Entity Resolution: Past, Present and Yet-to-Come
From Structured to Heterogeneous, to Crowd-sourced, to Deep Learned

ABSTRACT
Entity Resolution (ER) lies at the core of data integration, with a bulk of research focusing on its effectiveness and its time efficiency. Most past relevant works were crafted for addressing Veracity over structured (relational) data. They typically rely on schema, expert and external knowledge to maximize accuracy. Part of these methods have been recently extended to process large volumes of data through massive parallelization techniques, such as the MapReduce paradigm. With the present advent of Big Web Data, the scope moved towards Variety, aiming to handle semi-structured data collections, with noisy and highly heterogeneous information. Relevant works adopt a novel, loosely schema-aware functionality that emphasizes scalability and robustness to noise. Another line of research focuses on Velocity, i.e., processing data collections of a continuously increasing volume.

In this tutorial, we present the ER generations by discussing past, present, and yet-to-come mechanisms. For each generation, we outline the corresponding ER workflow along with the state-of-the-art methods per workflow step. Thus, we provide the participants with a deep understanding of the broad field of ER, highlighting the recent advances in crowd-sourcing and deep learning applications in this active research domain. We also equip them with practical skills in applying ER workflows through a hands-on session that involves our publicly available ER toolbox and data.

1 GOALS AND OBJECTIVES
Entity profiles assemble valuable information about real-world objects. Hence, entities constitutes the core organizational unit of structured (e.g., relational databases) as well as semi-structured data (e.g., knowledge bases, such as DBPedia and Geonames). Various data management applications, such as query answering [47], are based on entity semantics and connections in order to improve their performance. Typically, these applications require the integration of different profiles that pertain to the same real-world object [11, 18]. The task of inter-linking and deduplicating (i.e., canonicalizing) data instances that describe the same real-world objects is called Entity Resolution (ER) [12].

ER is a relatively old problem that was mainly crafted for structured data, which were described by schemata of known semantics and quality [11]. This schema knowledge allowed experts to develop customized solutions that effectively addressed Veracity, i.e., the various forms of inconsistencies, noise or errors in entity profiles, which are introduced during manual data entry, or by the limitations of the automatic extraction techniques [23]. For even higher effectiveness, labelled instances are also typically used in order to automatically learn matching rules that simultaneously maximize precision and recall [48, 60, 61].

© 2020 Copyright held by the owner/author(s). Published in Proceedings of the 23rd International Conference on Extending Database Technology (EDBT), March 30–April 2, 2020, ISBN 978-3-89318-053-7 on OpenProceedings.org. Distribution of this paper is permitted under the terms of the Creative Commons license CC-by-nc-nd 4.0.

Figure 1: The workflow of the 1st and 2nd ER generations.

The end-to-end workflow implemented by the 1st generation of ER solutions is depicted in Figure 1 [11]. The first step, Schema Matching, creates mappings between the attributes of the input entities based on their relatedness, as inferred from the similarity of their structure, name and/or values [6, 45]. By identifying semantically identical attributes (e.g., “profession” and “job”), it facilitates the schema-aware functionality of the subsequent workflow steps.

The second step, which is called Blocking, addresses the quadratic time complexity, O(n²), of brute-force ER, which compares every entity profile with all others [11]. Blocking reduces the executed comparisons to a significant extent by sacrificing recall to a minor extent. It restricts the computational cost by comparing only the most similar entity profiles, as they are determined by signatures that are composed of (combinations of) parts of values that correspond to the most informative attribute names [11], E.g., two person entities are likely matches if their addresses have the same zip code.

The entities that co-occur in at least one block are compared during the third step, which is called Entity Matching. This applies a combination of string similarity measures to the values of selected attribute names. The resulting degree of similarity is then used to assign the entity pairs into one of the three possible categories, i.e., match, non-match or uncertain [11]. In case of collective approaches, the latest decision is propagated to neighboring entities, i.e., entities connected with important relationships to the compared pair, so as to refine their matching likelihood [7, 16].

Note that each step accommodates both learning-based and non-learning methods [41]; the former methods leverage labelled instances to extract effective rules through a Machine Learning algorithm, while the latter methods rely on heuristics that capture expert or domain knowledge.

The same workflow lies at the core of the 2nd generation, which additionally targets Volume, i.e., the cases where the input data comprise (tens of) millions of entity profiles. Typically, this challenge is addressed through the new paradigm for massive parallelization, i.e., MapReduce [14]. Several techniques for Blocking [38] and Entity Matching [9] have been adapted to MapReduce so that they scale to voluminous datasets. Special care is also taken to avoid underutilization of the computational resources through Load Balancing techniques [39, 77].

A shift was marked by the 3rd generation of the ER end-to-end workflow, which is depicted in Figure 2. In addition to Veracity and Volume, its goal is to address Variety, which is caused by the unprecedented levels of schema heterogeneity and noise as well as the loose schema binding of unclear semantics [12, 17]. Instead of a database-like schema, there is a rich diversity of
The first step in the new workflow is *Schema Clustering*, which clusters together attributes with similar values, regardless of their semantics. The goal is to improve the performance of the subsequent steps. E.g., Blocking uses the created schema clusters and the associated signatures (i.e., blocking keys) to split large blocks into smaller ones. This significantly enhances precision for a negligible (if any) impact on recall. This idea has been successfully applied to Blocking via Attribute Clustering [52] and to Meta-blocking via BLAST [62].

The second step, which is called *Block Building*, creates a set of blocks by disregarding schema knowledge and the ensuing human intervention completely. Through a schema-agnostic approach that leverages redundancy, it is inherently crafted for tackling the unprecedented levels of schema heterogeneity in semi-structured data. In this way, it yields blocks of very high recall, but very low precision, independently of human intervention and domain/expert knowledge [12, 54].

The third step of the workflow is *Block Processing*, which enhances precision to a significant extent at a limited, if any, cost in recall [52, 54, 56]. To this end, it refines the original blocks by efficiently removing comparisons that are repeated or involve non-matching entities. Its techniques are distinguished into two categories: the *Block Cleaning* ones operate at the coarse-grained level of entire blocks (e.g., Block Clustering [26]), while the *Comparison Cleaning* ones operate at the fine-grained level of individual comparisons (e.g., Meta-blocking [18, 53] and Blast [62]). In both cases, all techniques are generic and schema-agnostic by definition, thus applying naturally to both structured and semi-structured data [56].

Subsequently, *Entity Matching* executes all comparisons contained in the final set of blocks. Typically, this process depends heavily on neighbor similarity, using the entity relations in the semi-structured data. This is done through an iterative process that discovers duplicate entities gradually and propagates the latest matches to related entities that could benefit from them [42, 44, 66]. This step can also consider probabilistic matching of the entities, e.g., [1, 36].

The end result of Entity Matching is a *similarity graph*, which conveys a node for every entity and a weighted edge for every pair of entities that have been compared. This intermediate model is transformed into the final outcome of ER by *Entity Clustering* [34], which partitions the graph nodes into equivalence clusters - every cluster contains all duplicate entity profiles that actually correspond to the same real-world object. These techniques are schema-agnostic by default, as they exclusively consider the information contained in the similarity graph.

The *4th generation* of ER goes beyond the previous ones, by also addressing *Velocity*. This pertains to the continuously increasing volume of available data that imposes special ER challenges, e.g., the data set can never be considered as final, and incoming data might alter the existing ones. To address them,
2 SCOPE AND COVERAGE

Our tutorial aims to provide an overview of the state-of-the-art techniques for all generations of End-to-End ER, analyzing each one in a different session of ~10 minutes. More emphasis is devoted to the approaches leveraging external knowledge in order to upgrade any workflow step in any generation (~30 minutes), while a hands-on session discusses the main ER tools and demonstrates the latest version of JedAI (~10 minutes). Together with the introduction, 5 minutes for questions and the conclusions, the intended duration of the tutorial is 1.5 hours. The content of the individual sessions is outlined below:

I. Introduction and motivation
   • Preliminaries on Entity Resolution [12, 18]
   • Fundamental Assumptions, Principles and Definitions [23]
II. The 1st ER Generation: Tackling Veracity
   • Schema Matching [6, 19]
   • Blocking [11, 37, 61]
   • Entity Matching [5, 16, 60]
III. The 2nd ER Generation: Tackling Volume and Veracity
   • Parallel Blocking [38]
   • Parallel Entity Matching [59]
   • Load Balancing [39, 77]
IV. The 3rd ER Generation: Tackling Variety, Volume and Veracity
   • Schema Clustering [52, 62]
   • Block Building [50–52]: Parallel Methods [12]
   • Block Processing [8, 26, 55, 56, 62]: Parallel Methods [21]
   • Entity Matching [42, 44, 66]: Parallel Methods [9, 22]
   • Entity Clustering [34]
V. The 4th ER Generation: Tackling Velocity, Variety, Volume and Veracity
   • Progressive ER for (Semi-)Structured Data [58, 63, 76]
   • Incremental Entity Resolution [33, 74]
   • Query-Driven Entity Resolution [2–4, 71]
   • Query Analytics for Entity Resolution [35, 64].
VI. Entity Resolution Revisited:
   Leveraging External Knowledge
   • Deep Learning for Entity Resolution [20, 48]
   • Crowd-sourced Entity Resolution:
     – Generating HTs [15, 43, 70]
     – Formulating HTs [25, 67–69, 72, 75]
     – Balancing accuracy and monetary cost [10, 28, 73, 78]
     – Restrict the labour cost [13, 30]
VII. Hands-on Session: ER tools
   • The state-of-the-art end-to-end ER tools [40]
   • The JedAI Open Source Toolkit [57]
VIII. Challenges and Final Remarks
   • Automatic Parameter Configuration [46, 49]
   • Multi-modal Entity Resolution
   • Conclusions

3 INTENDED AUDIENCE AND MATERIAL

Our tutorial is example-driven, avoiding excessive technical details and proofs. As a result, there is no prerequisite knowledge, apart from a basic understanding of data management technology. This renders it suitable for a broad audience, covering not only students and researchers, but also practitioners and developers. In other words, it is intended for anyone with an interest in understanding the main techniques for scalable and robust end-to-end Entity Resolution over structured and semi-structured data, using both non-learning and learning-based techniques.

In addition to the theoretical background in the state-of-the-art in the field, the tutorial also presents available entity-related resources, enabling the participants to directly work on the particular domain. Discussed resources include available data as well as the state-of-the-art tools for performing end-to-end Entity Resolution, like Magellan [40] and JedAI [57], which can be readily used to tackle ER problems via numerous combinations of the most prominent methods.

Tutorial Material. The material of the tutorial is distributed through the conference website1 as well as through a dedicated website2. In both locations, we also give pointers and guidelines for the ER toolkit that is used during the hands-on session. All relevant code is publicly released through the Apache License 2.0, which supports both academic and commercial uses.

4 PRESENTERS

The tutorial is given by three presenters:

1. George Papadakis is a Research Fellow at the Department of Informatics of the University of Athens, Greece, and an Internal Auditor of Information Systems at the Public Power Company, the main electricity company in Greece.
2. Ekaterini Ioannou is an Assistant Professor at the University of Tilburg, Netherlands.
3. Themis Palpanas is a Senior Member of the French University Insitute (IUF), and a Professor of Computer Science at the University of Paris, France.

All authors have published papers related to Entity Resolution, focusing on the efficient management of large data collections as well as on addressing various challenges, such as uncertainty, volatility, and correlations.

Acknowledgements. This work was partially funded by the EU H2020 project ExtremeEarth (825258).

REFERENCES


1https://diku-dk.github.io/edbticdt2020
2https://research.tilburguniversity.edu/en/projects/4ger