Task-Driven Big Data Integration

Luca Zecchini

Università degli Studi di Modena e Reggio Emilia, Modena, Italy

Abstract

Data integration aims at combining data acquired from different autonomous sources to provide the user with a unified view of this data. One of the main challenges in data integration processes is entity resolution, whose goal is to detect the different representations of the same real-world entity across the sources, in order to produce a unique and consistent representation for it. The advent of big data has challenged traditional data integration paradigms, making the offline batch approach to entity resolution no longer suitable for several scenarios (e.g., when performing data exploration or dealing with datasets that change with a high frequency). Therefore, it becomes of primary importance to produce new solutions capable of operating effectively in such situations.

In this paper, I present some contributions made during the first half of my PhD program, mainly focusing on the design of a framework to perform entity resolution in an on-demand fashion, building on the results achieved by the progressive and query-driven approaches to this task. Moreover, I also briefly describe two projects in which I took part as a member of my research group, touching on some real-world applications of big data integration techniques, to conclude with some ideas on the future directions of my research.

Keywords

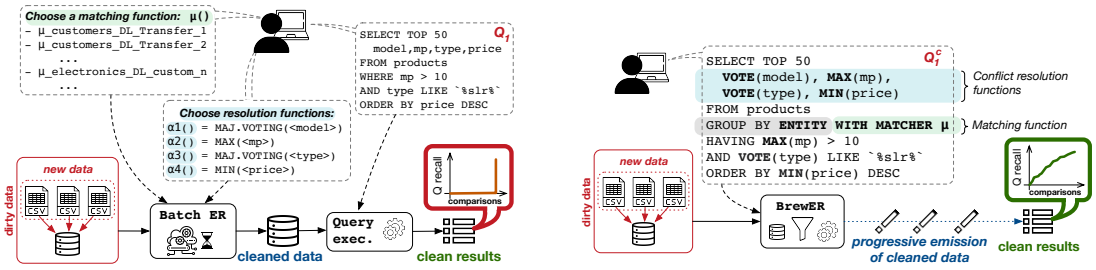
Big Data Integration, Entity Resolution, Data Preparation

1. Introduction

Data integration aims at combining data acquired from different autonomous sources to provide the user with a unified view of this data. The integration of different heterogeneous data sources inherently poses several challenges: the sources can be organized according to different schemas, they can contain duplicates, and so on. The advent of big data and the need to deal with its features (the famous four Vs: volume, velocity, veracity, and variety) required a significant evolution of data integration approaches and techniques, leading to the rise of new paradigms designed to address this scenario.

Among the many challenges that data integration has to overcome, a central role is played by Entity Resolution (ER), a.k.a. record linkage or deduplication, whose goal is to detect the different representations of the same real-world entity across the sources (or even within the same source), then to produce a unique and consistent representation for it. Performing ER can be extremely challenging, due to the inconsistency of the different representations (e.g., compliance with different conventions, presence of missing and wrong values, etc.) and the need to make the computational cost required to perform the comparisons between the pairs of

SEBD 2022: The 30th Italian Symposium on Advanced Database Systems, June 19-22, 2022, Tirrenia (PI), Italy
luca.zecchini@unimore.it (L. Zecchini)
0000-0002-4856-0838 (L. Zecchini)



(a) Batch ER pipeline.

(b) BrewER pipeline.

Figure 1: ER pipelines compared: batch ER vs BrewER.

records affordable (e.g., using a blocking function). Therefore, many algorithms and techniques have been designed to address this problem [1, 2].

ER constitutes an essential step in the data preparation and cleaning pipeline [3], which is required to ensure high data quality. An effective example for understanding the relevance of data quality is that of data-driven decision making, which guides business decisions. Performing data analysis on poor quality data would lead to potentially wrong results (*garbage in, garbage out*), causing a negative economic impact for the business. The importance of data quality is also increasingly highlighted in the field of artificial intelligence, where it is fundamental for the effective training of machine learning models. In fact, even state-of-the-art models may fail to perform well if the data used for their training is not properly prepared [4]. Therefore, increasing relevance is given to the idea of moving from a *model-centric* to a *data-centric* approach[1].

In this paper, I present some contributions made during the first half of my PhD program. In particular, in Section 2.1, I focus on BrewER, a framework to perform ER in an on-demand fashion, which represents my main research contribution so far. Then, in Sections 2.2-2.3, I describe two projects carried out by my research group (DBGroup[2]), which gave me the opportunity to touch some real-world applications of big data integration and data preparation techniques. Finally, in Section 3, I present some ideas on the future directions of my research.

2. Contributions

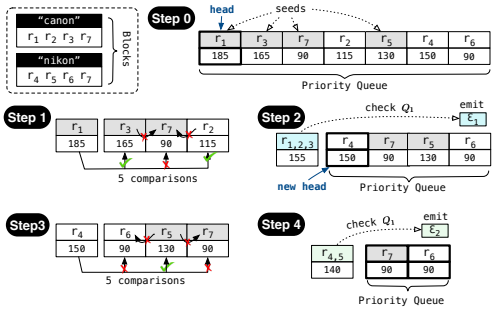
2.1. BrewER: Entity Resolution On-Demand

The well-established end-to-end *batch* approach to ER in big data scenario, mainly consisting of blocking, matching, and fusion phases, requires to perform ER on the whole dataset in order to run queries on the obtained clean version (Figure 1a). This is intuitively far from being an efficient solution in several scenarios. For example, it is the case for data exploration, where the data scientist is usually only interested in a certain portion of the dataset, and this interest can be expressed through a query. In fact, the batch approach implies a lot of useless comparisons, required to produce entities that are guaranteed not to appear in the result of the query. This means wasting time, computational resources, and even money (e.g., pay-as-you-go cloud

[1] https://datacentricai.org
[2] https://dbgroup.unimore.it

	id	brand	model	type	mp	price
ε_1	r_1	canon	eos 400d	dslr	10.1	185.00
	r_2	eos canon	rebel xti	reflex	1.01	115.00
	r_3	canon	eos 400d	dslr	10.1	165.00
ε_2	r_4	nikon	d-200	–	–	150.00
	r_5	nikon	d200	dslr	10.2	130.00
ε_3	r_6	nikon	coolpix	compct	8.0	90.00
ε_4	r_7	canon nikon olympus	olypus-1	dslr	–	90.00

(a) Dirty camera dataset.



(b) BrewER iterating on the priority queue.

Figure 2: BrewER in action.

computing). Such a situation can become critical when data changes with a high frequency (e.g., when dealing with web data or in scenarios such as stock market trading) and there is only a limited amount of time to detect the most relevant entities in the dataset according to the interest expressed by the data scientist.

Progressive approaches presented in literature [5, 6, 7] aim at maximizing the number of matches detected in a certain amount of time. However, these algorithms are guided by the matching likelihood and do not allow the data scientist to define a priority on the entities. Furthermore, in case of early stopping they would produce an approximate result, since they proceed by considering the single candidate pairs of records and not the complete resolution of the entities. On the other hand, the proposed query-driven solutions [8, 9] aim at cleaning only the portion of dataset effectively useful to answer the query, but they still operate in a batch manner, not supporting the progressive emission of the results.

BrewER (see [10] for a detailed explanation, [11] for a brief description of the intuition) is designed to perform ER in an *on-demand* fashion, guided by the query expressing the interest of the data scientist. BrewER is *query-driven*, since it performs ER only on the portion of the dataset that might be useful for answering the query, according to its WHERE clauses (interpreted in BrewER syntax as HAVING clauses, applied after performing a GROUP BY ENTITY step, as depicted in Figure 1b), and *progressive*, since it returns the resulting entities (i.e., completely resolved representative records) as soon as they are obtained, following the priority expressed by the data scientist through the ORDER BY clause. To achieve this result, BrewER performs a preliminary filtering of the blocks, keeping only the ones containing records (called *seeds*) whose values might lead to the generation of an entity included in the result. Then, the records appearing in the survived blocks are inserted in a priority queue, keeping for each one the list of its candidate matches. The priority is defined according to the value of the attribute appearing in the ORDER BY clause, in ascending or descending order. BrewER iterates on the priority queue, considering at each iteration the head element: if it is a record, its candidate matches are checked generating a completely resolved entity; otherwise (i.e., it is a completely resolved entity), it is emitted or discarded based on whether or not it satisfies the query.

BrewER is agnostic towards the choice of the matching and blocking functions, and its capability to perform clean queries on dirty data (as SQL select-project queries, with the ORDER BY clause defining the emission priority) makes it a novel and powerful approach to ER.

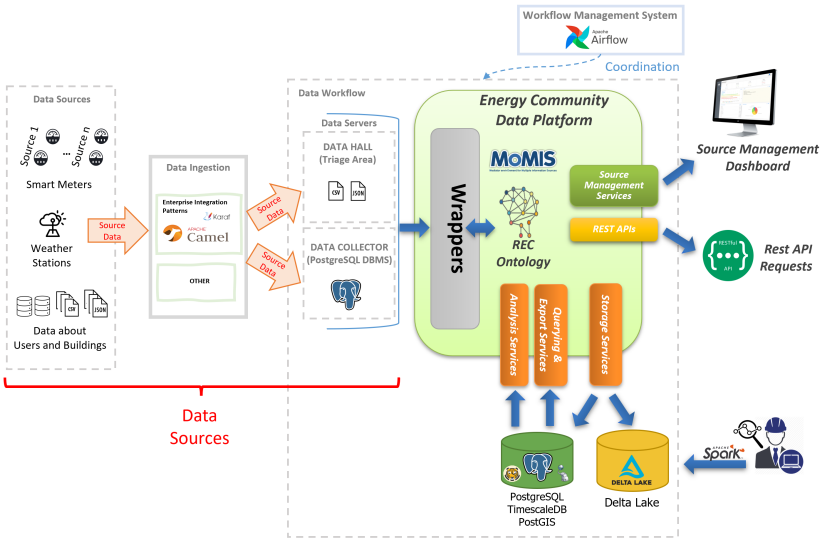


Figure 3: ECDP architecture.

2.2. ECDP: Energy Community Data Platform

ECDP (Energy Community Data Platform) [12] is a middleware platform designed to support the collection and the analysis of big data about the energy consumption inside local energy communities. Its goal is to promote a conscious use of energy by the users, paying particular attention to self-consumption. The project saw the collaboration of the DBGroup and DataRiver, respectively to design and to implement the platform, with the supervision of ENEA.

This project represents a concrete big data integration challenge, since the platform has to acquire data of different nature (from the relational data about the users and the buildings to the sensor data about the energy consumption/production or the weather conditions) from multiple sources. The modular architecture of ECDP, depicted in Figure 3, was designed to promote flexibility and scalability, meeting the needs of different types of users. In particular, ECDP supports: *(i)* a *data integration workflow*, which exploits MOMIS [13, 14] for data integration and a PostgreSQL RDBMS (optimized for time series using TimescaleDB and PostGIS) to store the integrated and aggregated data, ready to be accessed by the standard users; *(ii)* a *data lake workflow*, which relies on Delta Lake to store all raw data, allowing the advanced users to perform some further analysis on it using Apache Spark.

2.3. DXP: Digital Experience Platform

DXP (Digital Experience Platform), subject of an ongoing project commissioned to the DBGroup by Doxee, manages and processes billing data with the goal of providing services to the users (e.g., interactive billing) and analytics to the companies (e.g., churn prediction or user segmentation). In the first part of this projects, aimed at performing data analysis on billing data, we had to design a data preparation pipeline to get this data ready for the analysis. This gave me the opportunity to deal with data preparation challenges in a real-world context, understanding its fundamental impact on data analysis tasks.

3. Future Directions

The topics covered during the first half of my PhD program offer many open challenges and ideas on the future directions of my research, moving from (but not limited to) the path traced by BrewER. In fact, BrewER itself presents many development possibilities. An evolution based on the blocking graph, moving the priority from the record level to the block level, would make it possible to support even unbounded aggregation functions (e.g., SUM); moreover, a specific optimization for meta-blocking[3] [15, 16, 17] would allow to consider in the algorithm also the matching likelihood, further reducing the number of comparisons to be performed. Finally, supporting join would represent a significant improvement, considering also the challenges that need to be faced to adapt it to such a context.

Expanding the view from ER to the whole data preparation and cleaning pipeline, it is easy to notice the presence of many different tasks [3]. Several tasks present specifically designed solutions in the literature, but a lot of work still needs to be done towards an effectively usable holistic tool. Data preparation presents many open challenges, and it is a relevant topic that I aim to investigate, considering multi-task approaches [18] and possibly extending to other tasks the on-demand approach that inspired BrewER.

Finally, operating with personal data (it is the case for medical data, but also for DXP) raises privacy issues. Considering the impact on ER tasks, this implies further challenges to overcome and calls into question *privacy-preserving record linkage* techniques [19]. The future developments of DXP and new dedicated projects will allow me to delve into this topic, adapting the existing solutions designed by the DBGroup to address these challenges and hopefully adding new contributions to the research on this important aspect of big data management.

Acknowledgments

I wish to thank my tutor Prof. Sonia Bergamaschi and my co-tutor Giovanni Simonini, Prof. Felix Naumann, and all the members of the DBGroup.

References

[1] V. Christophides, V. Efthymiou, T. Palpanas, G. Papadakis, K. Stefanidis, An Overview of End-to-End Entity Resolution for Big Data, ACM Computing Surveys (CSUR) 53 (2021) 127:1–127:42.

[2] G. Papadakis, G. Mandilaras, L. Gagliardelli, et al., Three-dimensional Entity Resolution with JedAI, Information Systems (IS) 93 (2020) 101565.

[3] M. Hameed, F. Naumann, Data Preparation: A Survey of Commercial Tools, ACM SIGMOD Record 49 (2020) 18–29.

[4] L. Zecchini, G. Simonini, S. Bergamaschi, Entity Resolution on Camera Records without Machine Learning, in: International Workshop on Challenges and Experiences from Data Integration to Knowledge Graphs (DI2KG), volume 2726 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020.

[3] https://github.com/Gaglia88/sparker

[5] S. E. Whang, D. Marmaros, H. Garcia-Molina, Pay-As-You-Go Entity Resolution, IEEE Transactions on Knowledge and Data Engineering (TKDE) 25 (2013) 1111–1124.
[6] T. Papenbrock, A. Heise, F. Naumann, Progressive Duplicate Detection, IEEE Transactions on Knowledge and Data Engineering (TKDE) 27 (2015) 1316–1329.
[7] G. Simonini, G. Papadakis, T. Palpanas, S. Bergamaschi, Schema-agnostic Progressive Entity Resolution, in: IEEE International Conference on Data Engineering (ICDE), IEEE Computer Society, 2018, pp. 53–64.
[8] H. Altwaijry, D. V. Kalashnikov, S. Mehrotra, Query-Driven Approach to Entity Resolution, Proceedings of the VLDB Endowment (PVLDB) 6 (2013) 1846–1857.
[9] H. Altwaijry, S. Mehrotra, D. V. Kalashnikov, QuERy: A Framework for Integrating Entity Resolution with Query Processing, Proceedings of the VLDB Endowment (PVLDB) 9 (2015) 120–131.
[10] G. Simonini, L. Zecchini, S. Bergamaschi, F. Naumann, Entity Resolution On-Demand, Proceedings of the VLDB Endowment (PVLDB) 15 (2022) 1506–1518.
[11] L. Zecchini, Progressive Query-Driven Entity Resolution, in: International Conference on Similarity Search and Applications (SISAP), volume 13058 of *Lecture Notes in Computer Science (LNCS)*, Springer, 2021, pp. 395–401.
[12] L. Gagliardelli, L. Zecchini, D. Beneventano, et al., ECDP: A Big Data Platform for the Smart Monitoring of Local Energy Communities, in: International Workshop on Data Platform Design, Management, and Optimization (DataPlat), volume 3135 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022.
[13] S. Bergamaschi, S. Castano, M. Vincini, Semantic Integration of Semistructured and Structured Data Sources, ACM SIGMOD Record 28 (1999) 54–59.
[14] S. Bergamaschi, D. Beneventano, F. Mandreoli, et al., From Data Integration to Big Data Integration, in: A Comprehensive Guide Through the Italian Database Research Over the Last 25 Years, volume 31 of *Studies in Big Data*, Springer, 2018, pp. 43–59.
[15] G. Simonini, S. Bergamaschi, H. V. Jagadish, BLAST: a Loosely Schema-aware Meta-blocking Approach for Entity Resolution, Proceedings of the VLDB Endowment (PVLDB) 9 (2016) 1173–1184.
[16] L. Gagliardelli, G. Simonini, D. Beneventano, S. Bergamaschi, SparkER: Scaling Entity Resolution in Spark, in: International Conference on Extending Database Technology (EDBT), OpenProceedings.org, 2019, pp. 602–605.
[17] L. Gagliardelli, S. Zhu, G. Simonini, S. Bergamaschi, BigDedup: A Big Data Integration Toolkit for Duplicate Detection in Industrial Scenarios, in: ISTE International Conference on Transdisciplinary Engineering (TE), volume 7 of *Advances in Transdisciplinary Engineering (ATDE)*, IOS Press, 2018, pp. 1015–1023.
[18] G. Simonini, H. Saccani, L. Gagliardelli, L. Zecchini, D. Beneventano, S. Bergamaschi, The Case for Multi-task Active Learning Entity Resolution, in: Italian Symposium on Advanced Database Systems (SEBD), volume 2994 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021, pp. 363–370.
[19] D. Vatsalan, Z. Sehili, P. Christen, E. Rahm, Privacy-Preserving Record Linkage for Big Data: Current Approaches and Research Challenges, in: Handbook of Big Data Technologies, Springer, 2017, pp. 851–895.