Entity Resolution and Data Fusion: an integrated approach (DISCUSSION PAPER)

Domenico Beneventano[1], Sonia Bergamaschi[1],
Luca Gagliardelli[1], and Giovanni Simonini[2]

[1] University of Modena and Reggio Emilia, Modena, Italy
name.surname@unimore.it
[2] MIT CSAIL, Cambridge, MA, USA
giovanni@csail.mit.edu

Abstract. *Entity Resolution* and *Data Fusion* are fundamental tasks in a Data Integration process. Unfortunately, these tasks cannot be completely addressed by purely automated methods and, then, a "human-in-the-loop" approach, i.e., the interaction with the Integration Designer has to be considered. In fact, the application goal can be relevant to reduce the complexity and the cost of the whole integration process. Moreover, the Entity Resolution and Data Fusion tasks are often considered consecutive and independent of each other: the output of the first step is used as input of the second one.
In this paper, we will show how these tasks have not to be considered independent. In fact, the evaluation of data fusion results is fundamental for the Integration Designer to analyze, and eventually modify, the choices made during the Entity Resolution process. To show this, our highly scalable Entity Resolution tool, SparkER, will be extended with post-processing high-quality methods for matching. These methods will be integrated in the MOMIS Data Fusion system, extended as well with metrics for the evaluation of data fusion results.

Keywords: Data Integration · Entity Resolution · Data Fusion

1 Introduction

Data Integration is the problem of combining data residing at different autonomous sources, and providing the user with a unified view of these data. MOMIS (Mediator EnvirOnment for Multiple Information Sources) is an open source Data Integration System [3], characterized by a classical wrapper/mediator architecture, where the local data sources contain the real data, while a Global

Virtual Schema (GVS) provides a reconciled, integrated, and virtual view of the underlying data sources. In particular, MOMIS performs *Data Fusion*, i.e., the process of fusing multiple records representing the same real-world object into a single, consistent, and clean representation; to perform data fusion, several *conflict handling strategies* introduced in [7] are available in the MOMIS system (see section 2.2). As described in several papers MOMIS adopts a semi-automatic approach that retains the "human in the loop" where algorithms and tools are used to assist the Integration Designer performing the Data Fusion task [4, 17].

In the current MOMIS version, the Data Fusion process assumes that *Entity Resolution* (ER) has been already performed and thus a shared object identifier (ID) exists among different sources; in other words, the current version of MOMIS only implements an *exact match*. Multiple records with the same ID are fused by means of the Full Join Merge operator [4].

On 2016, we faced and solved the Entity Resolution problem by developing a set of novel techniques [1, 5, 12, 13, 15, 16]. In particular, we proposed `Blast` [13] (Blocking with Loosely-Aware Schema Techniques), an approach to reduce the ER complexity with indexing techniques aiming to group similar records in blocks and limit the comparison to only those records appearing in the same block. This approach was implemented in `SparkER` [9, 14], a highly scalable Entity Resolution tool designed to be parallelizable on Apache Spark. As highlighted in a very recent demo of `SparkER` [10], the Entity Resolution process can be improved by including the "human-in-the-loop".

On the other hand, the Entity Resolution and Data Fusion steps are often considered consecutive and independent of each other: the output of the first step is used as input of the second one in Data Integration systems. In this paper we will show how these tasks have not to be considered independent. In fact, the evaluation of data fusion results is fundamental for the Integration Designer to analyze, and eventually modify, the choices made during the Entity Resolution process.

In detail, the main contributions of this paper are the following:

- `SparkER` will be extended with post-processing methods to obtain one-to-one matching; such extended `SparkER` will be integrated in the MOMIS framework: the output will be used as input in the MOMIS Data Fusion system;
- `MomisDF`, the module of the MOMIS system which performs Data Fusion, will be extended as well with methods for the evaluation of data fusion results;
- We will show, by an example, how the evaluation of data fusion results can be used by the Integration Designer.

The complete `SparkER-MomisDF` workflow is shown in Figure 1. `SparkER` is composed by two main modules: (i) **blocker**: takes the input records and performs the blocking phase, providing as output the candidate pairs; (ii) **Entity matcher** takes the candidate pairs generated by the blocker and label them as match or no match by comparing pair's similarity with a threshold, so producing a `Match Table` of similar records with their similarity score. The **1-1 Matching** module take as input such `Match Table`, which may contain

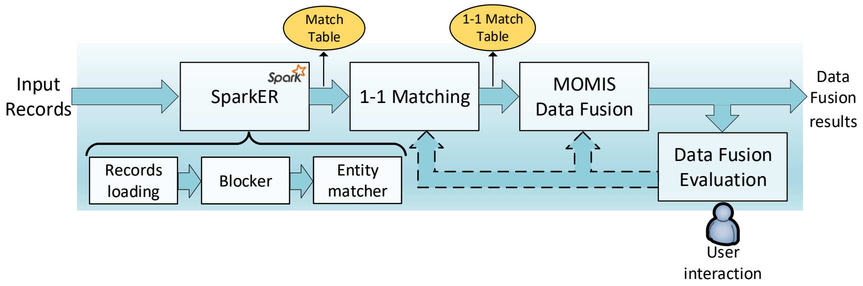


Fig. 1. The SparkER-MomisDF workflow

many-to-many matching, and returns a `1-1 Match Table` with only *one-to-one* matching. `MomisDF` performs data fusion of the input records, on the basis of the `1-1 Match Table`.

The structure of the paper is the following. Section 2 contains some preliminaries from literature which we use in our paper. Section 3 shows, by means of an example covering the whole `SparkER-MomisDF` workflow, how the evaluation of fusion results can be useful for the Integration Designer.

2 Preliminaries

This section contains some preliminaries from literature which we use in our paper. Section 2.1 introduces the post-processing methods for one-to-one matching proposed in [8]; section 2.2 introduces the methods for the evaluation of data fusion results proposed in [7].

2.1 Post-processing Methods for one-to-one Matching

As stated in [8] assuming de-duplicated sources, each record can at most match to one record of another source; hence, the matching result should exclusively contain *one-to-one* links as otherwise precision is deteriorated; in other words, in the most common two-source case, it is often desirable for the final matching to be one-to-one [18]. On the other hand, Most ER methods, and in particular, the ones using threshold-based techniques, as `SparkER`, often produce multi-links, i.e., one record is matched to many records of another source. For this reason, the authors of [8] proposed methods that can be executed after any entity resolution process to clean multi-links, i.e., to transform the result such that only *one-to-one* matching occurs in the final result. The following three post-processing strategies are analyzed and implemented in [8]:

- *Symmetric Best Match* (*Max1-both*): the basic idea is that for every record only the best matching record of the other source is accepted.
- *Maximum Weight matching* (MWM): a MWM is a matching that has maximum weight, i.e., that maximizes the sum of the overall similarities between records in the final linkage result.

- *Stable Marriage* (SM): a matching is defined as stable, if there are no two records of the different local classes who both have a higher similarity to each other than to their current matching record.

In [8] a complete evaluation of the different post-processing methods is performed by using both synthetic and real datasets. The linkage quality is assessed by *recall* and *precision*: recall measures the proportion of true-matches that have been correctly classified as matches after the linkage process; precision is defined as the fraction of classified matches that are true-matches. The aim of post-processing is to optimize precision while recall is ideally preserved. The result of the evaluation performed in [8] was that both *Max1-both* and *SM* are able to significantly improve the linkage quality; in general, *Max1-both* can achieve the best linkage quality in terms of precision; for applications favoring recall over precision, a *SM* should be applied.

2.2 Data Fusion Evaluation

To perform data fusion, several *conflict handling strategies* defined in [7] are available in the MOMIS system; in particular, the following strategies:

S1 *Take the information*: prefers values over null values;
S2 *Consider all possibilities*: creates all possible value combinations.

These strategies are implemented as default, i.e., before the intervention of the Designer, that can also apply some *Resolution Strategies* choosing by the *Conflict Resolution Functions* implemented in `MomisDF`, such as, takes an average value and takes the most recent value. The authors of [7] also propose methods for the evaluation of data fusion based on measures of quality of source data, such as *completeness* and *consistency*; for example, the (extensional) completeness is $\mid$ *unique objects in dataset* $\mid / \mid$ *unique objects in universe* $\mid$.
Instead of quality of data sources, we want to evaluate quality of fused data, then we reformulate such measures in terms of the *Global Class* (GC) resulting from the data fusion process (see next section for an example), by considering the following *Data Centric* Evaluation Measures[3]:

- **Density**: measures the fraction of non-NULL values.
  The density of a *Global Attribute* GA in GC is defined as

$$Density_{GA} = \frac{\mid non\text{-}NULL\ values\ in\ GA \mid}{\mid records\ in\ GC \mid}$$

  The density of the whole global class GC is defined as

$$Density_{CG} = \frac{\mid non\text{-}NULL\ values\ in\ GC \mid}{\mid attributes\ in\ GC \mid * \mid records\ in\ GC \mid}$$

[3] Such measures are also introduced in [6]. In our evaluation, the number of objects in the universe is identified with the number of unique entities in the fused data source, and then with the records of the Global Class GC.

- **Consistency**: a data set is consistent if it is free of conflicting information. The consistency of a *Global Attribute* GA in GC is defined as

$$Consistency_{GA} = \frac{\mid \textit{non-conflicting values in GA} \mid}{\mid \textit{records in GC} \mid}$$

The consistency of the whole global class GC is defined as

$$Consistency_{GC} = \frac{\mid \textit{non-conflicting values in GC} \mid}{\mid \textit{attributes in GC} \mid * \mid \textit{records in GC} \mid}$$

In a similar way, in [11] the concept of *F-quality* is introduced as a measure of quality of fused data rather than source data and two novel algorithms for Linked Data fusion with provenance tracking and quality assessment of fused data are proposed.

Our goal is different, we want to discuss how the evaluation of data fusion results can be used by the Integration Designer to analyze, and eventually modify, the choices made during the Entity Resolution process, to improve the final result. For these reasons, we only consider simple and common conflict handling strategies (i.e. **S1** and **S2**) and the straightforward evaluation defined above.

3 Example and Discussion

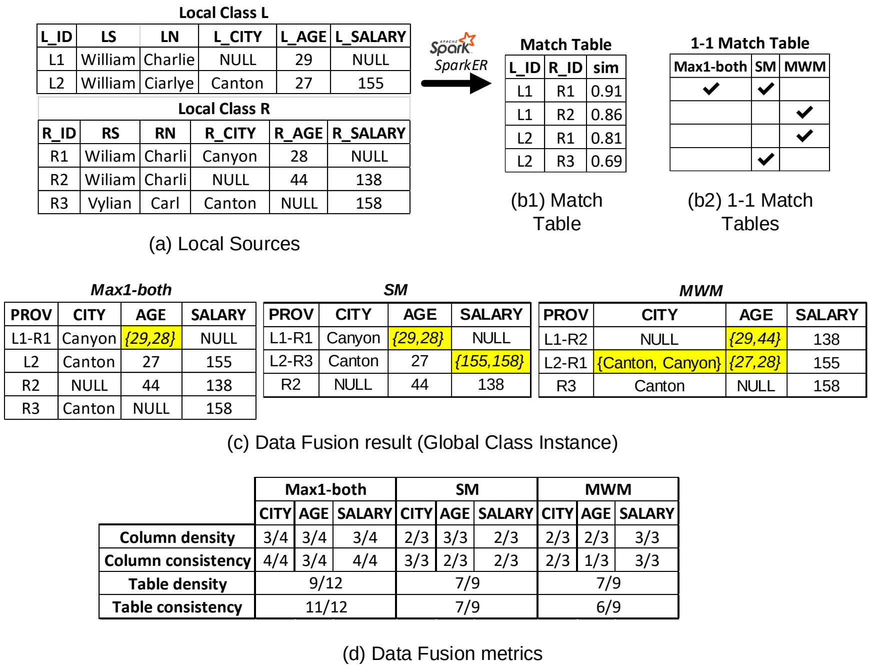
This section shows, by means of an example covering the whole SparkER-MOMIS workflow, how the evaluation of fusion results can be useful for the Integration Designer; in particular, this example will show how the evaluation of data fusion results changes by varying the post-processing 1-1 matching method.

We consider two local classes L and R and a global class GC with the same attributes: Surname (S), Name (N), City, Age and Salary (i.e, we assume that the *schema matching* and *global schema generation* phases have already been carried out with the MOMIS framework); moreover L and R have a local ID. An instance of L and R is shown in Figure 2(a).

Let us first consider the simplest case: all the attributes are used to perform Entity Resolution. The ER process performed with `SparkER` (see [10] for a detailed description of the tool) produces the *Match Table* shown in Figure 2(b). Note that the match table provides the pair of local identifiers of each record and their similarity, and that the obtained matching is *many-to-many* (e.g. L1 matches with R1, R2). Now it is possible to apply the three post-processing methods discussed in section 2.1 to obtain a *one-to-one* matching: each of these methods produces a *1-1 Match Table* as shown in Figure 2(b2) (a ✓denotes that the pair is in the *1-1 Match Table*).

Each *1-1 Match Table* is then used to perform Data Fusion; intuitively, two sequential operations are performed:

1. the natural outer join is performed:
   $\textit{Local Class L} ⟗ \textit{1-1 Match Table} ⟕ \textit{Local Class R}$



Local Class L

L_ID	LS	LN	L_CITY	L_AGE	L_SALARY
L1	William	Charlie	NULL	29	NULL
L2	William	Ciarlye	Canton	27	155

Local Class R

R_ID	RS	RN	R_CITY	R_AGE	R_SALARY
R1	Wiliam	Charli	Canyon	28	NULL
R2	Wiliam	Charli	NULL	44	138
R3	Vylian	Carl	Canton	NULL	158

(a) Local Sources

Spark
SparkER

Match Table

L_ID	R_ID	sim
L1	R1	0.91
L1	R2	0.86
L2	R1	0.81
L2	R3	0.69

(b1) Match Table

1-1 Match Table

Max1-both	SM	MWM
✓	✓	
		✓
		✓
	✓	

(b2) 1-1 Match Tables

Max1-both

PROV	CITY	AGE	SALARY
L1-R1	Canyon	*{29,28}*	NULL
L2	Canton	27	155
R2	NULL	44	138
R3	Canton	NULL	158

SM

PROV	CITY	AGE	SALARY
L1-R1	Canyon	*{29,28}*	NULL
L2-R3	Canton	27	*{155,158}*
R2	NULL	44	138

MWM

PROV	CITY	AGE	SALARY
L1-R2	NULL	*{29,44}*	138
L2-R1	{Canton, Canyon}	*{27,28}*	155
R3	Canton	NULL	158

(c) Data Fusion result (Global Class Instance)

	Max1-both			SM			MWM		
	CITY	AGE	SALARY	CITY	AGE	SALARY	CITY	AGE	SALARY
Column density	3/4	3/4	3/4	2/3	3/3	2/3	2/3	2/3	3/3
Column consistency	4/4	3/4	4/4	3/3	2/3	2/3	2/3	1/3	3/3
Table density	9/12			7/9			7/9		
Table consistency	11/12			7/9			6/9		

(d) Data Fusion metrics

Fig. 2. The SparkER-MomisDF workflow example

2. the *S1* and *S2* strategies (see Section 2.2) are applied to all the conflicting attributes involved in the Data Fusion process.

For example, the *Max1-Both* Match Table contains only the pair $(L1, R1)$ and only such two local records are fused together so obtaining the first record in the *Max1-Both* Global Class shown in Figure 2(c), where the *PROV* attribute represents the *data provenance* [2], i.e., intuitively, the input local records that contributed to the output global record. Conflicting values obtained by the *S2* strategy are highlighted in yellow. With the other *1-1 Match Tables*, different global class are obtained, as shown in Figure 2(c). Finally, the Integration Designer chooses which attributes to use in the evaluation phase; in our example, suppose they are City, Age and Salary. The Data Fusion evaluation, with the measures density and consistency is shown in Figure 2(d).

In this preliminary work, we have not yet achieved results on significant and real cases. However, some considerations can also be made about the (toy) example. First of all, *Max1-both* achieves the best match quality in terms of column and table consistency. This is consistent with the conclusion reached in [8] that, in general, *Max1-both* can achieve the best linkage quality in terms of precision. On the other hand, it is easy to verify that to increase column and table density, *SM* or *MWM* should be applied, with the obvious consequence that these methods deteriorate column and table consistency. For these reasons, we believe it is important to give the designer the opportunity to analyze and choose the best method. For example, in a context where the Salary attribute has greater importance, the best choice is the *MWM* method that maximizes both density and consistency for such attribute.

In this example we only discussed evaluation of data fusion results to varying of the post-processing 1-1 matching method. On the other hand, it will also be interesting to evaluate the results of the data fusion correspond to the changes in the configurations sparker (and therefore in the different Match Tables produced). In fact, as discussed in [10], the `SparkER` tool can work both in a completely unsupervised mode and in a supervised one. In the first case, the Integration Designer can use a default configuration and perform the process on its data without taking care of the parameters tuning. In the second case, she/he can supervise the entire process, in order to determine which are the best parameters for her/his data, thus producing a custom configuration.

4 Conclusions and future work

We discussed some preliminary ideas about an integrated approach for entity resolution and data fusion. We showed, by an example, how the evaluation of data fusion results can be used by the Integration Designer to analyze, and eventually modify, the choices made during the Entity Resolution process.

As future work, we will perform a complete evaluation of data fusion results with respect to the different post-processing methods, both using real datasets and with other evaluation measures. Another future work is to extend the data fusion evaluation to *Conflict Resolution* Functions, by considering *Ground Truth Based Evaluation* measures, such as the Accuracy, in order to evaluate the fraction of correct values selected by conflict resolution functions chosen by the Integration Designer.

References

1. Benedetti, F., Beneventano, D., Bergamaschi, S., Simonini, G.: Computing inter-document similarity with context semantic analysis. Information Systems **80**, 136–147 (2019). https://doi.org/10.1016/j.is.2018.02.009
2. Beneventano, D., Bergamaschi, S.: Provenance-aware semantic search engines based on data integration systems. Inter. J. of Organizational and Collective Intelligence (IJOCI) **4**(2), 1–30 (Apr 2014). https://doi.org/10.4018/ijoci.2014040101
3. Bergamaschi, S., Beneventano, D., Corni, A., Kazazi, E., Orsini, M., Po, L., Sorrentino, S.: The open source release of the MOMIS data integration system. In: Nineteenth Italian Symposium on Advanced Database Systems (SEBD). pp. 175–186 (2011)
4. Bergamaschi, S., Beneventano, D., Guerra, F., Orsini, M.: Data integration. In: Handbook of Conceptual Modeling - Theory, Practice, and Research Challenges, pp. 441–476. Springer (2011)
5. Bergamaschi, S., Ferrari, D., Guerra, F., Simonini, G., Velegrakis, Y.: Providing insight into data source topics. J. Data Semantics **5**(4), 211–228 (2016). https://doi.org/10.1007/s13740-016-0063-6

6. Bizer, C.: Data quality assessment and data fusion. University Lecture (2018)
7. Bleiholder, J., Naumann, F.: Data fusion. ACM Comput. Surv. **41**(1), 1:1–1:41 (Jan 2009). https://doi.org/10.1145/1456650.1456651
8. Franke, M., Sehili, Z., Gladbach, M., Rahm, E.: Post-processing methods for high quality privacy-preserving record linkage. In: Data Privacy Management, Cryptocurrencies and Blockchain Technology - ESORICS 2018 International Workshops, Barcelona, Spain, September 6-7, 2018, Proceedings. pp. 263–278 (2018)
9. Gagliardelli, L., Zhu, S., Simonini, G., Bergamaschi, S.: Bigdedup: a big data integration toolkit for duplicate detection in industrial scenarios. In: Transdisciplinary Engineering. vol. 7, pp. 1015–1023 (2018)
10. Gagliardelli, L., Simonini, G., Beneventano, D., Bergamaschi, S.: Sparker: Scaling entity resolution in spark. In: Proceedings of the Workshops of the EDBT/ICDT 2019 Joint Conference (EDBT/ICDT), Lisbon, Portugal (2019)
11. Michelfeit, J., Knap, T., Necaský, M.: Linked data integration with conflicts. CoRR **abs/1410.7990** (2014), http://arxiv.org/abs/1410.7990
12. Simonini, G., Bergamaschi, S.: Enhancing entity resolution efficiency with loosely schema-aware techniques. In: 24th Italian Symposium on Advanced Database Systems, SEBD 2016, Ugento, Lecce, Italy, June 19-22, 2016, Ugento, Lecce, Italia, June 19-22, 2016. pp. 270–277 (2016)
13. Simonini, G., Bergamaschi, S., Jagadish, H.V.: BLAST: a loosely schema-aware meta-blocking approach for entity resolution. PVLDB **9**(12), 1173–1184 (2016), http://www.vldb.org/pvldb/vol9/p1173-simonini.pdf
14. Simonini, G., Gagliardelli, L., Bergamaschi, S., Jagadish, H.V.: Scaling entity resolution: A loosely schema-aware approach. Inf. Syst. **83**, 145–165 (2019). https://doi.org/10.1016/j.is.2019.03.006
15. Simonini, G., Papadakis, G., Palpanas, T., Bergamaschi, S.: Schema-agnostic progressive entity resolution. In: 34th IEEE International Conference on Data Engineering, ICDE 2018, Paris, France, April 16-19, 2018. pp. 53–64. IEEE Computer Society (2018). https://doi.org/10.1109/ICDE.2018.00015
16. Simonini, G., Papadakis, G., Palpanas, T., Bergamaschi, S.: Schema-agnostic progressive entity resolution. IEEE Trans. Knowl. Data Eng. **31**(6), 1208–1221 (2019). https://doi.org/10.1109/TKDE.2018.2852763
17. Vincini, M., Beneventano, D., Bergamaschi, S.: Semantic integration of heterogeneous data sources in the MOMIS data transformation system. J. UCS **19**(13), 1986–2012 (2013). https://doi.org/10.3217/jucs-019-13-1986
18. Zhang, D., Rubinstein, B.I.P., Gemmell, J.: Principled graph matching algorithms for integrating multiple data sources. IEEE Trans. on Knowl. and Data Eng. **27**(10), 2784–2796 (Oct 2015). https://doi.org/10.1109/TKDE.2015.2426714