On Combining Collective Entity Resolution and Repairing (Extended Abstract)

Meghyn Bienvenu¹, Gianluca Cima² and Victor Gutiérrez-Basulto³

¹Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR 5800
²Department of Computer, Control and Management Engineering, Sapienza University of Rome
³School 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 cost of optimal cost. On the contrary, to the best of our knowledge, ours is the first work to explore the computational cost of optimal cost. On the contrary, to the best of our knowledge, ours is the first work to explore the computational cost of optimal cost.

The semantics of REPLACE is based on the notion of (optimal) solutions to database-specification pairs \((D, \Sigma)\). More specifically, a solution for a pair \((D, \Sigma)\) 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 \((D, \Sigma)\) 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 \(D_{\alpha} \not\models \delta_1\) as both \(\alpha_3\) and \(\alpha_4\) are chairs of KR-12. Notice, however, that \(\alpha_3\) and \(\alpha_4\) can be merged due to \(\sigma_1\), which resolves the inconsistency. So we obtain a first solution \(W_1 = (R_1, E_1)\),

\[\]
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 $\sigma_1$. However, if we merge them, then the resulting database is such that (i) $\delta_2$ is violated because the first author of paper $p_2$ is the chair of the conference where $p_1$ was published, and (ii) $p_1$ and $p_2$ must be merged due to $\rho_2$. So we have a second solution $S = (R_2, E_2)$, where $R_2$ contains the tuples with pid $p_1$ 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 $\prec_{\text{Mer}}$, $\prec_{\text{Del}}$, and $\prec_{\text{Par}}$ are defined as follows:

- $(R, E) \prec_{\text{Mer}} (R', E')$ if either (i) $E \subseteq E'$ or (ii) $E \subseteq E'$ and $R' \subseteq R$;
- $(R, E) \prec_{\text{Del}} (R', E')$ if either (i) $R' \subseteq R$ and $E \subseteq E'$;
- $(R, E) \prec_{\text{Par}} (R', E')$ if either (i) $E \subseteq E'$ and $R' \subseteq R$ or (ii) $R' \subseteq R$ and $E \subseteq E'$.

For $X \in \{\text{Mer, Del, Par}\}$, a solution $W$ for $(D, \Sigma)$ is an $\leq_{X}$-optimal solution for $(D, \Sigma)$ if there is no solution $W'$ for $(D, \Sigma)$ such that $W \prec_{X} W'$ and denote by $\text{Sol}_{X} (D, \Sigma)$ the set of $\leq_{X}$-optimal solutions for $(D, \Sigma)$. 

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

As there may be many optimal solutions, we adopt the notions of possible and certain query answers to reason about alternative solutions. For $X \in \{\text{Mer, Del, Par}\}$, we say that a tuple $\overline{c}$ is an $X$-certain answer (resp. $X$-possible answer) to a query $q$ w.r.t. $(D, \Sigma)$ if $\overline{c}$ is an answer to $q$ in every (resp. some) $X$-optimal solution. We use $X$-$\text{CertAns}(q, \Sigma)$ and $X$-$\text{PossAns}(q, D, \Sigma)$ for the sets of $X$-certain and $X$-brave answers. We further introduce the novel notions of most informative possible and certain answers ($X$-$\text{MPossAns}(q, \Sigma)$ and $X$-$\text{MCertAns}(q, \Sigma)$), 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:

- $X$-$\text{MaxRec}$: decide whether $W \in \text{Sol}_{X} (D, \Sigma)$;
- $X$-$\text{CertAns}$ (resp. $X$-$\text{PossAns}$): decide whether $\overline{c} \in X$-$\text{CertAns}(q, D, \Sigma)$ (resp. $\overline{c} \in X$-$\text{PossAns}(q, D, \Sigma)$);
- $X$-$\text{MCertAns}$ (resp. $X$-$\text{MPossAns}$): decide whether a tuple of sets of constants $\overline{C}$ is such that $\overline{C} \in X$-$\text{MCertAns}(q, D, \Sigma)$ (resp. $\overline{C} \in X$-$\text{MPossAns}(q, D, \Sigma)$).

In future work, we plan to develop a prototype implementation of $\text{REPLACE}$ based on logic-based technologies, such as answer set programming (ASP). Most informative certain answers will require special treatment, due to their $\mathcal{D}$ 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