On Combining Collective Entity Resolution and Repairing (Extended Abstract)

Meghyn Bienvenu[1], Gianluca Cima[2] and Víctor Gutiérrez-Basulto[3]

[1]Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR 5800
[2]Department of Computer, Control and Management Engineering, Sapienza University of Rome
[3]School of Computer Science & Informatics, Cardiff University

Abstract
This work summarizes the salient aspects of our recent work [1], about combining collective entity resolution and repairing.

Keywords
Data Quality, Declarative Framework, Logical Rules and Constraints, Entity Resolution, Database Repairing

Data quality (DQ) is one of the most fundamental problems in data management, encompassing several issues such as entity resolution (ER), consistency, completeness, currency, etc. The different facets of DQ have mostly been considered in isolation, giving rise to increasingly sophisticated methods over the years. However, datasets can be expected to suffer from multiple DQ issues.

In our work, we propose a novel declarative framework for jointly tackling the ER and the consistency issues. The ER task is the problem of identifying/matching/merging pairs of syntactically different entity references (constants occurring in a database) that are actually denoting the same real-world entity [2]. We consider so-called collective ER [3], in which we consider multiple tables and/or entity types together, e.g. a merge of a pair of authors may trigger a subsequent merge of a pair of papers. As regards data consistency, we assume that the consistency requirements are specified by means of declarative constraints, and we consider the problem of restoring consistency through the removal of conflicting database facts, as in classical database repairing [4].

The idea of combining ER and repairing has been pioneered in [5], with the goal of generating a single repair of optimal cost. On the contrary, to the best of our knowledge, ours is the first work to explore the computational properties of reasoning over a space of alternative solutions for the combined task, analogously to how consistent query answering reasons over alternative repairs [6].

Our framework, called Replace, builds upon the recently proposed Lace framework [7], employing hard and soft rules to define mandatory and possible merges of constants and adopting the well-known class of *denial constraints* [8], which generalize the conditional FDs considered in [5], to express consistency requirements. A *hard rule* takes the form $q(x, y) \Rightarrow \mathsf{EqO}(x, y)$, where $q(x, y)$ is a *conjunctive query (CQ)* composed by standard relational atoms and atoms using similarity predicates ($\approx$), and EqO is a special symbol used to store merges. Intuitively, such a rule states that (c_1, c_2) being an answer to q provides sufficient conditions for concluding that c_1 and c_2 refer to the same entity. *Soft rules* have a similar form $q(x, y) \dashrightarrow \mathsf{EqO}(x, y)$, but state instead that (c_1, c_2) being an answer to q provides reasonable evidence for c_1 and c_2 denoting the same entity.

Definition 1. *A* DQ specification *takes the form* $\Sigma = \langle \Gamma, \Delta \rangle$, *where* $\Gamma = \Gamma_h \cup \Gamma_s$ *is a finite set of hard and soft rules, and* Δ *is a finite set of denial constraints.*

The semantics of Replace is based on the notion of *(optimal) solutions* to database-specification pairs (D, Σ). More specifically, a solution for a pair (D, Σ) takes the form of a pair $W = (R, E)$, where R is a subset of D and E is an equivalence relation over the constants appearing in the database $D' = D \setminus R$. Intuitively, R indicates the facts to remove from D while E expresses the constants to merge, i.e. all constants from the same equivalence class are deemed to be references to the same entity.

Formally, a pair $W = (R, E)$ is a solution to a database-specification pair (D, Σ) if $R \subseteq D$ and E is an ER solution to $(D \setminus R, \Sigma)$ in the sense of [7]. We recall that in the latter work ER solutions are build 'dynamically', which means that (soft and hard) rule bodies are evaluated on induced databases resulting from applying the already 'derived' merges.

Example 1. *Consider Figure 1. First note that* $D_{\mathsf{ex}} \not\models \delta_1$ *as both* a_3 *and* a_4 *are chairs of KR-12. Notice, however, that* a_3 *and* a_4 *can be merged due to* σ_1*, which resolves the inconsistency. So we obtain a first solution* $W_1 = (R_1, E_1)$,

ENIGMA-23, September 03–04, 2023, Rhodes, Greece
meghyn.bienvenu@labri.fr (M. Bienvenu); cima@diag.uniroma1.it (G. Cima); gutierrezbasultov@cardiff.ac.uk (V. Gutiérrez-Basulto)
0000-0001-6229-8103 (M. Bienvenu); 0000-0003-1783-5605 (G. Cima); 0000-0002-9421-8566 (V. Gutiérrez-Basulto)

Author(aid, email, inst)

aid	email	inst
a_1	wtaka@gm.com	Tokyo
a_2	wtaka@tku.jp	Tokyo
a_3	mnk@ox.uk	NYU
a_4	mnk@gm.com	NYU

Paper(pid, title, fid, year, venue, cid)

pid	title	fid	year	venue	cid
p_1	Computational Complexity of CQA	a_1	2009	IJCAI	a_3
p_2	CQA: Computational Complexity	a_2	2009	IJCAI	a_3
p_3	A Framework for Collective ER	a_1	2010	AAAI	a_2
p_4	Answering CQ over DL Ontologies	a_2	2012	KR	a_3
p_5	AI Techniques for Data Management	a_2	2012	KR	a_4

$\delta_1 = \neg(\exists p, t, f, y, v, c, p', t', f', c'.\, \mathsf{Paper}(p, t, f, y, v, c) \wedge \mathsf{Paper}(p', t', f', y, v, c') \wedge c \neq c')$
$\delta_2 = \neg(\exists p, t, a, y, v.\, \mathsf{Paper}(p, t, a, y, v, a))$
$\rho_1 = \mathsf{Paper}(x, t, f, y, v, c) \wedge \mathsf{Paper}(y, t', f, y, v, c) \wedge t \approx t' \Rightarrow \mathsf{EqO}(x, y)$
$\sigma_1 = \mathsf{Author}(x, e, i) \wedge \mathsf{Author}(y, e', i) \wedge e \approx e' \dashrightarrow \mathsf{EqO}(x, y)$

Figure 1: Database D_{ex} and DQ specification $\Sigma_{\mathsf{ex}} = \langle \Gamma_{\mathsf{ex}}, \Delta_{\mathsf{ex}} \rangle$ with $\Gamma_{\mathsf{ex}} = \{\rho_1, \sigma_1\}$ and $\Delta_{\mathsf{ex}} = \{\delta_1, \delta_2\}$. The extension of the similarity predicate $\approx$ (restricted to the constants in D_{ex}) is the symmetric and reflexive closure of $\{(e_1, e_2), (e_3, e_4), (t_1, t_2)\}$, where e_i and t_i are the email of author a_i and title of paper p_i, respectively. The attributes fid and cid indicate, respectively, the first author of the paper and the chair of the conference (pair (venue,year)), respectively.

where $R_1 = \emptyset$ and E_1 is the equivalence relation induced by $\{(a_3, a_4)\}$. Constants a_1 and a_2 can also be merged due to σ_1. However, if we merge them, then the resulting database is such that: (i) *δ_2 is violated because the first author of paper p_3 is the chair of the conference where p_3 was published, and* (ii) *p_1 and p_2 must be merged due to ρ_1. So we have a second solution $W_2 = (R_2, E_2)$, where R_2 contains the tuples with pid p_3 and E_2 is the equivalence relation induced by $\{(a_1, a_2), (a_3, a_4), (p_1, p_2)\}$.*

Among all the solutions, it is natural to focus only on the 'best' ones, i.e. those maximizing the merges performed and minimizing the facts removed. These two criteria may conflict, as deleting more facts may enable more merges. We thus consider three natural ways to compare solutions: give priority to the maximization of merges (Mer), give priority to the minimization of deletions (Del), or adopt the Pareto principle and accord equal priority to both criteria (Par). Specifically, the preorders $\prec_{\textsc{Mer}}$, $\prec_{\textsc{Del}}$, and $\prec_{\textsc{Par}}$ are defined as follows:

- $(R, E) \prec_{\textsc{Mer}} (R', E')$ iff either (*i*) $E \subset E'$ or (*ii*) $E \subseteq E'$ and $R' \subset R$;
- $(R, E) \prec_{\textsc{Del}} (R', E')$ iff either (*i*) $R' \subset R$ or (*ii*) $R' \subseteq R$ and $E \subset E'$;
- $(R, E) \prec_{\textsc{Par}} (R', E')$ iff either (*i*) $E \subset E'$ and $R' \subseteq R$ or (*ii*) $R' \subset R$ and $E \subseteq E'$.

For $X \in \{\textsc{Mer}, \textsc{Del}, \textsc{Par}\}$, a solution W for (D, Σ) is an *$\preceq_X$-optimal solution for (D, Σ)* if there is no solution W' for (D, Σ) such that $W \prec_X W'$, and denote by $\mathsf{Sol}_X(D, \Sigma)$ the set of $\preceq_X$-optimal solutions for (D, Σ).

Example 2. *Recall Example 1. We have that $\mathsf{Sol}_{\textsc{Mer}} = \{W_2\}$, $\mathsf{Sol}_{\textsc{Del}} = \{W_1\}$, and $\mathsf{Sol}_{\textsc{Par}} = \{W_1, W_2\}$.*

As there may be many optimal solutions, we adopt the notions of possible and certain query answers to reason about alternative solutions. For $X \in \{\textsc{Mer}, \textsc{Del}, \textsc{Par}\}$, we say that a tuple $\vec{c}$ is an *X-certain answer* (resp. *X-possible answer*) to a query q w.r.t. (D, Σ) if $\vec{c}$ is an answer to q in every (resp. some) X-optimal solution. We use X-$\mathsf{certAns}(q, D, \Sigma)$ and X-$\mathsf{possAns}(q, D, \Sigma)$ for the sets of X-certain and X-brave answers. We further introduce the novel notions of *most informative* possible and certain answers (X-$\mathsf{MIpossAns}(q, D, \Sigma)$ and X-$\mathsf{MIcertAns}(q, D, \Sigma)$), which take the form of tuples of *sets of constants*. Most informative answers offer a more compact presentation of query results, avoiding the output of distinct but equivalent tuples. In our running example, this would mean returning $(\{a_3, a_4\})$ rather than both (a_3) and (a_4) when querying for chair of KR-12. We refer readers to the full paper for formal definitions.

Aside from introducing the new framework, we outlined the precise data complexity of the following tasks:

- X-MaxRec: decide whether $W \in \mathsf{Sol}_X(D, \Sigma)$;
- X-CertAns (resp. X-PossAns): decide whether $\vec{c} \in X$-$\mathsf{certAns}(q, D, \Sigma)$ (resp. $\vec{c} \in X$-$\mathsf{possAns}(q, D, \Sigma)$);
- X-MIcertAns (resp. X-MIpossAns): decide whether a tuple of sets of constants $\vec{C}$ is such that $\vec{C} \in X$-$\mathsf{MIcertAns}(q, D, \Sigma)$ (resp. $\vec{C} \in X$-$\mathsf{MIpossAns}(q, D, \Sigma)$).

In future work, we plan to develop a prototype implementation of Replace based on logic-based technologies, such as answer set programming (ASP). Most informative certain answers will require special treatment, due to their DP_2 complexity, which goes beyond what is supported by ASP. It would also be relevant to integrate similarity measures defined via machine learning predicates, in the style of [9], and to allow for both *global merges* (the ones considered here) and *local merges* (suitable when merging values rather than references), as has been recently considered in [10, 11].

Acknowledgments

This work has been supported by the ANR AI Chair INTENDED (ANR-19-CHIA-0014), by MUR under the PNRR project FAIR (PE0000013), and by the Royal Society (IES\R3\193236).

References

[1] M. Bienvenu, G. Cima, V. Gutiérrez-Basulto, REPLACE: A logical framework for combining collective entity resolution and repairing, in: Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI 2023), 2023.

[2] P. Singla, P. M. Domingos, Entity resolution with markov logic, in: Proceedings of the Sixth IEEE International Conference on Data Mining (ICDM 2006), 2006, pp. 572–582.

[3] I. Bhattacharya, L. Getoor, Collective entity resolution in relational data, ACM Transactions on Knowledge Discovery from Data 1 (2007) 5.

[4] J. Chomicki, J. Marcinkowski, Minimal-change integrity maintenance using tuple deletions, Information and Computation 197 (2005) 90–121.

[5] W. Fan, S. Ma, N. Tang, W. Yu, Interaction between record matching and data repairing, Journal of Data and Information Quality 4 (2014) 16:1–16:38.

[6] M. Arenas, L. E. Bertossi, J. Chomicki, Consistent query answers in inconsistent databases, in: Proceedings of the Eighteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 1999), 1999, pp. 68–79.

[7] M. Bienvenu, G. Cima, V. Gutiérrez-Basulto, LACE: A logical approach to collective entity resolution, in: Proceedings of the Forty-First ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS 2022), 2022, pp. 379–391.

[8] L. E. Bertossi, Database Repairing and Consistent Query Answering, Synthesis Lectures on Data Management, Morgan & Claypool Publishers, 2011.

[9] T. Deng, W. Fan, P. Lu, X. Luo, X. Zhu, W. An, Deep and collective entity resolution in parallel, in: Proceedings of the Thirty-Eighth IEEE International Conference on Data Engineering (ICDE 2022), 2022, pp. 2060–2072.

[10] M. Bienvenu, G. Cima, V. Gutiérrez-Basulto, Y. Ibáñez-García, Combining global and local merges in logic-based entity resolution, in: Proceedings of the Twentieth International Conference on Principles of Knowledge Representation and Reasoning (KR 2023), 2023.

[11] R. Fagin, P. G. Kolaitis, D. Lembo, L. Popa, F. Scafoglieri, A framework for combining entity resolution and query answering in knowledge bases, in: Proceedings of the Twentieth International Conference on Principles of Knowledge Representation and Reasoning (KR 2023), 2023.