking performance
- Schema-agnostic settings are preferable
- Cardinality thresholds are preferable
- Syntactic representations are preferable
  Conclusion verified by Sparkly [2].

[2] Derek Paulsen, Yash Govind, AnHai Doan: Sparkly: A Simple yet Surprisingly Strong TF/IDF Blocker for Entity Matching. Proc. VLDB Endow. 16(6): 1507-1519 (2023)

More language models for blocking [2]

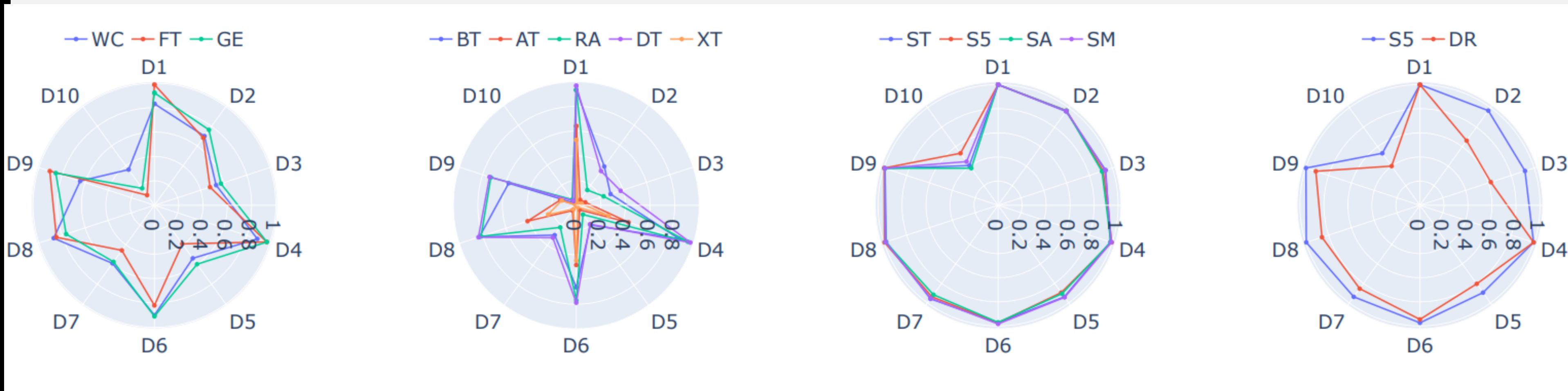
Three main types:

1. Static models → each word is transformed into a fixed, context-agnostic vector
   word2vec
2. BERT-based models → token and sentence embeddings that capture context via the multi-headed attention in their transformer design
3. SentenceBERT models → effective, dynamic sentence embeddings efficiently via their Siamese architecture

Static	BERT-based	SentenceBERT
Word2Vec (WC)	BERT (BT)	S-MPNet (ST)
GloVe (GE)	AlBERT (AT)	S-GTR-T5 (S5)
FastText (FT)	RoBERTa (RA)	S-DistilRoBERTa (SA)
	DistilBERT (DT)	S-MiniLM (SM)
	XLNet (XT)	

[2] Alexandros Zeakis, George Papadakis, Dimitrios Skoutas, Manolis Koubarakis: Pre-trained Embeddings for Entity Resolution: An Experimental Analysis. Proc. VLDB Endow. 16(9): 2225-2238 (2023)

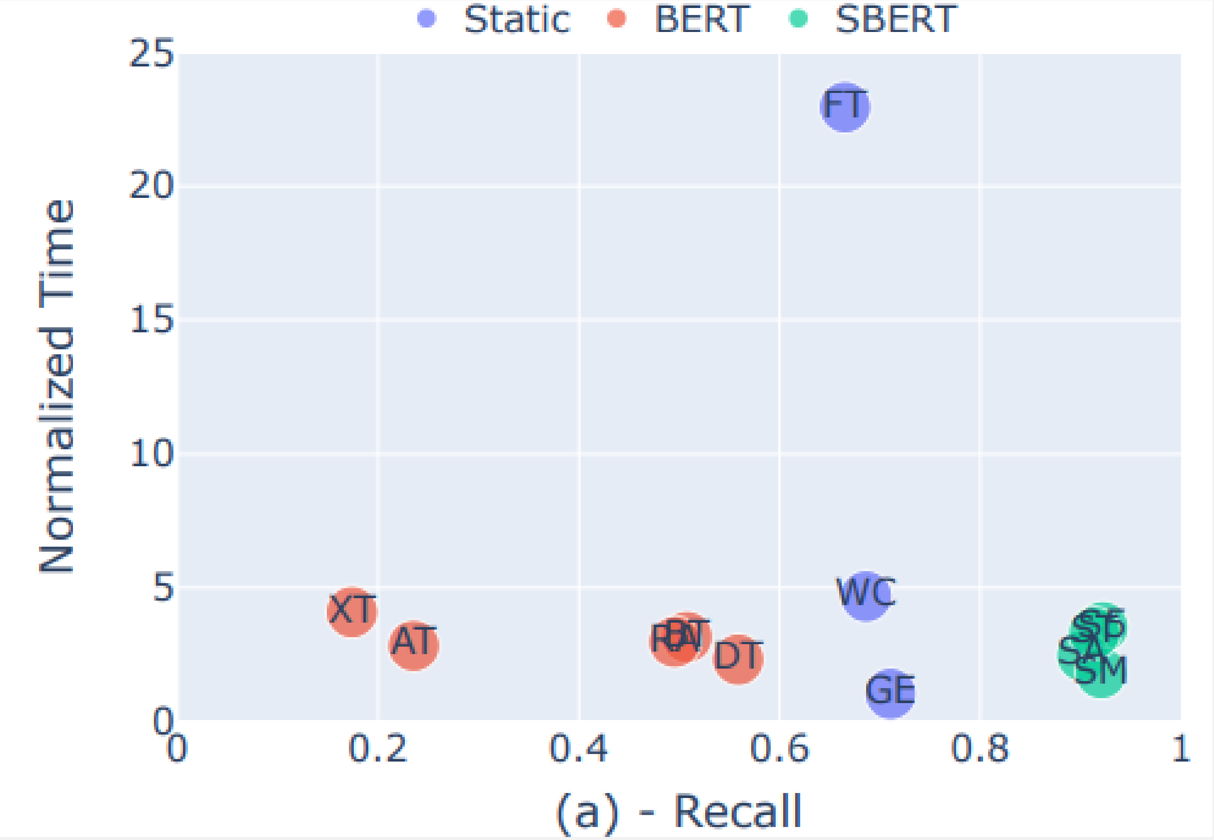
Performance Results



(a) Static (b) BERT (c) SBERT (d) SotA

- Plots show blocking recall with kNN search for k = 10.
- SentenceBERT based models (especially S-GTR-T5) perform very well.
- BERT based models (especially XLNet and AlBERT) perform poorly, even worse than static.
- Compared to DeepBlocker, S-GTR-T5 achieves about 15% higher recall on average, despite its lack of fine-tuning.

Overall Performance



- X-axis Blocking Recall, Y-axis normalized time (over fastest model).
- SentenceBERT models clearly outperform all others.

Comparison to SotA

[3]

k	S-GTR-T5	DeepBlocker	Sparkly	TokenJoin	kNN-Join
1	6.5%	19.5%	**6.1%**	9.2%	8.9%
5	**3.0%**	15.3%	4.2%	6.2%	4.4%
10	**2.6%**	13.5%	3.3%	4.9%	3.2%

Average recall distance per method and number of candidates (k) over the 10 datasets.

[3] Alexandros Zeakis, Dimitrios Skoutas, Dimitris Sacharidis, Odysseas Papapetrou, Manolis Koubarakis: TokenJoin: Efficient Filtering for Set Similarity Join with Maximum Weighted Bipartite Matching. Proc. VLDB Endow. 16(4): 790-802 (2022)

Unsupervised Matching [4][5]

We consider algorithms that:

1. Are crafted for bipartite similarity graphs (Clean-Clean ER)
   - Dirty ER was examined in [6], Multi-source ER by FAMER
2. Have a learning-free functionality
   - We perform fine-tuning based on the ground-truth
3. Time complexity ≤ $O(n^2)$
   - n stands for the number of input entities
   - E.g., the Hungarian algorithm is excluded, due to its cubic complexity, $O(n^3)$
4. Space complexity is $O(n+m)$
   - m denotes the number of edges

[4] George Papadakis, Vasilis Efthymiou, Emmanouil Thanos, Oktie Hassanzadeh: Bipartite Graph Matching Algorithms for Clean-Clean Entity Resolution: An Empirical Evaluation. EDBT 2022

[5] George Papadakis, Vasilis Efthymiou, Emmanouil Thanos, Oktie Hassanzadeh, Peter Christen: An analysis of one-to-one matching algorithms for entity resolution. VLDB J. 32(6): 1369-1400 (2023)

[6] Oktie Hassanzadeh, Fei Chiang, Renée J. Miller, Hyun Chul Lee: Framework for Evaluating Clustering Algorithms in Duplicate Detection. Proc. VLDB Endow. 2(1), (2009)

Selected Algorithms

1. Connected components (**CNC**) – O(m)
2. Ricochet Sequential Rippling Clustering (**RSR**) – O(n m)
3. Row Column Assignment Clustering (**RCA**) – $O(|V_1||V_2|)$
4. Best Assignment Heuristic (**BAH**)
   – Additional configuration parameter: 10,000 search steps or 2 min. run-time
5. Best Match Clustering (**BMC**) – O(m)
   – Additional configuration parameter: the node partition used as basis
6. Exact Clustering (**EXC**) – O(n m)
7. Király’s Clustering (**KRC**) – O(n + m log m)
8. Unique Mapping Clustering (**UMC**) – O(m log m)

Common configuration parameter: similarity threshold *t*

Weighting functions for similarity graphs

<table>
<tr><th colspan="2" rowspan="3"></th><th colspan="4">Scope</th></tr>
<tr><th colspan="2">Schema-agnostic</th><th colspan="2">Schema-based</th></tr>
<tr><th>Representation model</th><th>Similarity Measure</th><th>Representation model</th><th>Similarity Measure</th></tr>
<tr><td rowspan="3">Form</td><td rowspan="2">Syntactic Similarity</td><td>character n-grams (n=2,3,4)
and
token n-grams (n=1,2,3)</td><td>1) Arcs Similarity
2) Cosine Similarity with TF Weights
3) Cosine Similarity with TF-IDF Weights
4) Jaccard Similarity
5) Generalized Jaccard Similarity with TF Weights
6) Generalized Jaccard Similarity with TF-IDF Weights</td><td>Character-level</td><td>1) Damerau-Levenshtein
2) Levenshtein Distance
3) q-grams Distance
4) Jaro Similarity
5) Needleman Wunch
6)Longest Common Subsequence
7) Longest Common Substring</td></tr>
<tr><td>character n-gram graphs (n=2,3,4)
and
token n-gram graphs (n=1,2,3)</td><td>1) Containment Similarity
2) Value Similarity
3) Normalized Value Similarity
4) Overall Similarity</td><td>Token-level</td><td>1) Cosine Similarity
2) Monge-Elkan
3) Block Distance
4) Dice Similarity
5) Overlap Coefficient
6) Euclidean Distance
7) Jaccard Similarity
8) Generalized Jaccard Similarity
9) Euclidean Distance</td></tr>
<tr><td>Semantic Similarity</td><td>fastText
and
S-GTR-T5</td><td>1) Cosine Similarity
2) Euclidean Similarity
3) World Mover's Similarity</td><td>fastText
and
S-GTR-T5</td><td>1) Cosine Similarity
2) Euclidean Similarity
3) World Mover's Similarity</td></tr>
</table>

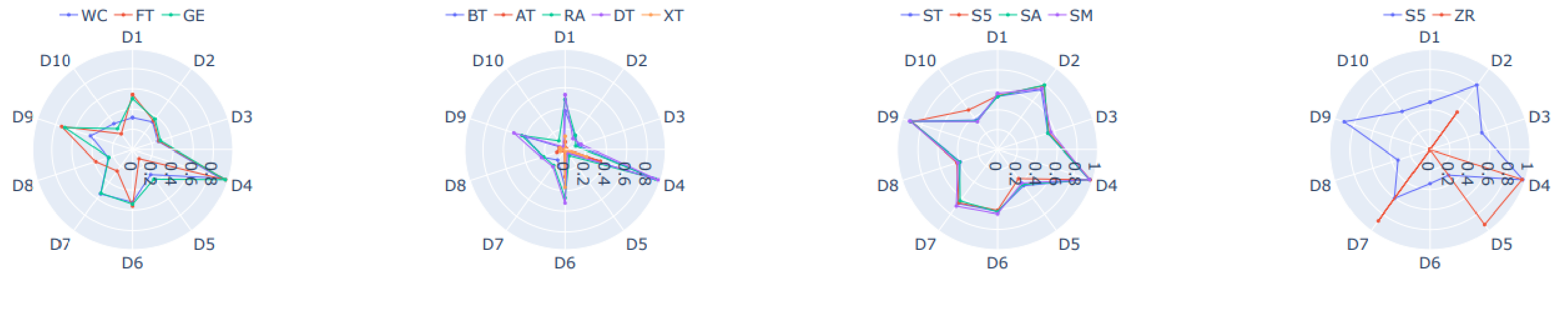
Comparison to SotA

- Baseline methods:
  - **ZeroER** [7], the state-of-the-art unsupervised matching algorithm
  - **DITTO** [8], the state-of-the-art deep learning-based matching algorithm
- Best clustering algorithm:
  - Unique Mapping Clustering coupled with:
    - schema-agnostic TF-IDF weights
    - cosine similarity
- Results w.r.t. to F-measure:

Dataset	ZeroER	DITTO	UMC	configuration
D2	0.52	0.89	**0.95**	character bi-grams, t=0.35
D3	0.48	0.76	0.60	token bi-grams, t=0.05
D4	0.96	**0.99**	**0.99**	token uni-grams, t=0.40
D5	0.86	0.96	0.94	character four-grams, t=0.35

[7] Renzhi Wu et al. ZeroER: Entity Resolution using Zero Labeled Examples. In SIGMOD (2020). 1149–1164.
[8] Yuliang Li et al. Deep Entity Matching with Pre-Trained Language Models. Proc. VLDB Endow. 14, 1 (2020), 50–60.

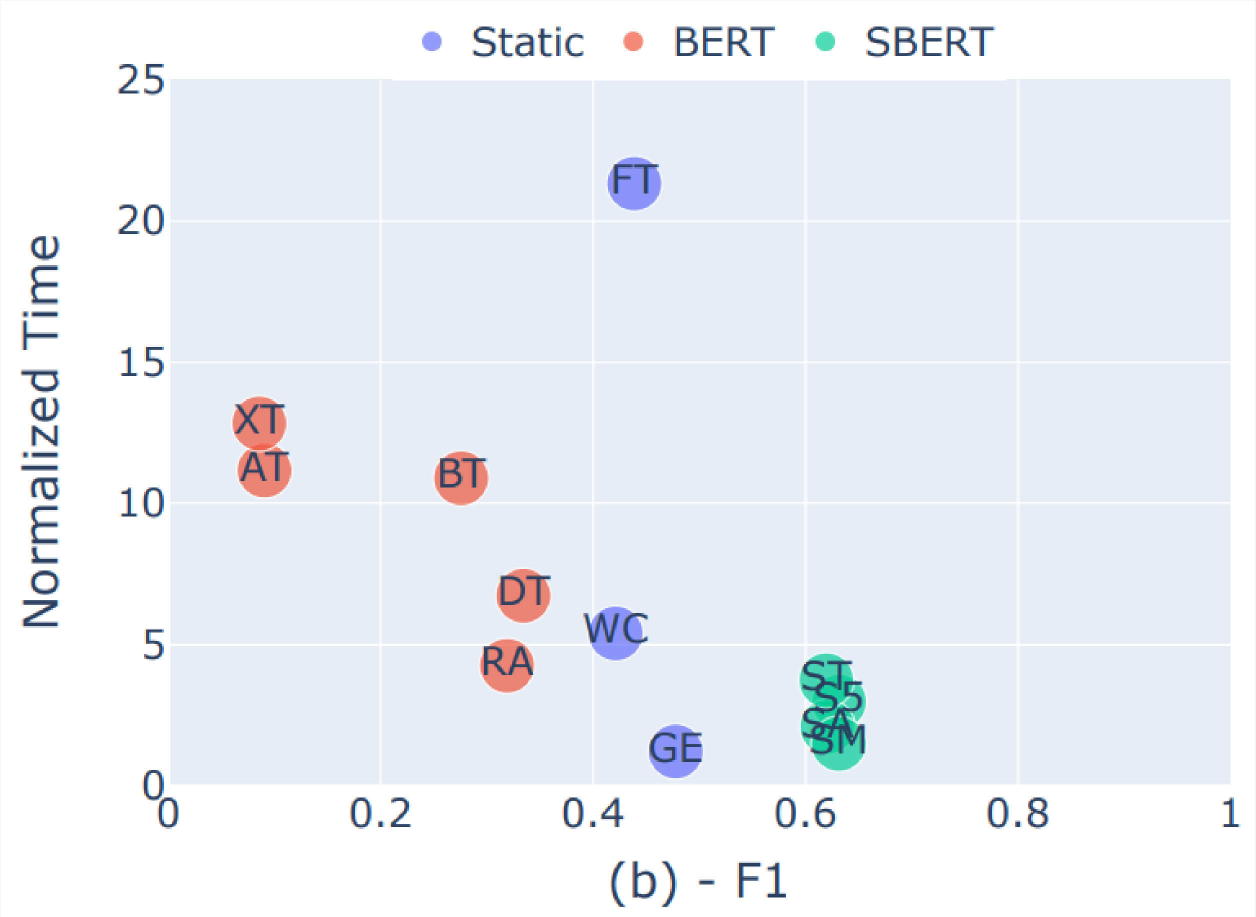
More language models for matching [2]



(a) Static models (b) BERT models (c) SBERT models (d) State-of-the-art

- Plots show maximum F1 score per case.
- SentenceBERT based models (especially S-GTR-T5) perform very well.
- BERT based models (especially XLNet and AlBERT) perform poorly, even worse than static.
- Compared to ZeroER, S-GTR-T5 (end-to-end, k = 10) is clearly better.
- ZeroER even fails to terminate in most cases.

Overall Performance



- X-axis F1 Measure, Y-axis normalized time (over fastest model).
- SBERT models clearly outperform all others.

- Introduction
- The First Four Generations
- The Fifth Generation:
  Leveraging External Knowledge

Part D – Final Remarks and Challenges

Conclusions

- The five generations scheme allows for easily categorizing works on ER.
- The 5th generation dominates most recent state-of-the-art works.
- Language models, especially SentenceBERT ones, achieve top performance in both blocking and matching.

Challenge 1 – Large Language Models

- Increasingly replacing language models, e.g., [1][2][3]
- So far: straightforward applications based on prompt engineering
  - Zero, one, few shot prompting
  - Hand-written matching rules
  - Performance comparable to DL-based matching algorithms
- Challenge: *leverage LLMs in more complex ways, e.g., end-to-end workflows*

[1] Avanika Narayan, Ines Chami, Laurel Orr, and Christopher Ré. 2022. Can Foundation Models Wrangle Your Data? Proceedings of the VLDB Endowment 16, 4 (2022), 738–746

[2] Ralph Peeters and Christian Bizer. 2023. Using ChatGPT for Entity Matching. In New Trends in Database and Information Systems (Communications in Computer and Information Science). Springer Nature Switzerland, Cham, 221–230.

[3] Peeters, Ralph, and Christian Bizer. "Entity Matching using Large Language Models." arXiv preprint arXiv:2310.11244 (2023).

Challenge 2 – ER systems

- Literature focuses on stand-alone methods
- More emphasis on end-to-end systems
  - E.g., https://github.com/AI-team-UoA/pyJedAI
  - Library that partially covers the five generations
  - Processes data of any structuredness
  - Open-source, extensible based on the Python ecosystem
- Challenge: *how to integrate learning-based algorithms into end-to-end pipelines?*

Challenge 3 – Automatic Configuration

Facts:

- Several parameters in every method
  - Applies to all generations and workflow steps
- Performance sensitive to internal configuration
- Fine-tuning required, but huge configuration space for end-to-end pipelines

Challenge [4]:

optimize parameter configuration when:

1. Known ground-truth and pipeline
2. Known ground-truth, but unknown pipeline
3. Unknown ground-truth and pipeline

STELAR

Spatio-TEmporal Linked data tools for the AgRi-food data space

[4] Vasilis Efthymiou et al. Self-configured Entity Resolution with pyJedAI. 2023 IEEE International Conference on Big Data.

Thank You!