Large-Scale Entity Resolution

Erhard Rahm, Eric Peukert

Synonyms

Object Matching | Record Linkage | Link Discovery | Data Deduplication

Definition

The goal of entity resolution is the identification of semantically equivalent objects within one data source or between different sources. In the context of Big Data, there is a growing need for large-scale entity resolution to find matching entities within very large and between many data sources. This requires effectively parallelizing entity resolution tasks within cluster environments.

Overview

Entity resolution (ER) is the task to identify semantically equivalent entities referring to the same real-word object (e.g. persons, products, publications, or movies) within one data source or between different sources. This task is also known as data deduplication, object matching, record linkage or link discovery. ER is of core importance for data cleaning and data integration and has been addressed for a long time in practice and research (Rahm and Do 2000, Elmagarmid et al 2007, Christen 2012).

The traditional focus of ER was on structured data in relational databases. The example in **Figure 1** shows two input relations from different sources with address data that contain duplicates. The example illustrates some of the challenges of data deduplication tasks such as heterogeneous attribute names and value differences due to different conventions (e.g. for gender), abbreviations, omissions, nick names and errors in the data. Typically, ER requires to first determine comparable attributes and then computing and combining the similarity for multiple attributes, e.g., for name, gender and address in the example.

Cno	*LastName*	*FirstName*	*Gender*	*Address*	*Phone/Fax*
24	Smith	Christoph	M	23 Harley St, Chicago IL, 60633-2394	333-222-6542 / 333-222-6599
493	Smith	Kris L.	F	2 Hurley Place, South Fork MN, 48503-5998	444-555-6666

CID	*Name*	*Street*	*City*	*Sex*
11	Kristen Smith	2 Hurley Pl	South Fork, MN 48503	0
24	Christian Smith	Hurley St 2	S Fork MN	1

Figure 1: Exemplary entity resolution task (from (Rahm and Do 2000))

Large-scale entity resolution for Big Data introduces a number of challenges:

- Data sources can be very large with many millions of entities making it hard to achieve both effective and efficient entity resolution. Effectiveness asks for a good match quality so that only truly matching entities are identified (good precision) and that all matches are found (good recall). Efficiency is a problem since it is not feasible to pairwise compare all entities with each other for large datasets. Hence, it is imperative to reduce the number of comparisons (by filtering and blocking techniques) and to also apply parallel entity resolution on many processors.

- Relevant data can be spread over thousands of data sources so that determining pairwise matches between two sources is not sufficient. Rather, a more holistic entity resolution is necessary in such cases where matching entities of all sources are grouped or clustered together such that all entities in such clusters match with each other (Rahm 2016). An example for such large-scale entity resolution are product offers from thousands of web shops that should be matched with each other, e.g., to allow a price comparisons (Köpcke et al 2012).
- Entities especially from the web or from social networks, are often only semi-structured and contain free text and also image content. Data quality is further reduced due to the frequent use of heterogeneous names and abbreviations as well as missing values.
- Data sources can change quickly such that existing entities are changed and deleted and new entities are added. As a result, entity resolution should be an incremental process such that previously computed matches are retained and the match result is only updated for changed or new entities.

Entity resolution is typically implemented as a complex workflow to identify the matches within one data source or in several data sources. The output is either a set of pairwise correspondences (links) of matching entities, also called mapping. Alternatively all matching entities can be grouped or clustered together which is especially useful for more than two data sources. Fig. 2 shows the main steps of typical ER workflows that include preprocessing/data cleaning, blocking, similarity computation and match classification (Christen 2012, Elmagarmid et al 2007, Dong and Srivastava 2015). Optionally, the results of pairwise matching can be clustered to group all directly or indirectly matching entities.

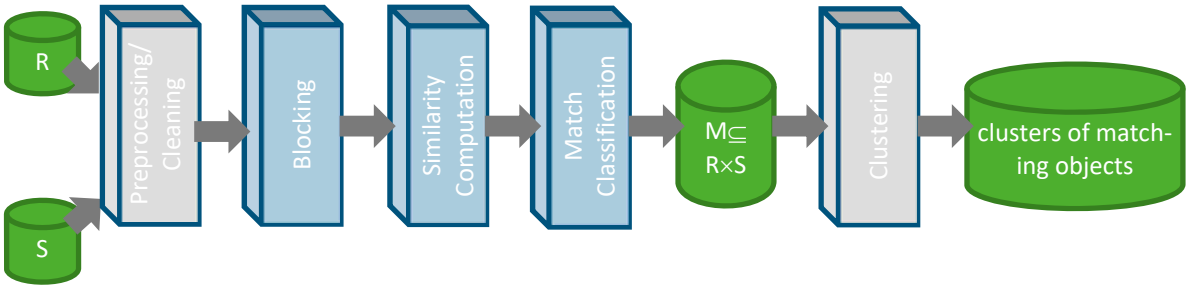

Figure 2: Entity resolution workflow for two input sources R and S

In the ***preprocessing*** phase missing attributes can be computed or values are cleaned with the help of background knowledge. Textual content can be harmonized by removing capital letters, replacing special characters, tokenizing textual values or applying NLP techniques such as stemming or stop word removal. Furthermore, names or keywords relevant for matching (so-called features) can be extracted from textual attributes for further processing. For example, matching product offers can benefit from extracting manufacturer information and so-called product codes from the product descriptions for later similarity computation (Köpcke et al 2012).

The ***blocking*** phase is crucial to scale ER to large datasets to reduce the number of match comparisons. Similar entities are grouped within blocks, e.g., based on approaches such as Standard Blocking or Sorted Neighborhood (Christen 2012). With Standard Blocking entities are grouped into partitions or blocks based on a blocking function on the values of one or more attributes. The subsequent similarity computation is restricted to the pairwise matching of entities from the same block. For product offers one could use manufacturer as a blocking

key so that only offers for products of the same manufacturer need to be compared with each other. This significantly reduces the search space but often reduces recall. Multi-pass blocking applies multiple blocking functions to reduce the loss on recall at the expense of more comparisons (Christen 2012).

The ***similarity computation*** phase computes pairwise similarities based on attribute level matching with domain-specific or general similarity functions, e.g., utilizing string similarity metrics (edit distance, n-gram, TF/IDF, etc.). Moreover, context-based matchers are applicable that incorporate the entity neighborhood for similarity computation, e.g. related products such as the cameras for which an accessory product such as a replacement battery can be used.

In the ***match classification*** phase, the computed entity similarities are used to decide whether a pair of entities matches or not. The classification can be expressed as rules based on a weighted combination of similarity values and a similarity threshold or it can be derived from a machine learning-based classifier that has been determined for suitable training data. The match candidates can be further post-processed or filtered, e.g., to only consider the best matches if there are several match candidate per entity. For clean datasets without duplicates there should be at most one matching entity in another dataset.

In the ***clustering*** phase the computed matches (correspondences) can be used to group all directly and indirectly matching entities. The simplest approach of Connected Components computes a transitive closure of all correspondences. It can achieve a high match recall but often suffers from poor precision since a single wrong link can lead to large clusters. More sophisticated approaches such as correlation clustering (Pan et al 2015) try to maximize the entity similarities within clusters and to minimize similarities between clusters. (Hassanzadeh et al 2009) and (Saeedi et al 2017) comparatively evaluate several approaches for entity clustering.

These phases are implemented within many tools for entity resolution and link discovery as surveyed in (Köpcke and Rahm 2010, Christen 2012, Nentwig et al 2017). The comparative evaluation of several tools in (Köpcke et al 2010) showed that learning-based match classification mostly achieves better match quality than rule-based approaches especially for more complex match tasks such as for product matching. Most of the previous tools, however, lack support for parallel processing making them insufficiently capable to meet the performance and scalability requirements of Big Data applications.

Key Research Findings

The performance and scalability requirements for Big Data applications require parallel entity resolution approaches in addition to the use of blocking. (Kolb et al 2012) has therefore investigated the utilization of Hadoop clusters and the MapReduce paradigm for parallel entity resolution. As illustrated in Figure 3, there are two main phases. In the map phase, the input records are read in parallel and the blocking function is applied to the input data entities. The blocking key is used to distribute and group entities of the same block to the same processing node for the matching phase. In the reduce phase, a pairwise matching is performed on all record pairs of the same block. Different blocks are processed in parallel.

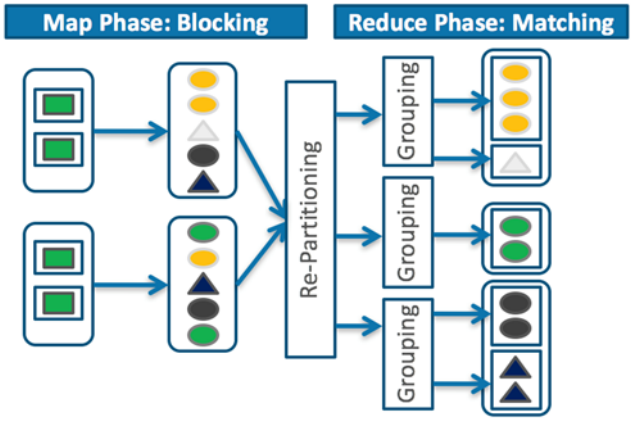

Figure 3: Parallel entity resolution with Map Reduce

Such a parallel entity resolution approach faces a potential load balancing problem in the reduce phase since the block sizes may vary to a large degree. The slowest reducer dominates the overall execution time and large blocks prevent the effective utilization of more than a few nodes. For load balancing, (Kolb et al 2012) proposes two solutions called BlockSplit and PairRange. BlockSplit splits large blocks such there are multiple smaller match tasks that can be processed on different reducer nodes without creating overload situations. The approach leads to the replication of a subset of the entities that are part of several match tasks. The alternate PairRange approach globally enumerates all pairwise comparisons and evenly distributes these over all reducers. In both approaches an additional MapReduce-Job is needed to compute the number and sizes of blocks for the configured blocking scheme. The load balancing approaches achieved a high scalability and proved to be stable against data skew. They are part of the DeDoop match tool that additionally offers machine-learning-based classification and a web-based UI for managing entity resolution tasks (Kolb and Rahm 2013).

Recently, alternative processing models such as Apache Flink and Apache Spark have become popular which provide a richer set of operators than MapReduce and in-memory computing for improved performance. Saeedi et. al. ported many of the Dedoop features to Apache Flink within the FAMER system (Saeedi et al 2017). In contrast to DeDoop, FAMER also supports multi-source (holistic) entity resolution with more than two input sources. For this purpose, it offers parallel implementations for several entity clustering schemes.

Further proposals for holistic (multi-source) entity resolution with entity clustering include (Böhm et al 2012), (Pershina et al 2015) and (Nentwig et al 2016). (Gruenheid et al 2014) studies incremental entity resolution that can improve scalability since new entities and sources are compared to existing clusters rather than with all other sources and entities.

Directions for Future Research

Novel machine learning techniques such as *deep learning* have been very successful in areas such as face or speech recognition and text mining. Such approaches are also promising for entity resolution, especially for web entities with textual descriptions, where deep neural networks may be able to automatically identify match-relevant features within such descriptions without extensive and complex preprocessing. Initial approaches in this direction are promising (Ebraheem et al 2017) but do not yet fully exploit the potential of deep learning. Both the training of such methods and the model application needs to be parallelized to be applicable for Big Data Integration tasks. The approaches should also be extended to include not only text attributes but also image content for matching.

Another area deserving more attention is scalable (parallel) entity resolution for graph-structured data where graphs (networks) can include entities and relationships of different types (e.g. publication and author entities with author and cite relationships). Here, the similarity between entities of different graphs should consider not only the entity attributes but also the graph neighborhood. Furthermore, both entities (vertices) and relationships (edges) of the different types need to be matched and possibly fused within an integrated data graph. Moreover, holistic matching with more than two graphs and incremental entity resolution should be supported (Rahm 2016).
Another dimension that is beginning to gain interest is temporal record linkage that pays attention to the evolution of entities over time such as address changes (Li et al 2011, Chiang et al 2014, Christen et al 2017). Such techniques should also become scalable to large and many data sources, e.g., within an incremental approach that keeps track of different entity versions.

References

- Böhm C, de Melo G, Naumann F, Weikum G (2012) LINDA: distributed Web-of- Data-scale entity matching. Proc. CIKM
- Chiang YH, Doan A, Naughton JF (2014) Modeling entity evolution for temporal record matching. Proc. ACM SIGMOD
- Christen P (2012) Data Matching - Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection. Springer
- Christen V, Groß A, Fisher J, Wang Q, Christen P, Rahm E (2017) Temporal group linkage and evolution analysis for census data. Proc. EDBT
- Dong XL, Srivastava D (2015) Big Data Integration. Morgan and Claypool
- Ebraheem M, Thirumuruganathan S, Joty S, Ouzzani M, Tang N (2017) DeepER -- Deep Entity Resolution. CoRR abs/1710.00597
- Elmagarmid AK, Ipeirotis PG, Verykios VS (2007) Duplicate record detection: A survey. IEEE TKDE, 19(1):1–16
- Gruenheid A, Dong XL, Srivastava D (2014) Incremental record linkage. PVLDB, 7(9):697–708
- Hassanzadeh O, Chiang F, Lee HC, Miller RJ (2009) Framework for evaluating clustering algorithms in duplicate detection. PVLDB, 2(1):1282–1293
- Kolb L, Thor A, Rahm E (2012). Load balancing for MapReduce-based entity resolution. Proc. ICDE
- Kolb L, Rahm E (2013) Parallel entity resolution with Dedoop. Datenbank-Spektrum, 13(1), 23-32.
- Köpcke H, Rahm E (2010) Frameworks for entity matching: A comparison. Data & Knowledge Engineering, 69(2):197–210
- Köpcke H, Thor A, Rahm, E (2010) Evaluation of entity resolution approaches on real-world match problems. PVLDB, 3(1-2), 484-493
- Köpcke H, Thor A, Thomas S, Rahm E (2012) Tailoring entity resolution for matching product offers. Proc. EDBT, 545–550
- Li P, Dong XL, Maurino A, Srivastava D (2011) Linking temporal records. Proc. VLDB Endowment, 4 (11): 956–967
- Nentwig M, Groß A, Rahm E (2016) Holistic entity clustering for linked data. In IEEE Data Mining Workshops (ICDMW)

- Nentwig M, Hartung M, Ngonga Ngomo AC, Rahm E (2017) A survey of current link discovery frameworks. Semantic Web, 8(3), 419-436
- Pan X, Papailiopoulos D, Oymak S, Recht B, Ramchandran K, Jordan M (2015) Parallel correlation clustering on big graphs. Proc. Advances in Neural Information Processing Systems
- Pershina M, Yakout M, Chakrabarti K (2015) Holistic entity matching across knowledge graphs. Proc. IEEE Big Data Conf
- Rahm E, Do HH (2000) Data Cleaning: Problems and Current Approaches. IEEE Data Engineering Bulletin
- Rahm E (2016) The case for holistic data integration. Proc. ADBIS, Springer LNCS 9809
- Saeedi A, Peukert E, Rahm E (2017) Comparative Evaluation of Distributed Clustering Schemes for Multi-source Entity Resolution. Proc. ADBIS, Springer LNCS 10509