Entity Resolution: Past, Present and Yet-to-Come
From Structured to Heterogeneous, to Crowd-sourced, to Deep Learned

George Papadakis
National and Kapodistrian
University of Athens, Greece
gpapadis@di.uoa.gr

Ekaterini Ioannou
University of Tilburg, Netherlands
Ekaterini.Ioannou@uvt.nl

Themis Palpanas
University of Paris, France
themis@mi.parisdescartes.fr

ABSTRACT
Entity Resolution (ER) lies at the core of data integration, with a bulk of research focusing on its effectiveness and its time efficiency. Most past relevant works were crafted for addressing Veracity over structured (relational) data. They typically rely on schema, expert and external knowledge to maximize accuracy. Part of these methods have been recently extended to process large volumes of data through massive parallelization techniques, such as the MapReduce paradigm. With the present advent of Big Web Data, the scope moved towards Variety, aiming to handle semi-structured data collections, with noisy and highly heterogeneous information. Relevant works adopt a novel, loosely schema-aware functionality that emphasizes scalability and robustness to noise. Another line of present research focuses on Velocity, i.e., processing data collections of a continuously increasing volume.

In this tutorial, we present the ER generations by discussing past, present, and yet-to-come mechanisms. For each generation, we outline the corresponding ER workflow along with the state-of-the-art methods per workflow step. Thus, we provide the participants with a deep understanding of the broad field of ER, highlighting the recent advances in crowd-sourcing and deep learning applications in this active research domain. We also equip them with practical skills in applying ER workflows through a hands-on session that involves our publicly available ER toolbox and data.

1 GOALS AND OBJECTIVES
Entity profiles assemble valuable information about real-world objects. Hence, entities constitute the core organizational unit of *structured* (e.g., relational databases) as well as *semi-structured data* (e.g., knowledge bases, such as DBPedia and Geonames). Various data management applications, such as query answering [47], are based on entity semantics and connections in order to improve their performance. Typically, these applications require the integration of different profiles that pertain to the same real-world object [11, 18]. The task of inter-linking and deduplicating (i.e., canonicalizing) data instances that describe the same real-world objects is called *Entity Resolution* (ER) [12].

ER is a relatively old problem that was mainly crafted for structured data, which were described by schemata of known semantics and quality [11]. This schema knowledge allowed experts to develop customized solutions that effectively addressed **Veracity**, i.e., the various forms of inconsistencies, noise or errors in entity profiles, which are introduced during manual data entry, or by the limitations of the automatic extraction techniques [23]. For even higher effectiveness, labelled instances are also typically used in order to automatically learn matching rules that simultaneously maximize precision and recall [48, 60, 61].

Figure 1: The workflow of the 1^{st} and 2^{nd} ER generations.

The *end-to-end* workflow implemented by **the 1^{st} generation** of ER solutions is depicted in Figure 1 [11]. The first step, *Schema Matching*, creates mappings between the attributes of the input entities based on their relatedness, as inferred from the similarity of their structure, name and/or values [6, 45]. By identifying semantically identical attributes (e.g., “profession” and “job”), it facilitates the schema-aware functionality of the subsequent workflow steps.

The second step, which is called *Blocking*, addresses the quadratic time complexity, $O(n^2)$, of brute-force ER, which compares every entity profile with all others [11]. Blocking reduces the executed comparisons to a significant extent by sacrificing recall to a minor extent. It restricts the computational cost by comparing only the most similar entity profiles, as they are determined by signatures that are composed of (combinations of) parts of values that correspond to the most informative attribute names [11]. E.g., two person entities are likely matches if their addresses have the same zip code.

The entities that co-occur in at least one block are compared during the third step, which is called *Entity Matching*. This applies a combination of string similarity measures to the values of selected attribute names. The resulting degree of similarity is then used to assign the entity pairs into one of the three possible categories, i.e., `match`, `non-match` or `uncertain` [11]. In case of collective approaches, the latest decision is propagated to *neighboring entities*, i.e., entities connected with important relationships to the compared pair, so as to refine their matching likelihood [7, 16].

Note that each step accommodates both *learning-based* and *non-learning methods* [41]; the former methods leverage labelled instances to extract effective rules through a Machine Learning algorithm, while the latter methods rely on heuristics that capture expert or domain knowledge.

The same workflow lies at the core of **the 2^{nd} generation**, which additionally targets **Volume**, i.e., the cases where the input data comprise (tens of) millions of entity profiles. Typically, this challenge is addressed through the new paradigm for *massive parallelization*, i.e., Map/Reduce [14]. Several techniques for Blocking [38] and Entity Matching [9] have been adapted to MapReduce so that they scale to voluminous datasets. Special care is also taken to avoid underutilization of the computational resources through Load Balancing techniques [39, 77].

A shift was marked by **the 3^{rd} generation** of the ER end-to-end workflow, which is depicted in Figure 2. In addition to Veracity and Volume, its goal is to address **Variety**, which is caused by the unprecedented levels of schema heterogeneity and noise as well as the loose schema binding of unclear semantics [12, 17]. Instead of a database-like schema, there is a rich diversity of

Series ISSN: 2367-2005 10.5441/002/edbt.2020.85

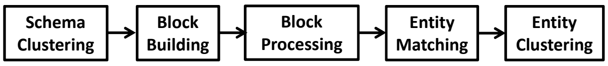

Figure 2: The workflow of the 3^{rd} ER generation.

the domains. For example, there are ∼2,600 different vocabularies in the LOD cloud but only 109 from them are shared by more than one entity collection [22]. This results in hundreds, or even thousands, of different attributes with high entity frequency, rendering inapplicable the schema-aware methods of the first two generations [31, 54].

The first step in the new workflow is *Schema Clustering*, which clusters together attributes with similar values, regardless of their semantics. The goal is to improve the performance of the subsequent steps. E.g., Blocking uses the created schema clusters and the associated signatures (i.e., blocking keys) to split large blocks into smaller ones. This significantly enhances precision for a negligible (if any) impact on recall. This idea has been successfully applied to Blocking via Attribute Clustering [52] and to Meta-blocking via BLAST [62].

The second step, which is called *Block Building*, creates a set of blocks by disregarding schema knowledge and the ensuing human intervention completely. Through a schema-agnostic approach that leverages redundancy, it is inherently crafted for tackling the unprecedented levels of schema heterogeneity in semi-structured data. In this way, it yields blocks of very high recall, but very low precision, independently of human intervention and domain/expert knowledge [12, 54].

The third step of the workflow is *Block Processing*, which enhances precision to a significant extent at a limited, if any, cost in recall [52, 54, 56]. To this end, it refines the original blocks by efficiently removing comparisons that are repeated or involve non-matching entities. Its techniques are distinguished into two categories: the *Block Cleaning* ones operate at the coarse-grained level of entire blocks (e.g., Block Clustering [26]), while the *Comparison Cleaning* ones operate at the fine-grained level of individual comparisons (e.g., Meta-blocking [18, 53] and Blast [62]). In both cases, all techniques are generic and schema-agnostic by definition, thus applying naturally to both structured and semi-structured data [56].

Subsequently, *Entity Matching* executes all comparisons contained in the final set of blocks. Typically, this process depends heavily on neighbor similarity, using the entity relations in the semi-structured data. This is done through an iterative process that discovers duplicate entities gradually and propagates the latest matches to related entities that could benefit from them [42, 44, 66]. This step can also consider probabilistic matching of the entities, e.g., [1, 36].

The end result of Entity Matching is a *similarity graph*, which conveys a node for every entity and a weighted edge for every pair of entities that have been compared. This intermediate model is transformed into the final outcome of ER by *Entity Clustering* [34], which partitions the graph nodes into equivalence clusters - every cluster contains all duplicate entity profiles that actually correspond to the same real-world object. These techniques are schema-agnostic by default, as they exclusively consider the information contained in the similarity graph.

The 4^{th} generation of ER goes beyond the previous ones, by also addressing **Velocity**. This pertains to the continuously increasing volume of available data that imposes special ER challenges, e.g., the data set can never be considered as final, and incoming data might alter the existing ones. To address them,

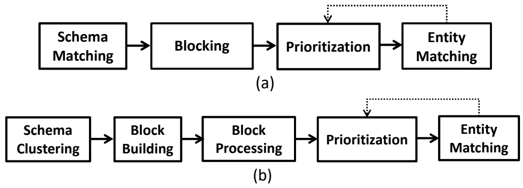

Figure 3: The end-to-end workflow of the 4^{th} ER generation for (a) structured and (b) semi-structured data.

Progressive ER produces useful results in a pay-as-you-go manner before the full completion of ER. Figures 3(a) and (b) illustrate the schema-aware [58] and the schema-agnostic [63] workflows. *Incremental ER* [33] is another mechanism, which minimizes the cost for updating the existing results when new evidence becomes available. Some methods also consider that the new evidence might be conflicting with already processed data [36]. Another mechanism is *Query-driven ER* [4], which gradually resolves entities that are returned as results to incoming queries. A different mechanism is supporting queries for obtaining aggregate, statistical insights about the collection of resulting entities [35].

Note that all generations can be upgraded by exploiting **external knowledge** to achieve higher performance. To this category fall *Deep Learning* techniques for ER [20, 48], which incorporate contextual information in the form of word- or character-embeddings, and *Crowd-sourced ER* [13, 15, 24, 30, 32, 67–70, 72, 75], which relies on human feedback. These two approaches lie at the focus of the latest breakthroughs in ER.

After examining the four ER generations, our tutorial proceeds with a hands-on session that focuses on the state-of-the-art tools for end-to-end Entity Resolution, like Magellan [40]. We then present *JedAI* [57], which constitutes a comprehensive, open-source toolkit that implements most of the state-of-the-art methods for every step of the 3^{rd} and 4^{th} ER generations. Thus, it enables users to build versatile workflows on-the-fly and can be readily used both for experimentation and for integration in Entity Resolution applications. It is distributed under the Apache License 2.0 and, thus, it is suitable for both the academic and the commercial domain.

Overall, our tutorial provides researchers with a complete coverage of the state-of-the-art ER methods along with a discussion of the main open research problems. Practitioners get a good overview of the benefits of the primary ER methods and learn how to use them to improve the productivity of their businesses. They also learn to identify the methods or products that are more suitable for a particular task at hand, or better fit their general needs. Additionally, the audience and especially the developers of information integration tools benefit from the hands-on session, learning how to integrate (parts of) the JedAI Toolkit into their applications. Developers also become acquainted with novel ideas that could well improve their existing products.

Related Tutorials. Our tutorial provides for the first time a novel holistic and systematic view of the evolution of ER, stressing the current state-of-the-art in deep learning and crowdsourcing applications. We categorize the main ER methods into four generations, going from those crafted for maximizing Veracity over structured data, all the way to those tackling Veracity, Volume, Variety and Velocity over semi-structured data. No other tutorial covers comprehensively large-scale, end-to-end ER for both structured and semi-structured data. Past tutorials on the subject [17, 27, 29, 65] focus either on one of these data types, or cover partially the end-to-end ER workflow.

2 SCOPE AND COVERAGE

Our tutorial aims to provide an overview of the state-of-the-art techniques for all generations of End-to-End ER, analyzing each one in a different session of ~10 minutes. More emphasis is devoted to the approaches leveraging external knowledge in order to upgrade any workflow step in any generation (~30 minutes), while a hands-on session discusses the main ER tools and demonstrates the latest version of JedAI (~10 minutes). Together with the introduction, 5 minutes for questions and the conclusions, the intended duration of the tutorial is *1.5 hours.* The content of the individual sessions is outlined below:

I. Introduction and motivation
- Preliminaries on Entity Resolution [12, 18]
- Fundamental Assumptions, Principles and Definitions [23]

II. The 1st ER Generation: Tackling Veracity
- Schema Matching [6, 19]
- Blocking [11, 37, 61]
- Entity Matching [5, 16, 60]

III. The 2nd ER Generation: Tackling Volume and Veracity
- Parallel Blocking [38]
- Parallel Entity Matching [59]
- Load Balancing [39, 77]

IV. The 3rd ER Generation: Tackling Variety, Volume and Veracity
- Schema Clustering [52, 62]
- Block Building [50–52]: Parallel Methods [12]
- Block Processing [8, 26, 55, 56, 62]: Parallel Methods [21]
- Entity Matching [42, 44, 66]: Parallel Methods [9, 22]
- Entity Clustering [34]

V. The 4th ER Generation: Tackling Velocity, Variety, Volume and Veracity
- Progressive ER for (Semi-)Structured Data [58, 63, 76]
- Incremental Entity Resolution [33, 74]
- Query-Driven Entity Resolution [2–4, 71]
- Query Analytics for Entity Resolution [35, 64].

VI. Entity Resolution Revisited: Leveraging External Knowledge
- Deep Learning for Entity Resolution [20, 48]
- Crowd-sourced Entity Resolution:
  - Generating HITs [15, 43, 70]
  - Formulating HIts [25, 67–69, 72, 75]
  - Balancing accuracy and monetary cost [10, 28, 73, 78]
  - Restrict the labour cost [13, 30]

VII. Hands-on Session: ER tools
- The state-of-the-art end-to-end ER tools [40]
- The JedAI Open Source Toolkit [57]

VIII. Challenges and Final Remarks
- Automatic Parameter Configuration [46, 49]
- Multi-modal Entity Resolution
- Conclusions

3 INTENDED AUDIENCE AND MATERIAL

Our tutorial is example-driven, avoiding excessive technical details and proofs. As a result, there is no prerequisite knowledge, apart from a basic understanding of data management technology. This renders it suitable for a broad audience, covering not only students and researchers, but also practitioners and developers. In other words, it is intended for anyone with an interest in understanding the main techniques for scalable and robust end-to-end Entity Resolution over structured and semi-structured data, using both non-learning and learning-based techniques.

In addition to the theoretical background in the state-of-the-art in the field, the tutorial also presents available entity-related resources, enabling the participants to directly work on the particular domain. Discussed resources include available data as well as the state-of-the-art tools for performing end-to-end Entity Resolution, like Magellan [40] and JedAI [57], which can be readily used to tackle ER problems via numerous combinations of the most prominent methods.

Tutorial Material. The material of the tutorial is distributed through the conference website[1] as well as through a dedicated website[2]. In both locations, we also give pointers and guidelines for the ER toolkit that is used during the hands-on session. All relevant code is publicly released through the Apache License 2.0, which supports both academic and commercial uses.

4 PRESENTERS

The tutorial is given by three presenters:

(1) *George Papadakis* is a Research Fellow at the Department of Informatics of the University of Athens, Greece, and an Internal Auditor of Information Systems at the Public Power Company, the main electricity company in Greece.
(2) *Ekaterini Ioannou* is an Assistant Professor at the University of Tilburg, Netherlands.
(3) *Themis Palpanas* is a Senior Member of the French University Insitute (IUF), and a Professor of Computer Science at the University of Paris, France.

All authors have published papers related to Entity Resolution, focusing on the efficient management of large data collections as well as on addressing various challenges, such as uncertainty, volatility, and correlations.

Acknowledgements. This work was partially funded by the EU H2020 project ExtremeEarth (825258).

REFERENCES

[1] Parag Agrawal, Omar Benjelloun, Anish Das Sarma, Chris Hayworth, Shubha U. Nabar, Tomoe Sugihara, and Jennifer Widom. 2006. Trio: A System for Data, Uncertainty, and Lineage. In *VLDB*. 1151–1154.
[2] Hotham Altwaijry, Dmitri Kalashnikov, and Sharad Mehrotra. 2013. Query-Driven Approach to Entity Resolution. *PVLDB* 6, 14 (2013), 1846–1857.
[3] Hotham Altwaijry, Dmitri Kalashnikov, and Sharad Mehrotra. 2017. QDA: A Query-Driven Approach to Entity Resolution. *TKDE* (2017).
[4] Hotham Altwaijry, Sharad Mehrotra, and Dmitri V. Kalashnikov. 2015. QuERy: A Framework for Integrating Entity Resolution with Query Processing. *PVLDB* 9, 3 (2015), 120–131.
[5] Omar Benjelloun, Hector Garcia-Molina, David Menestrina, Qi Su, Steven Euijong Whang, and Jennifer Widom. 2009. Swoosh: a generic approach to entity resolution. *VLDB J.* 18, 1 (2009), 255–276.
[6] Philip A. Bernstein, Jayant Madhavan, and Erhard Rahm. 2011. Generic Schema Matching, Ten Years Later. *PVLDB* 4, 11 (2011), 695–701.
[7] Indrajit Bhattacharya and Lise Getoor. 2007. Collective entity resolution in relational data. *TKDD* 1, 1 (2007), 5.
[8] Guilherme Dal Bianco, Marcos André Gonçalves, and Denio Duarte. 2018. BLOSS: Effective meta-blocking with almost no effort. *Inf. Syst.* 75 (2018), 75–89.
[9] Christoph Böhm, Gerard de Melo, Felix Naumann, and Gerhard Weikum. 2012. LINDA: distributed web-of-data-scale entity matching. In *CIKM*. 2104–2108.
[10] Chengliang Chai, Guoliang Li, Jian Li, Dong Deng, and Jianhua Feng. 2018. A partial-order-based framework for cost-effective crowdsourced entity resolution. *VLDB J.* 27, 6 (2018), 745–770.
[11] Peter Christen. 2012. *Data Matching*. Springer.
[12] Vassilis Christophides, Vasilis Efthymiou, and Kostas Stefanidis. 2015. *Entity Resolution in the Web of Data.* Morgan & Claypool Publishers.
[13] Sanjib Das, Paul Suganthan G. C., AnHai Doan, Jeffrey F. Naughton, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, Vijay Raghavendra, and Youngchoon Park. 2017. Falcon: Scaling Up Hands-Off Crowdsourced Entity Matching to Build Cloud Services. In *SIGMOD*. 1431–1446.
[14] Jeffrey Dean and Sanjay Ghemawat. 2008. MapReduce: simplified data processing on large clusters. *Commun. ACM* 51, 1 (2008), 107–113.

[1] https://diku-dk.github.io/edbticdt2020
[2] https://research.tilburguniversity.edu/en/projects/4ger

[15] Gianluca Demartini, Djellel Eddine Difallah, and Philippe Cudré-Mauroux. 2013. Large-scale linked data integration using probabilistic reasoning and crowdsourcing. *VLDB J.* 22, 5 (2013), 665–687.
[16] Xin Dong, Alon Y. Halevy, and Jayant Madhavan. 2005. Reference Reconciliation in Complex Information Spaces. In *SIGMOD*. 85–96.
[17] Xin Luna Dong and Divesh Srivastava. 2013. Big Data Integration. *PVLDB* 6, 11 (2013), 1188–1189.
[18] Xin Luna Dong and Divesh Srivastava. 2015. *Big Data Integration*. Morgan & Claypool Publishers.
[19] Songyun Duan, Achille Fokoue, Oktie Hassanzadeh, Anastasios Kementsietsidis, Kavitha Srinivas, and Michael J. Ward. 2012. Instance-Based Matching of Large Ontologies Using Locality-Sensitive Hashing. In *ISWC*. 49–64.
[20] Muhammad Ebraheem, Saravanan Thirumuruganathan, Shafiq R. Joty, Mourad Ouzzani, and Nan Tang. 2018. Distributed Representations of Tuples for Entity Resolution. *PVLDB* 11, 11 (2018), 1454–1467.
[21] Vasilis Efthymiou, George Papadakis, George Papastefanatos, Kostas Stefanidis, and Themis Palpanas. 2017. Parallel meta-blocking for scaling entity resolution over big heterogeneous data. *Inf. Syst.* (2017).
[22] Vasilis Efthymiou, George Papadakis, Kostas Stefanidis, and Vassilis Christophides. 2019. MinoanER: Schema-Agnostic, Non-Iterative, Massively Parallel Resolution of Web Entities. In *EDBT*. 373–384.
[23] Ahmed K. Elmagarmid, Panagiotis G. Ipeirotis, and Vassilios S. Verykios. 2007. Duplicate Record Detection: A Survey. *TKDE* 19, 1 (2007), 1–16.
[24] Donatella Firmani, Sainyam Galhotra, Barna Saha, and Divesh Srivastava. 2018. Robust Entity Resolution Using a CrowdOracle. *IEEE Data Eng. Bull.* 41, 2 (2018), 91–103.
[25] Donatella Firmani, Barna Saha, and Divesh Srivastava. 2016. Online Entity Resolution Using an Oracle. *PVLDB* 9, 5 (2016), 384–395.
[26] Jeffrey Fisher, Peter Christen, Qing Wang, and Erhard Rahm. 2015. A Clustering-Based Framework to Control Block Sizes for Entity Resolution. In *KDD*. 279–288.
[27] Avigdor Gal. 2014. Tutorial: Uncertain Entity Resolution. *PVLDB* 7, 13 (2014), 1711–1712.
[28] Sainyam Galhotra, Donatella Firmani, Barna Saha, and Divesh Srivastava. 2018. Robust Entity Resolution using Random Graphs. In *SIGMOD*. 3–18.
[29] Lise Getoor and Ashwin Machanavajjhala. 2012. Entity Resolution: Theory, Practice & Open Challenges. *PVLDB* 5, 12 (2012), 2018–2019.
[30] Chaitanya Gokhale, Sanjib Das, AnHai Doan, Jeffrey F. Naughton, Narasimhan Rampalli, Jude W. Shavlik, and Xiaojin Zhu. 2014. Corleone: hands-off crowdsourcing for entity matching. In *SIGMOD*. 601–612.
[31] Behzad Golshan, Alon Halevy, George Mihaila, and Wang-Chiew Tan. 2017. Data Integration: After the Teenage Years. In *PODS*. 101–106.
[32] Yash Govind, Erik Paulson, Palaniappan Nagarajan, Paul Suganthan G. C., AnHai Doan, Youngchoon Park, Glenn Fung, Devin Conathan, Marshall Carter, and Mingju Sun. 2018. CloudMatcher: A Hands-Off Cloud/Crowd Service for Entity Matching. *PVLDB* 11, 12 (2018), 2042–2045.
[33] Anja Gruenheid, Xin Luna Dong, and Divesh Srivastava. 2014. Incremental Record Linkage. *PVLDB* 7, 9 (2014), 697–708.
[34] Oktie Hassanzadeh, Fei Chiang, Renée J. Miller, and Hyun Chul Lee. 2009. Framework for Evaluating Clustering Algorithms in Duplicate Detection. *PVLDB* 2, 1 (2009), 1282–1293.
[35] Ekaterini Ioannou and Minos Garofalakis. 2015. Query Analytics over Probabilistic Databases with Unmerged Duplicates. *TKDE* 27, 8 (2015), 2245–2260.
[36] Ekaterini Ioannou, Wolfgang Nejdl, Claudia Niederée, and Yannis Velegrakis. 2010. On-the-Fly Entity-Aware Query Processing in the Presence of Linkage. *PVLDB* 3, 1 (2010), 429–438.
[37] Mayank Kejriwal and Daniel P. Miranker. 2013. An Unsupervised Algorithm for Learning Blocking Schemes. In *ICDM*. 340–349.
[38] Lars Kolb, Andreas Thor, and Erhard Rahm. 2012. Dedoop: Efficient Deduplication with Hadoop. *PVLDB* 5, 12 (2012), 1878–1881.
[39] Lars Kolb, Andreas Thor, and Erhard Rahm. 2012. Load Balancing for MapReduce-based Entity Resolution. In *ICDE*. 618–629.
[40] Pradap Konda, Sanjib Das, Paul Suganthan G. C., AnHai Doan, Adel Ardalan, Jeffrey R. Ballard, Han Li, Fatemah Panahi, Haojun Zhang, Jeffrey F. Naughton, Shishir Prasad, Ganesh Krishnan, Rohit Deep, and Vijay Raghavendra. 2016. Magellan: Toward Building Entity Matching Management Systems. *PVLDB* 9, 12 (2016), 1197–1208.
[41] Hanna Köpcke, Andreas Thor, and Erhard Rahm. 2010. Evaluation of entity resolution approaches on real-world match problems. *PVLDB* 3, 1 (2010), 484–493.
[42] Simon Lacoste-Julien, Konstantina Palla, Alex Davies, Gjergji Kasneci, Thore Graepel, and Zoubin Ghahramani. 2013. SIGMa: simple greedy matching for aligning large knowledge bases. In *KDD*. 572–580.
[43] Guoliang Li, Yudian Zheng, Ju Fan, Jiannan Wang, and Reynold Cheng. 2017. Crowdsourced Data Management: Overview and Challenges. In *SIGMOD*. 1711–1716.
[44] Juanzi Li, Jie Tang, Yi Li, and Qiong Luo. 2009. RiMOM: A Dynamic Multistrategy Ontology Alignment Framework. *TKDE* 21, 8 (2009), 1218–1232.
[45] Jayant Madhavan, Philip A. Bernstein, and Erhard Rahm. 2001. Generic Schema Matching with Cupid. In *VLDB*. 49–58.
[46] Ruhaila Maskat, Norman W. Paton, and Suzanne M. Embury. 2016. Pay-as-you-go Configuration of Entity Resolution. *T. Large-Scale Data- and Knowledge-Centered Systems* (2016), 40–65.
[47] Davide Mottin, Matteo Lissandrini, Yannis Velegrakis, and Themis Palpanas. 2016. Exemplar queries: a new way of searching. *VLDB J.* 25, 6 (2016), 741–765.
[48] Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra. 2018. Deep Learning for Entity Matching: A Design Space Exploration. In *SIGMOD*. 19–34.
[49] Kevin O'Hare, Anna Jurek, and Cassio de Campos. 2018. A new technique of selecting an optimal blocking method for better record linkage. *Inf. Syst.* 77 (2018), 151–166.
[50] George Papadakis, George Alexiou, George Papastefanatos, and Georgia Koutrika. 2015. Schema-agnostic vs Schema-based Configurations for Blocking Methods on Homogeneous Data. *PVLDB* 9, 4 (2015), 312–323.
[51] George Papadakis, Ekaterini Ioannou, Claudia Niederée, Themis Palpanas, and Wolfgang Nejdl. 2012. Beyond 100 million entities: large-scale blocking-based resolution for heterogeneous data. In *WSDM*. 53–62.
[52] George Papadakis, Ekaterini Ioannou, Themis Palpanas, Claudia Niederée, and Wolfgang Nejdl. 2013. A Blocking Framework for Entity Resolution in Highly Heterogeneous Information Spaces. *TKDE* 25, 12 (2013), 2665–2682.
[53] George Papadakis, Georgia Koutrika, Themis Palpanas, and Wolfgang Nejdl. 2014. Meta-Blocking: Taking Entity Resolutionto the Next Level. *TKDE* 26, 8 (2014), 1946–1960.
[54] George Papadakis and Wolfgang Nejdl. 2011. Efficient entity resolution methods for heterogeneous information spaces. In *ICDE Workshops*. 304–307.
[55] George Papadakis, George Papastefanatos, and Georgia Koutrika. 2014. Supervised Meta-blocking. *PVLDB* 7, 14 (2014), 1929–1940.
[56] George Papadakis, Jonathan Svirsky, Avigdor Gal, and Themis Palpanas. 2016. Comparative Analysis of Approximate Blocking Techniques for Entity Resolution. *PVLDB* 9, 9 (2016), 684–695.
[57] George Papadakis, Leonidas Tsekouras, Emmanouil Thanos, George Giannakopoulos, Themis Palpanas, and Manolis Koubarakis. 2018. The return of JedAI: End-to-End Entity Resolution for Structured and Semi-Structured Data. *PVLDB* 11, 12 (2018), 1950–1953.
[58] Thorsten Papenbrock, Arvid Heise, and Felix Naumann. 2015. Progressive Duplicate Detection. *TKDE* 27, 5 (2015), 1316–1329.
[59] Vibhor Rastogi, Nilesh N. Dalvi, and Minos N. Garofalakis. 2011. Large-Scale Collective Entity Matching. *PVLDB* 4, 4 (2011), 208–218.
[60] Orion Fausto Reyes-Galaviz, Witold Pedrycz, Ziyue He, and Nick J. Pizzi. 2017. A supervised gradient-based learning algorithm for optimized entity resolution. *DKE* (2017).
[61] Anish Das Sarma, Ankur Jain, Ashwin Machanavajjhala, and Philip Bohannon. 2012. An automatic blocking mechanism for large-scale de-duplication tasks. In *CIKM*. 1055–1064.
[62] Giovanni Simonini, Sonia Bergamaschi, and H. V. Jagadish. 2016. BLAST: a Loosely Schema-aware Meta-blocking Approach for Entity Resolution. *PVLDB* 9, 12 (2016), 1173–1184.
[63] Giovanni Simonini, George Papadakis, Themis Palpanas, and Sonia Bergamaschi. 2019. Schema-Agnostic Progressive Entity Resolution. *IEEE Trans. Knowl. Data Eng.* 31, 6, 1208–1221.
[64] Yannis Sismanis, Ling Wang, Ariel Fuxman, Peter J. Haas, and Berthold Reinwald. 2009. Resolution-Aware Query Answering for Business Intelligence. In *ICDE*. 976–987.
[65] Kostas Stefanidis, Vasilis Efthymiou, Melanie Herschel, and Vassilis Christophides. 2014. Entity resolution in the web of data. In *WWW*.
[66] Fabian M. Suchanek, Serge Abiteboul, and Pierre Senellart. 2011. PARIS: Probabilistic Alignment of Relations, Instances, and Schema. *PVLDB* 5, 3 (2011), 157–168.
[67] Vasilis Verroios and Hector Garcia-Molina. 2015. Entity Resolution with crowd errors. In *ICDE*. 219–230.
[68] Vasilis Verroios, Hector Garcia-Molina, and Yannis Papakonstantinou. 2017. Waldo: An Adaptive Human Interface for Crowd Entity Resolution. In *SIGMOD*. 1133–1148.
[69] Norases Vesdapunt, Kedar Bellare, and Nilesh N. Dalvi. 2014. Crowdsourcing Algorithms for Entity Resolution. *PVLDB* 7, 12 (2014), 1071–1082.
[70] Jiannan Wang, Tim Kraska, Michael J. Franklin, and Jianhua Feng. 2012. CrowdER: Crowdsourcing Entity Resolution. *PVLDB* 5, 11 (2012), 1483–1494.
[71] Jiannan Wang, Sanjay Krishnan, Michael J. Franklin, Ken Goldberg, Tim Kraska, and Tova Milo. 2014. A sample-and-clean framework for fast and accurate query processing on dirty data. In *SIGMOD*. 469–480.
[72] Jiannan Wang, Guoliang Li, Tim Kraska, Michael J. Franklin, and Jianhua Feng. 2013. Leveraging transitive relations for crowdsourced joins. In *SIGMOD*. 229–240.
[73] Sibo Wang, Xiaokui Xiao, and Chun-Hee Lee. 2015. Crowd-Based Deduplication: An Adaptive Approach. In *SIGMOD*. 1263–1277.
[74] Steven Euijong Whang and Hector Garcia-Molina. 2014. Incremental entity resolution on rules and data. *VLDB J.* 23, 1 (2014), 77–102.
[75] Steven Euijong Whang, Peter Lofgren, and Hector Garcia-Molina. 2013. Question Selection for Crowd Entity Resolution. *PVLDB* 6, 6 (2013), 349–360.
[76] Steven Euijong Whang, David Marmaros, and Hector Garcia-Molina. 2013. Pay-As-You-Go Entity Resolution. *TKDE* 25, 5 (2013), 1111–1124.
[77] Wei Yan, Yuan Xue, and Bradley Malin. 2013. Scalable load balancing for mapreduce-based record linkage. In *IPCCC*. 1–10.
[78] Chen Jason Zhang, Rui Meng, Lei Chen, and Feida Zhu. 2015. CrowdLink: An Error-Tolerant Model for Linking Complex Records. In *ExploreDB*. 15–20.