Scaling with Confidence: 
Entity Resolution under 
Weighted Constraints

Qing Wang, Research School of Computer Science, The Australian National University, Canberra, Australia
Zeyu Shen, Research School of Computer Science, The Australian National University, Canberra, Australia

ABSTRACT

Constraints ubiquitously exist in many real-life applications for entity resolution. However, it is always challenging to effectively specify and use such constraints for performing ER tasks. In particular, not every constraint is equally robust. Adding weights to express the “confidence” on constraints thus becomes a natural choice. In this paper, the authors study entity resolution (ER), the problem of determining which records in one or more databases refer to the same entities, in the presence of weighted constraints. They propose a unified framework that allows us to associate a weight for each constraint, capturing the confidence for its robustness in an ER model. The authors develop an approach to learn weighted constraints based on domain knowledge, and investigate how effectively and efficiently weighted constraints can be used for generating an ER clustering and for determining a propagation order across multiple entity types. Their experimental study shows that using weighted constraints can lead to improved ER quality and scalability.

Keywords: Constraints, Data Matching, Deduplication, Entity Resolution, Record Linkage, Weights

INTRODUCTION

In real-life applications, constraints on ER ubiquitously exist. Such constraints can be obtained from a variety of sources: background knowledge (Schewe & Wang, 2013), external data sources (Wang, Wang, Li, & Gao, 2013), domain experts, etc. Some constraints may be captured at the instance level (Tung, Han, Lakshmanan, & Ng, 2001; Wagstaf & Cardie, 2000), e.g., “PVLDB” refers to “VLDB Endowment” and vice versa. Some constraints can be specified at the schema level (Arasu, Re, & Suciu, 2009; Chaudhuri, Das Sarma, Ganti, & Kaushik, 2007; Shen, Li, & Doan, 2005), e.g., two paper records refer to two different papers if they do not have the same page number. Using constraints allow us to leverage rich domain semantics for improved ER quality. Nevertheless, constraints may not be completely satisfied by records in the underlying databases due to the existence of dirty data, missing data, exceptions, etc. In such cases, com-
mon approaches are to conduct manual reviews of conflicts, or relax the satisfaction requirement of constraints by allowing some constraints to be violated in terms of a predefined cost model. This helps to produce solutions, but often also requires expensive computational resources for finding an optimal solution. Such burden has prevented constraints from being widely applied to solve the ER problem in many application areas.

In this paper, we study the question of how to specify and use weighted constraints for performing ER tasks. We attempt to establish a unified framework that incorporates semantic capabilities (in form of weighted constraints) into existing ER algorithms to improve the quality of ER, while still being computationally efficient. A key ingredient in achieving this is to associate each constraint with a weight that indicates the confidence of domain experts on the robustness of semantic knowledge it represents. A good number of works have studied constraints in the literature of ER (Arasu et al., 2009; Chaudhuri et al., 2007; Doan, Lu, Lee, & Han, 2003; Dong, Halevy, & Madhavan, 2005; Tung et al., 2001; Wagstaf & Cardie, 2000; Schewe & Wang, 2014). However, little work has been carried out on how to deal with weighted constraints. Our work was motivated by Dedupalog (Arasu et al., 2009), a declarative framework for resolving entities using constraints without weights, and has extended the previous work (Z. Shen & Wang, 2014) by taking the interaction among multiple entity types into consideration. We believe that weights can provide useful insights concerning ambiguity or conflicting information. For instance, when two records \( u \) and \( v \) are identified as a match by a constraint \( r_1 \) and as a non-match by a constraint \( r_2 \) simultaneously, how can we decide whether \((u, v)\) is a match or non-match? If \( r_1 \) and \( r_2 \) have the weights 0.9 and 0.6 respectively, one may decide that \((u, v)\) should be a match since \( r_1 \) has a higher weight than \( r_2 \).

**Contributions**

We propose a unified framework that allows us to associate a weight for each constraint, capturing the confidence for its robustness in an ER model. This framework has the following three main components:

- **ER Matching**: We propose a two-stage process consisting of soft prediction and hard validation. Soft predication generates a matching result that can integrate contradictory information provided by soft constraints, and hard validation repairs erroneous matches and non-matches that are detected.
- **ER Clustering**: We explore two methods, called relative constrained neighbour (RCN) and pairwise nearest neighbour (PNN), for building an efficient mechanism that can generate a robust ER clustering in the presence of weighted constraints.
- **ER Propagation**: We characterize propagation paths among different entity types using a dependency graph, and develop a way of choosing a “good” propagation order based on the weights of constraints.

Our experimental results provide us the following observations:

1. **Weighted Constraints are Useful**: Weighted constraints may considerably improve both accuracy and efficiency of entity resolution because weights can serve as a perfect means for solving various conflicts that may be computationally challenging, such as conflicts between positive and negative constraints.
Optimization of Multidimensional Aggregates in Data Warehouses
[www.igi-global.com/article/optimization-multidimensional-aggregates-data-warehouses/3367?camid=4v1a](www.igi-global.com/article/optimization-multidimensional-aggregates-data-warehouses/3367?camid=4v1a)

CORBA-Based Run-Time Architectures for Workflow Management Systems
[www.igi-global.com/article/corba-based-run-time-architectures/51159?camid=4v1a](www.igi-global.com/article/corba-based-run-time-architectures/51159?camid=4v1a)