Evaluation of Duplicate Detection Algorithms: From Quality Measures to Test Data Generation

Fabian Panse
Universität Hamburg
Hamburg, Germany
panse@informatik.uni-hamburg.de

Felix Naumann
Hasso Plattner Institute, University of Potsdam
Potsdam, Germany
felix.naumann@hpi.de

Abstract—**Duplicate detection identifies multiple records in a dataset that represent the same real-world object. Many such approaches exist, both in research and in industry. To investigate essential properties of duplicate detection algorithms, such as their result quality or runtime behavior, they must be executed on suitable test data. The quality evaluation requires that these test data are labeled, constituting a ground truth. Correctly labeled, sizable, and real or at least realistic test datasets, however, are not easy to obtain, creating an obstacle for the advancement of research. In this tutorial, we present common methods to evaluate duplicate detection algorithms and to generate labeled test data. We close with a discussion of open problems.**

I. Introduction

Duplicate detection, also known as *entity resolution* [1] or *record linkage* [2], is an essential aspect in data cleaning [3]–[5], data integration [6], [7] and schema matching [8], [9]. Whereas the identification of exact duplicates is rather trivial, the detection of so-called *fuzzy* duplicates can be a tough challenge, depending on how much the duplicate records differ from one another due to data errors (e.g., typos, OCR, or calculation errors), heterogeneous representations (e.g., different formats, vocabularies/terminologies, languages, or units of measurements), as well as missing and outdated values. Because the detection of duplicates is such an important but also difficult task, much research has been performed in this area over the last decades [10]–[13].

The detection of duplicates is not only a complex problem, but the suitability of a solution strongly depends on (i) the considered domain, (ii) the characteristics of the given data (e.g., size, schema complexity, error proneness), (iii) the quality requirements of the user in terms of precision and recall, and (iv) cost including the amount of required training data, careful design of similarity rules or runtime budget. Due to all these (partly conflicting) factors and the resulting heterogeneity of use cases, none of the existing algorithms has shown to be a generally applicable and superior solution. Instead, in every use case, it remains a difficult (and expensive) task to choose and configure them, so that they provide adequate results.

Due to these circumstances, the *evaluation* of duplicate detection algorithms with proper test data is an essential necessity and needs to be performed carefully. However, the selection of suitable evaluation measures and the acquisition of correctly labeled test data is challenging.

In this tutorial, we give an overview of the dimensions and techniques for evaluating duplicate detection algorithms with a focus on quality measures and test data generation. We present state-of-the-art research and discuss open challenges, covering the composition of a typical duplicate detection pipeline, the large range of existing quality measures, and different ways to generate test data with labeled duplicates. We provide overviews of existing methods and tools for test data generation and discuss their respective strengths and weaknesses. We also focus on data profiling as an important prerequisite for domain-independent and targeted pollution of real-life datasets with fuzzy duplicates. Finally, we motivate open problems by discussing existing use cases.

II. Tutorial Outline

This tutorial is split into five parts:

(1) **Duplicate Detection.** We introduce the traditional problem of duplicate detection using several real-life use cases and a formal definition, describe its application contexts, and sketch the steps of a typical duplicate detection pipeline.

(2) **Quality Evaluation.** We motivate the evaluation of duplicate detection algorithms and discuss which requirements different application contexts impose on such an evaluation. Thereafter, we present the composition of test datasets and discuss several kinds of ground truths (e.g., gold and silver standards) as well as several measures to quantify the algorithms' quality.

(3) **Test Data Generation.** We discuss methods to generate labeled test data and compare them with regard to salient properties, such as correctness, domain independence, and scalability. Moreover, we present a study on popular test datasets and give an overview of existing tools for automatic test data generation.

(4) **Data Profiling.** We discuss how metadata shall influence test data generation of arbitrary domains. We give an overview of several data profiling algorithms to discover such metadata to ultimately assist in producing realistic test data.

(5) **Open Challenges.** Finally, we propose a set of requirements for test data generation tools, evaluate existing tools based on these requirements, and discuss open problems based on the results of this evaluation.

2375-026X/21/$31.00 ©2021 IEEE
DOI 10.1109/ICDE51399.2021.00269

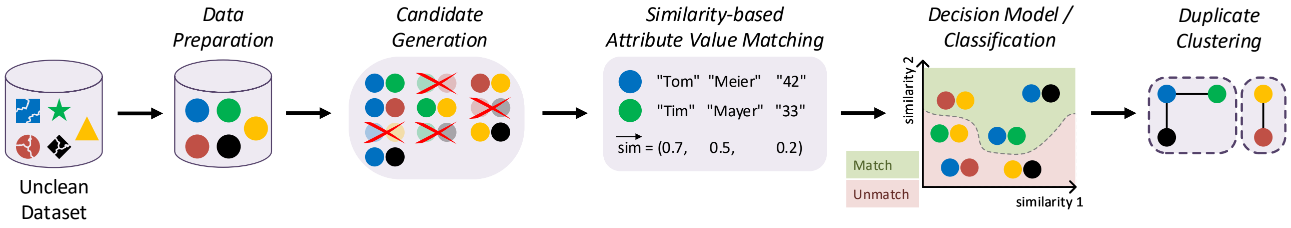


Fig. 1. The five steps of a typical duplicate detection pipeline based on pairwise record comparisons

III. Goals and Objectives

This section provides details about the tutorial parts.

A. Duplicate Detection

We introduce our audience to the topic of duplicate detection in four steps. First, we motivate its use through some real life use cases. Second, we describe the problem of detecting duplicates formally. Thereafter, we briefly discuss different application contexts, such as integration, cleaning and linkage, in which the detection of duplicates is relevant and highlight their differences. Finally, we present the different steps of a typical duplicate detection pipeline (see Figure 1) and briefly sketch some popular algorithms that are used to execute them.

The first step, data preparation, standardizes, cleans and enhances data with additional meta information [14]. The second step generates candidates with the aim to reduce the quadratic search space by using simple and efficient mechanisms to pair only those records that are potential duplicates [15], [16]. In the third step, the records of each of these candidate pairs are compared at the attribute level by using similarity (or distance) measures, such as Levenshtein, Jaro-Winkler, Monge-Elkan, or Jaccard [6], [11], or learned similarity models. Based on these attribute similarities, a decision model decides in the fourth step whether two compared records are duplicates (Match) or not (Unmatch). For this purpose, a variety of methods can be used ranging from simple distance-based models to automatically learned models (supervised and unsupervised) [6], [13], [17], [18]. The final step is a clustering [19], [20] that uses the pairwise duplicate decisions to compute a globally consistent result, i.e., one that satisfies transitivity.

B. Quality Evaluation

We start the second part by motivating the necessity for evaluating duplicate detection algorithms in research and practice. This can be to investigate the behavior of newly developed algorithms, but also to identify and tune an adequate setting (i.e., selection of a suitable algorithm as well as the setting of its parameters) for a particular use case. Moreover, datasets with labeled duplicates are required as training data if a supervised machine learning algorithm should be used in any step of the duplicate detection pipeline (e.g., a support-vector machine in the decision model).

Thereafter, we specify the composition of a suitable test dataset that depends on the application context under consideration. Moreover, we discuss different ways to model information on true duplicates within test datasets, especially when they are not fully known or not known with certainty. Here, we distinguish gold and silver standards [21].

Finally, we present several measures that have been used to quantify the quality of a duplicate detection process [22]–[25]. These measures ranges from well-known pairwise measures, such as recall, precision and F_1-score, to less known pairwise measures, such as the H-measure [26], to measures comparing entire clusterings, such as the closest cluster F_1-score [22], the variation of information [27], or the generalized merge distance [22]. In this discussion, we illustrate how much these measures can differ in their understanding of quality [28].

C. Test Data Generation

In this main part of the tutorial, we describe and compare several methods for generating test data. Moreover, we present a study on popular test datasets, such as CDDB, Cora, the Fodor's and Zagat's restaurant[1], or the Magellan datasets[2], and give an overview on existing test data generators.

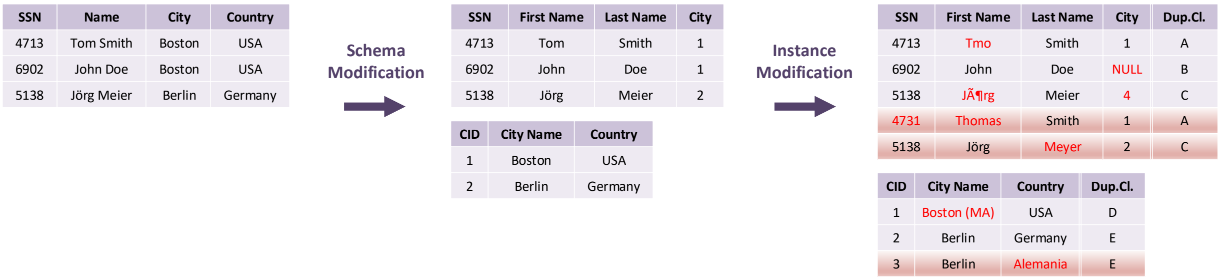
Generation Approaches. Basically, labeled test data for duplicate detection can be generated in five ways:

(i) by manually labeling duplicates in an existing unclean dataset (with the optional help of some duplicate detection tools),
(ii) by running several duplicate detection algorithms on an existing unclean dataset in order to create a silver or annealing standard,
(iii) by integrating several duplicate-free data sources based on an error-free global identifier (e.g., the ISBN or SSN),
(iv) by synthesizing a complete dataset (including duplicates) from scratch, or
(v) by polluting an existing clean dataset with duplicates, errors and inhomogeneities.

Each of these approaches have their benefits and drawbacks, which we discuss and compare. For example, one problem with manually labeled datasets is that such a labeling process is expensive and can be applied only to small datasets.

Current Test Datasets. We give an overview of the evaluation of duplicate detection algorithms in more than 50 contributions from top journals and conferences and point out open problems in these evaluations. These include the limitations of existing

[1] Compiled at http://hpi.de/naumann/projects/repeatability/datasets
[2] https://sites.google.com/site/anhaidgroup/useful-stuff/data



SSN	Name	City	Country
4713	Tom Smith	Boston	USA
6902	John Doe	Boston	USA
5138	Jörg Meier	Berlin	Germany

SSN	First Name	Last Name	City
4713	Tom	Smith	1
6902	John	Doe	1
5138	Jörg	Meier	2

CID	City Name	Country
1	Boston	USA
2	Berlin	Germany

SSN	First Name	Last Name	City	Dup.Cl.
4713	Tmo	Smith	1	A
6902	John	Doe	NULL	B
5138	JÃ¶rg	Meier	4	C
4731	Thomas	Smith	1	A
5138	Jörg	Meyer	2	C

CID	City Name	Country	Dup.Cl.
1	Boston (MA)	USA	D
2	Berlin	Germany	E
3	Berlin	Alemania	E

Fig. 2. Example of modifications on schema and instance level performed by data pollution processes

test datasets, but also the unsatisfactory or even missing documentation of their creation and use.

Error Types and Generation. In both data synthetization and data pollution, errors are created artificially, leading to the problem that the resulting errors and error patterns may be less realistic than those of a real unclean dataset. It is therefore necessary that as many error types (e.g., syntactic, but also semantic and phonetic) as possible are covered and that their selection can be domain and case-specific. In this part, we give a brief overview on potential error types [3], [29] and what information (context or auxiliary) we need to apply them.

Existing Test Data Generators. Because the manual generation of individual test datasets can be time-consuming, it makes sense to use automated approaches. We present and discuss several test data generators, distinguishing between data synthetization tools, such as DBGen [30] and Febrl [29], and data pollution tools, such as TDGen [31], GeCo [32], and DaPo [33]. We also take into account other test data generation tools whose topics are closely related to duplicate detection, such as BART [34] and iBench [35].

D. Data Profiling

In data pollution systems the input data can be changed in two basic ways (see Figure 2). First, the schema (and sometimes even the data model) may be modified, e.g., by splitting/merging attributes or normalizing tables, which includes a migration of the instance data from the old to the new schema. Second, duplicates, errors and heterogeneity are injected into the transformed instance data. Both changes must be as realistic as possible: simple error injection rules are in turn easy to overcome by duplicate detection systems, invalidating their evaluation. In particular, considering cross-attribute properties, such as data dependencies, can yield data that is more aligned with real-world situations. To collect such meta information, data profiling [36] plays an important role in the pollution process. The same applies when synthesizing data based on the properties of a real-life dataset (e.g., to preserve its confidentiality).

Collecting Metadata. To determine realistic errors (e.g., outliers) for the individual attributes or combination of attributes, metadata about these attributes are needed. These include basic statistics, such as the minimum, maximum, average, variance and entropy of numerical attributes, and the length, number of tokens, token lengths and number of occurrences of individual characters or tokens for non-numerical attributes, as well as histograms [37] modeling their distributions.

But it also includes constraints and dependencies, such as unique constraints, inclusion dependencies and functional dependencies [38], which help to understand, prepare and clean the input data, or vice versa to pollute it appropriately. These metadata can be exact (i.e., they apply unconditionally), approximate (i.e., they apply to most but not all records) or conditional (i.e., they apply to only a specific subset of all data). Knowledge of constraints helps to choose suitable error types, such as a violation of an inclusion dependency, during the pollution process. For example, BART is a test data generator that focuses on the injection of errors violating a predefined set of functional dependencies [34].

Data Types and Domains. Often, the input data do not provide any information on data types or the provided types are unspecific (e.g., ZIP codes are not typed as five-digit numbers, but as integers or strings). One major task of data profiling is, therefore, to identify the actual data types of the individual attributes by analyzing the structure of the given instance data. Based on these types, semantic domains, such as book titles, personal names, or postal addresses, can be assigned to the attributes. The acquisition of such information allows the domain-specific selection of suitable error types.

Temporal Metadata. With increasing velocity, outdated data become a major quality issue [39]. However, to simulate outdated values realistically, we need to know:

- when, how often, and how a value changes,
- which intra-record (e.g., phone landline number ↔ residence) and inter-record (e.g., residence of family members) dependencies exist, and
- which integrity constraints are not allowed to be violated at any point in time.

Such statistics and dependencies can be learned only from a historical/temporal database or a data history.

E. Open Challenges

In the last part of the tutorial we evaluate existing approaches to automatic test data generation and derive a number of open challenges from the results of this evaluation. We start with presenting a set of requirements that are essential for test data generation tools [33]. This includes (i) scalability,

(ii) domain independence, (iii) realistic data values & patterns, (iv) realistic & variable error patterns, (v) a simple but adaptable configuration, and (vi) representation diversity (e.g., the tool should be able to produce data of different models).

Thereafter, we compare the different test data generators of Section III-C based on these requirements and highlight several common deficits. Finally, we summarize the current status of duplicate detection evaluation and test data generation and briefly discuss the open challenges we have identified in the course of this tutorial.

IV. Biographies

Fabian Panse is a postdoctoral researcher at the Universität Hamburg (Germany), where he teaches and researches in the fields of deduplication, uncertain data management and data quality since 2009. During this time, he wrote several papers that address the problems of measuring data quality, evaluating duplicate detection algorithms and test data generation.

Felix Naumann is professor at the Hasso Plattner Institute in Potsdam (Germany), where he is heading the information systems group since 2006. He is working in the areas of data integration, data cleaning, and data profiling. Felix has published a large number of papers on these topics at several high-quality conferences and journals. He has also written books on information integration, duplicate detection and data profiling, which have been well received in the community.

References

[1] G. Papadakis, E. Ioannou, and T. Palpanas, "Entity Resolution: Past, Present and Yet-to-Come," in *Proceedings of the International Conference on Extending Database Technology (EDBT)*, 2020, pp. 647–650, tutorial.

[2] T. N. Herzog, F. J. Scheuren, and W. E. Winkler, *Data Quality and Record Linkage Techniques*. Springer, 2007.

[3] E. Rahm and H. H. Do, "Data Cleaning: Problems and Current Approaches," *IEEE Data Eng. Bull.*, vol. 23, no. 4, pp. 3–13, 2000.

[4] V. Ganti and A. D. Sarma, *Data Cleaning: A Practical Perspective*, ser. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2013.

[5] I. F. Ilyas and X. Chu, *Data Cleaning*. ACM, 2019.

[6] A. Doan, A. Halevy, and Z. G. Ives, *Principles of Data Integration*. Morgan Kaufmann, 2012.

[7] X. L. Dong and D. Srivastava, *Big Data Integration*, ser. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2015.

[8] E. Rahm and P. A. Bernstein, "A Survey of Approaches to Automatic Schema Matching," *VLDB Journal*, vol. 10, no. 4, pp. 334–350, 2001.

[9] A. Bilke and F. Naumann, "Schema Matching using Duplicates," in *Proceedings of the International Conference on Data Engineering (ICDE)*, 2005, pp. 69–80.

[10] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios, "Duplicate Record Detection: A Survey," *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, vol. 19, no. 1, pp. 1–16, 2007.

[11] F. Naumann and M. Herschel, *An Introduction to Duplicate Detection*, ser. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2010.

[12] J. R. Talburt, *Entity Resolution and Information Quality*. Morgan Kaufmann, 2011.

[13] P. Christen, *Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*, ser. Data-Centric Systems and Applications. Springer, 2012.

[14] I. K. Koumarelas, L. Jiang, and F. Naumann, "Data Preparation for Duplicate Detection," *Journal on Data and Information Quality (JDIQ)*, vol. 12, no. 3, pp. 15:1–15:24, 2020.

[15] P. Christen, "A Survey of Indexing Techniques for Scalable Record Linkage and Deduplication," *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, vol. 24, no. 9, pp. 1537–1555, 2012.

[16] G. Papadakis, D. Skoutas, E. Thanos, and T. Palpanas, "Blocking and Filtering Techniques for Entity Resolution: A Survey," *ACM Computing Surveys*, vol. 53, no. 2, pp. 31:1–31:42, 2020.

[17] P. Christen, "Automatic Record Linkage using Seeded Nearest Neighbour and Support Vector Machine Classification," in *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*, 2008, pp. 151–159.

[18] A. Doan, P. Konda, P. S. G. C., Y. Govind, D. Paulsen, K. Chandrasekhar, P. Martinkus, and M. Christie, "Magellan: Toward Building Ecosystems of Entity Matching Solutions," *Communications of the ACM (CACM)*, vol. 63, no. 8, pp. 83–91, 2020.

[19] O. Hassanzadeh, F. Chiang, R. J. Miller, and H. C. Lee, "Framework for Evaluating Clustering Algorithms in Duplicate Detection," *PVLDB*, vol. 2, no. 1, pp. 1282–1293, 2009.

[20] U. Draisbach, P. Christen, and F. Naumann, "Transforming Pairwise Duplicates to Entity Clusters for High-quality Duplicate Detection," *Journal on Data and Information Quality (JDIQ)*, vol. 12, no. 1, pp. 3:1–3:30, 2020.

[21] T. Vogel, A. Heise, U. Draisbach, D. Lange, and F. Naumann, "Reach for Gold: An Annealing Standard to Evaluate Duplicate Detection Results," *Journal on Data and Information Quality (JDIQ)*, vol. 5, no. 1-2, pp. 5:1–5:25, 2014.

[22] D. Menestrina, S. Whang, and H. Garcia-Molina, "Evaluating Entity Resolution Results," *PVLDB*, vol. 3, no. 1, pp. 208–219, 2010.

[23] H. Maidasani, G. Namata, B. Huang, and L. Getoor, "Entity Resolution Evaluation Measures," University of Maryland, Tech. Rep., 2012.

[24] M. Barnes, "A Practioner's Guide to Evaluating Entity Resolution Results," *CoRR*, vol. abs/1509.04238, 2015.

[25] C. Nanayakkara, P. Christen, T. Ranbaduge, and E. Garrett, "Evaluation Measure for Group-based Record Linkage," *International Journal of Population Data Science (IJPDS)*, vol. 4, no. 1, 2019.

[26] D. J. Hand, "Measuring Classifier Performance: a Coherent Alternative to the Area under the ROC Curve," *Mach. Learn.*, vol. 77, no. 1, pp. 103–123, 2009.

[27] M. Meila, "Comparing clusterings - An Information Based Distance," *Journal of Multivariate Analysis*, vol. 98, pp. 873 – 895, 2007.

[28] D. J. Hand and P. Christen, "A Note on using the F-measure for Evaluating Record Linkage Algorithms," *Stat. Comput.*, vol. 28, no. 3, pp. 539–547, 2018.

[29] P. Christen and A. Pudjijono, "Accurate Synthetic Generation of Realistic Personal Information," in *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, 2009, pp. 507–514.

[30] M. Hernández and S. Stolfo, "Real-world Data is Dirty: Data Cleansing and The Merge/Purge Problem," *Data Min. Knowl. Discov.*, vol. 2, no. 1, pp. 9–37, 1998.

[31] T. Bachteler and J. Reiher, "Tdgen: A Test Data Generator for Evaluating Record Linkage Methods," German Record Linkage Center, Tech. Rep. wp-grlc-2012-01, 2012.

[32] P. Christen and D. Vatsalan, "Flexible and Extensible Generation and Corruption of Personal Data," in *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*, 2013, pp. 1165–1168.

[33] K. Hildebrandt, F. Panse, N. Wilcke, and N. Ritter, "Large-Scale Data Pollution with Apache Spark," *IEEE Trans. Big Data*, vol. 6, no. 2, pp. 396–411, 2020.

[34] P. C. Arocena, B. Glavic, G. Mecca, R. J. Miller, P. Papotti, and D. Santoro, "Messing Up with BART: Error Generation for Evaluating Data-Cleaning Algorithms," *PVLDB*, vol. 9, no. 2, pp. 36–47, 2015.

[35] P. C. Arocena, B. Glavic, R. Ciucanu, and R. J. Miller, "The iBench Integration Metadata Generator," *PVLDB*, vol. 9, no. 3, pp. 108–119, 2015.

[36] Z. Abedjan, L. Golab, and F. Naumann, "Profiling Relational Data: A Survey," *VLDB Journal*, vol. 24, no. 4, pp. 557–581, 2015.

[37] Y. Ioannidis, "The History of Histograms (abridged)," in *Proceedings of the International Conference on Very Large Databases (VLDB)*, Berlin, Germany, 2003, pp. 19–30.

[38] J. Liu, J. Li, C. Liu, and Y. Chen, "Discover Dependencies from Data - A Review," *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, vol. 24, no. 2, pp. 251–264, 2012.

[39] M. Milani, Z. Zheng, and F. Chiang, "CurrentClean: Spatio-Temporal Cleaning of Stale Data," in *Proceedings of the International Conference on Data Engineering (ICDE)*. IEEE, 2019, pp. 172–183.