Multimodal Technologies and Interaction

Article

A Record Linkage-Based Data Deduplication Framework with DataCleaner Extension

Otmane Azeroual [1,*], Meena Jha [2], Anastasija Nikiforova [3,4], Kewei Sha [5], Mohammad Alsmirat [6,7] and Sanjay Jha [2]

[1] German Centre for Higher Education Research and Science Studies (DZHW), Schützenstraße 6A, 10117 Berlin, Germany
[2] School of Engineering and Technology, Central Queensland University, Sydney, NSW 2000, Australia; m.jha@cqu.edu.au (M.J.); s.jha@cqu.edu.au (S.J.)
[3] Institute of Computer Science, University of Tartu, Narva mnt 18, 51009 Tartu, Estonia; nikiforova.anastasija@gmail.com
[4] European Open Science Cloud (EOSC) Task Force "FAIR Metrics and Data Quality", 1050 Brussels, Belgium
[5] College of Science and Engineering, University of Houston Clear Lake, 2700 Bay Area Blvd, Houston, TX 77058, USA; sha@uhcl.edu
[6] Department of Computer Science, University of Sharjah, University City Rd., Sharjah 27272, United Arab Emirates; masmirat@just.edu.jo
[7] Department of Computer Science, Jordan University of Science and Technology, Irbid 22110, Jordan
* Correspondence: azeroual@dzhw.eu; Tel.: +49-30-2064177-38

Abstract: The data management process is characterised by a set of tasks where data quality management (DQM) is one of the core components. Data quality, however, is a multidimensional concept, where the nature of the data quality issues is very diverse. One of the most widely anticipated data quality challenges, which becomes particularly vital when data come from multiple data sources which is a typical situation in the current data-driven world, is duplicates or non-uniqueness. Even more, duplicates were recognised to be one of the key domain-specific data quality dimensions in the context of the Internet of Things (IoT) application domains, where smart grids and health dominate most. Duplicate data lead to inaccurate analyses, leading to wrong decisions, negatively affect data-driven and/or data processing activities such as the development of models, forecasts, simulations, have a negative impact on customer service, risk and crisis management, service personalisation in terms of both their accuracy and trustworthiness, decrease user adoption and satisfaction, etc. The process of determination and elimination of duplicates is known as deduplication, while the process of finding duplicates in one or more databases that refer to the same entities is known as Record Linkage. To find the duplicates, the data sets are compared with each other using similarity functions that are usually used to compare two input strings to find similarities between them, which requires quadratic time complexity. To defuse the quadratic complexity of the problem, especially in large data sources, record linkage methods, such as blocking and sorted neighbourhood, are used. In this paper, we propose a six-step record linkage deduplication framework. The operation of the framework is demonstrated on a simplified example of research data artifacts, such as publications, research projects and others of the real-world research institution representing Research Information Systems (RIS) domain. To make the proposed framework usable we integrated it into a tool that is already used in practice, by developing a prototype of an extension for the well-known DataCleaner. The framework detects and visualises duplicates thereby identifying and providing the user with identified redundancies in a user-friendly manner allowing their further elimination. By removing the redundancies, the quality of the data is improved therefore improving analyses and decision-making. This study makes a call for other researchers to take a step towards the "golden record" that can be achieved when all data quality issues are recognised and resolved, thus moving towards absolute data quality.

check for updates

Citation: Azeroual, O.; Jha, M.; Nikiforova, A.; Sha, K.; Alsmirat, M.; Jha, S. A Record Linkage-Based Data Deduplication Framework with DataCleaner Extension. *Multimodal Technol. Interact.* **2022**, *6*, 27. https://doi.org/10.3390/mti6040027

Academic Editors: J Dinesh Peter, Yu-Dong Zhang and Steven L. Fernandes

Received: 19 January 2022
Accepted: 7 April 2022
Published: 11 April 2022

Keywords: data quality; data linkage; data matching; duplicate detection; deduplication; research data; RIS; similarity; blocking; sorted neighbourhood; graphs; modelling; simulation; uniqueness

1. Introduction

The detection of duplicates in data has been a widely discussed and extensively researched topic [1–3], which popularity does not decrease, particularly in the light of the development and an increasing number and variety of advances in Industry 4.0, the increasing number of Cyber-Physical Systems (CPS) and actors representing them, Internet of Things (IoT) and Industrial IoT (IIoT) applications, etc. Data uniqueness is one of the central, although not the only, data quality issues and, respectively, criteria/metric to be verified to consider data as qualitative (together with data completeness, accuracy, currency and consistency, etc.). It also has proved to be one of the most commonly accepted data quality dimensions [4], which is also seen as the first research object in the context of data quality, studies of which were first carried out by statisticians in the late 1960s. This is the topic of interest for both theoreticians and practitioners, where the area of application may vary from IoT applications, where the data come from different heterogeneous data sources and data duplicates may cause significant costs for transmission, handling and storing costs, to databases of research information systems (RIS), data warehouses, where identification and elimination of duplicates is part of the ETL (Extract–Transform–Load) process, data lakes and data lakehouses preparing data for data science, machine learning and business analytics projects. This is also the case for more specific use cases such as system simulation and modelling (SSM), where duplicates among other data quality problems can negatively affect both engineering and operational levels of SSMs themselves, as well as the whole ecosystem in which respective solutions (algorithms, models, etc.) are further used. As regards the IoT, the term refers to a large number of heterogeneous “smart objects” (“things” in the “Internet of Things”) connected to the Internet, with applications and services that use data from these objects to create interactions [5]. According to Miorandi [6], the term “smart object” stands for entities that: (a) have a physical embodiment and a set of associated physical characteristics such as size, shape, etc., (b) have a minimal set of communication functions, such as the ability to be discovered and to accept incoming messages and respond to them, (c) have a unique identifier, (d) are associated with at least one name and one address, where the name is a human-readable description of the object and can be used for reasoning purposes, while the address refers to a machine-readable string that can be used to communicate with the object, (e) have some basic computing capabilities, which can range from the ability to match an incoming message with a specific footprint and ending with the ability to perform complex computations, including discovery of services and network management tasks, and (f) may be a means to sense physical phenomena such as temperature, light, electromagnetic radiation level, or to trigger actions that affect on physical reality/actuators. The IoT has an increasing number of applications, including manufacturing, medicine, production, disaster management, retail sector, smart cities and other areas [7–10], where IoT-based technologies aim at enabling organisations and individuals to make better-informed decisions, be more productive and improve health and quality of life [11]. However, an unavoidable prerequisite for this, as for all data-driven artifacts (services, products, etc.) despite their complexity and domain, is the quality of these data.

Duplicates occur for many reasons that can be traced to processes of data integration in the database, human enforced error due to manual data entry or different sources collecting different sets of data on one and the same object or event, etc. [12]. This, however, may have a negative impact on analyses, decision-making, service personalisation, user adoption and satisfaction, customer service, etc. Even in the scientific community and in the case of research data and RIS this issue may affect evaluations and outcomes leading to incorrect strategic decisions leading to far-reaching dangerous consequences. For these reasons, the detection of duplicates is an important measure of data quality assurance.

Duplicates can be detected by periodically inspecting an already existing database and identifying them, or immediately, i.e., when new data are recorded by comparing the new record online with the existing data records. In the latter case, the user is provided with a list of possible/potential duplicates while inserting a new data entry. The user can then decide to create a new data record or use the existing data record. Existing internal and external data sources typically have a batch run/job, in which duplicates are recognised and processed as automatically as possible, which makes it possible to ensure these checks for not only human-inserted data but also the data generated by sensors and transmitted to the main database, where it should be allocated with the data received from other sources. The aim is to keep user interference as little as possible. Data linking can also help search for duplicates between two different sources. This can lead to not only resolving the issue of duplicates by eliminating duplicate records but also to data enrichment by merging respective data entries referring to one and the same data object, supplementing the original record with more detail obtained from another entry (if the level of details of these records is different) thereby contributing to their completeness. An example of this could be a comparison of publication data from PubMed′s scientists with the data from the Scopus database.

There are several algorithms to address and resolve duplicates, where probably the most popular one is the data similarity-based approach [13]. It identifies the data similarity to decide whether the data objects being considered are two unique objects or duplicates [14]. This is particularly important in the context of data integration and data cleaning. What is important is that the duplicate detection procedure should primarily be effective and accurate to produce high-quality data [15]. The aim is to find all possible duplicates and eliminate them. The degree of similarity chosen, and the decision-making method play a decisive role in identifying duplicates. According to Elmagarmid, Ipeirotis, and Verykios [16], to measure the effectiveness of the duplicate detection method, duplicate assignments must be known. However, there are hardly any freely available sources for which the assignments, known as the golden standard [16]. This is because the determination of double assignments is at least partly manual, resulting in a complex process for identifying them. Determining this mapping for a larger data set is very resource-intensive, taking up to hundreds of man-hours.

The detection and prevention of duplicates should be a high priority in an organisation’s (including universities or research institutions) data processing and management strategy. Duplicate checks can significantly increase the quality of the information system and its outputs. This saves storage space and increases performance because queries are processed more quickly because fewer records need to be processed. Furthermore, what is more important is that duplicates can falsify analyses and incorrect and incomplete data that can potentially cause great damage. Due to such corruption or data quality problems, an authority may make poor strategic decisions.

Therefore, this paper stresses the relevance and importance of this topic, as duplicate entries in data of different types. The use case proposed refers to the RIS, where the issue of duplicates lead to inefficient data management having a negative effect on a number of processes in scientific institutions and giving rise to a variety of negative outcomes at various levels [15,17]. Recent studies emphasise the importance of this problem in the healthcare and (bio)medical domains [18].

This paper proposes a prototype of a solution that allows the user to perform efficient duplicate determination with their further elimination/cleaning. It is based on six steps: (1) data preparation referring to the standardisation of data, (2) search area definition and reduction of the number of comparisons that should be carried out (referring to blocking and sorted neighbourhood methods), (3) matching attribute values by means of similarity functions, (4) a decision model, which is based on the similarity vector, (5) clustering duplicates, which are then (6) verified based on the parameters of both precision and recall.

The solution presented is intended to detect duplicates for medium to large databases. The focus is placed on the simple integration and configuration of duplicate detection so that the solution can be easily adapted to different users, including those without domain knowledge. The Record Linkage method integrates different data sources and identifies records pertaining to the same object [19]. There are many indexing techniques available to deduplicate large databases. However, they are not free from their overheads. For example, the q-gram-based indexing technique is very slow and not suitable for removing duplicates from large databases [20]. Christen [13], has investigated various indexing techniques that were developed for record linkage and deduplication and identified that there is a need for a proper definition of blocking keys as training data may not be available for true matches and true non-matches. The indexing techniques chosen and evaluated were: traditional blocking; Q-Gram-Based Indexing; Suffix Array-Based Indexing; Canopy Clustering; and String-Map-Based Indexing. The indexing methods are difficult to apply successfully in practice [13]. In this paper, we argue that a high level of efficiency with good recognition performance at the same time can be achieved by using the Record Linkage method, which would be in-built into one of the existing data quality management tools such as DataCleaner.

The contribution of the paper is threefold: (1) a six-step deduplication framework relying on the Record Linkage is proposed providing a detailed step-by-step description of the process, contributing to the general understanding of the deduplication and possible techniques, demonstrated on a real-world RIS and respective research database containing data on publications, research projects and others, (2) the framework is then developed into a working prototype of an extension for one of the most widely-used data quality DataCleaner tool, thereby providing users with a new opportunity, which is not available in the non-commercial version. This allows us to make the proposed framework usable by integrating it into a tool that is already used in practice rather than imposing new tools where each tool is intended for only one task. In addition, given that the deduplication process was carried out for the real-world research database, the study contributes to the research institution by performing data cleaning in it.

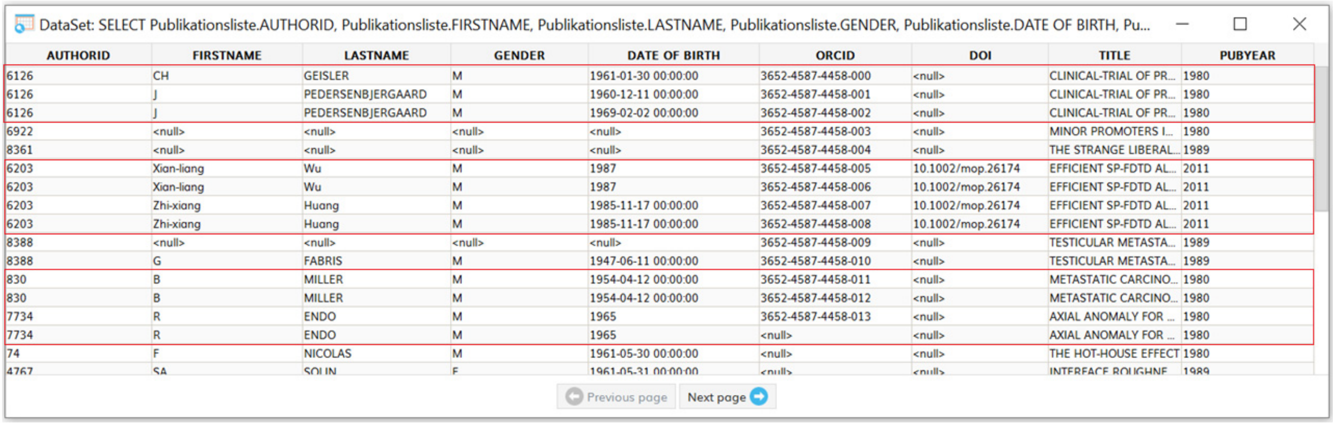
The remainder of the paper is structured as follows: Section 2 discusses the relevance of the topic of duplicate detection. Section 3 discusses the benefits of record linking to graphs, and presents the record linkage framework including an evaluation with real-world examples of research data used in the practices. Section 4 discusses deduplicate DataCleaner extension. Section 5 is focused on discussion and future work, and finally, the results are summarised in Section 6.

2. Problem Relevance and Classification

If a database has been used for some time, it reaches a certain size, which makes it unmanageable if the relevant support/aid is not provided. For example, a user entering new data does not know in advance whether the data already exist. In this case, a user might search for similar data records using an integrated search function. However, it is very time-consuming and may not be performed if the user assumes that the new data are not duplicates. A typical example is merging multiple data from different data sources. Similarly, duplicates occur when databases overlap. This is a typical case for many companies and research institutions. This is all the more so in the IoT, when data on the same data object can be collected by different “smart objects”, also known as “things” if there is no proper data integration procedure (i.e., no common object identifier is specified and each object saves and transmits data on the same real-world object by using the internal identifier assigned to it).

The detection of duplicates is related to the identification of different data records representing the same, unambiguous real-world objects in a database [21]. When data are entered manually, employees often create new data records, in the absence of identifying the existing record. There could be a possibility that the record already exists and by re-entering the same record it becomes duplicated. There could be many reasons for the duplicated records. For instance, even if there is some sort of check of whether such a record already exists, there may be different spelling used in the first and the current entries. Therefore, the user does not find it and created a duplicate data record. In some cases, the user just wanted to update data, such as the name of the scientist or the street and institution's address and zip code, and accidentally created a duplicate. Since many institutions do not check their data regularly, duplicates remain in the information system and cause problems for the institutions. If the institutions do not establish a regular duplicate check, the number of duplicate entries continues to rise. This high number of duplicates, however, reduces the quality of the data and follow-up activities, such as their processing, analysis and decision-making.

Figure 1 shows an example of duplicate records for a database of publications (retrieved from Web of Science), which we will later use for demonstration purposes, which collects data on the article, i.e., its title, DOI, year of publication, as well as the data on its author—ID, first name, last name, gender, birth date and ORCID. This database contains three sets of duplicates, as shown in the red boxes in Figure 1, which remained unnoticed in the past since duplicate checks were not conducted. More precisely, due to duplicate records, 3 publications appeared in 11 entries and are counted as 11 publications, contradicting the state of the actual publications.


DataSet: SELECT Publikationsliste.AUTHORID, Publikationsliste.FIRSTNAME, Publikationsliste.LASTNAME, Publikationsliste.GENDER, Publikationsliste.DATE OF BIRTH, Pu...
Previous page
Next page

AUTHORID	FIRSTNAME	LASTNAME	GENDER	DATE OF BIRTH	ORCID	DOI	TITLE	PUBYEAR
6126	CH	GEISLER	M	1961-01-30 00:00:00	3652-4587-4458-000	<null>	CLINICAL-TRIAL OF PR...	1980
6126	J	PEDERSENBJERGAARD	M	1960-12-11 00:00:00	3652-4587-4458-001	<null>	CLINICAL-TRIAL OF PR...	1980
6126	J	PEDERSENBJERGAARD	M	1969-02-02 00:00:00	3652-4587-4458-002	<null>	CLINICAL-TRIAL OF PR...	1980
6922	<null>	<null>	<null>	<null>	3652-4587-4458-003	<null>	MINOR PROMOTERS I...	1980
8361	<null>	<null>	<null>	<null>	3652-4587-4458-004	<null>	THE STRANGE LIBERAL...	1989
6203	Xian-liang	Wu	M	1987	3652-4587-4458-005	10.1002/mop.26174	EFFICIENT SP-FDTD AL...	2011
6203	Xian-liang	Wu	M	1987	3652-4587-4458-006	10.1002/mop.26174	EFFICIENT SP-FDTD AL...	2011
6203	Zhi-xiang	Huang	M	1985-11-17 00:00:00	3652-4587-4458-007	10.1002/mop.26174	EFFICIENT SP-FDTD AL...	2011
6203	Zhi-xiang	Huang	M	1985-11-17 00:00:00	3652-4587-4458-008	10.1002/mop.26174	EFFICIENT SP-FDTD AL...	2011
8388	<null>	<null>	<null>	<null>	3652-4587-4458-009	<null>	TESTICULAR METASTA...	1989
8388	G	FABRIS	M	1947-06-11 00:00:00	3652-4587-4458-010	<null>	TESTICULAR METASTA...	1989
830	B	MILLER	M	1954-04-12 00:00:00	3652-4587-4458-011	<null>	METASTATIC CARCINO...	1980
830	B	MILLER	M	1954-04-12 00:00:00	3652-4587-4458-012	<null>	METASTATIC CARCINO...	1980
7734	R	ENDO	M	1965	3652-4587-4458-013	<null>	AXIAL ANOMALY FOR ...	1980
7734	R	ENDO	M	1965	<null>	<null>	AXIAL ANOMALY FOR ...	1980
74	F	NICOLAS	M	1961-05-30 00:00:00	<null>	<null>	THE HOT-HOUSE EFFECT	1980
4767	SA	SOLIN	F	1961-05-31 00:00:00	<null>	<null>	INTERFACE ROUGHNE...	1989

Figure 1. Examples of duplicates.

The detection of exact duplicates, where the values of all attributes forming a record comply completely called "exact duplicate", is relatively trivial and can be easily implemented by simple sorting, while the detection of "fuzzy duplicates" is a complex process [2]. Record linkage is being used for the identification of duplicates.

One of the first definitions of "automatic record linkage" was presented in 1959 [22]. The detection of duplicates is part of the research area of information and data quality, where the quality is referred to as a "fitness for use" and is used in the data quality area [23]. The aim of duplicate detection is to increase the quality of the information in terms of freedom from errors and improve completeness. Freedom from errors is the conformity of the information with reality, which is improved by duplicate detection, as it reveals the redundant representations of a real-world object. The completeness, however, is improved because the duplicates found often contain different levels of detail on the real-world object, which together result in a more complete and comprehensive mapping of an object.

Poor quality of data and duplicates can harm the company in many ways [24–26]. Some of the examples of harming the economy are: a mail-order company regularly sends catalogs to customers where duplicates in the customer database cause unnecessary printing and shipping costs. What is more, customers who have received catalogs twice have doubts about the company′s quality management, which is damaging the image of the company. Another example is a trading company grants credit to its customers. Duplicates in a customer database can lead to a case when a customer uses a credit line several times. In the case of the customer′s bankruptcy, there is a risk of high debt losses. An additional comparison of customer data with data from the insolvency reports may prevent business with companies that are already insolvent. The most common use of duplicate checks is address databases. Address records created twice mean that mail is sent to the customer more than once. This multiple posting not only costs a company a lot of money but also creates a bad reputation among customers. Address duplicates may occur due to: (a) different address spellings, (b) typing errors, and (c) differently abbreviated names, street names or another salutation abbreviation. Therefore, address databases are checked for duplicates more often compared to other areas or even on a regular basis. The latest case helps them to find duplicates in the database, compare them and merge the data in case of duplicates into one database entry.

Duplicate or non-uniqueness is one of the central, although not the only, data quality issues and respectively criteria/metric to be verified to consider data as qualitative (together with data completeness, accuracy, currency and consistency, etc.). It is, however, of great economic importance. In order to recognise such duplicates in practice, we propose error-tolerant methods that deal with them, thereby helping users to identify duplicates without additional efforts needed from them.

3. Materials and Methods

Data quality problems, including duplicate problems, should be disclosed and discovered as soon as the data are added to the data store rather than later after processing. To avoid duplicates, binding guidelines should be set (in written form) specifying how the records and each particular attribute should be registered. In the case of the RIS and research data, this refers to the way in which publication titles, affiliations, author addresses, and other details depending on the case, should be recorded. It is particularly recommended to clean up duplicates using IT technology that would allow the results to be displayed graphically so that identified errors can be easily and quickly observed and understood. This is particularly important in our case because our aim is to propose a universal approach that would be suitable for different users independently from their knowledge and area of expertise, where visualisation can provide benefits by acting as a communication tool that simplifies and improves the identification of the underlying issues.

The tool should be particularly useful for large data sets or when data from different existing systems are merged. With the help of such advanced tools, the status of the data is recorded and carefully documented. It determines not only the errors but also the frequency of errors, i.e., error rates, in graphical representations and provides information on whether the supplied data meet the quality requirements. At this stage, duplicate data should be easily identified. In addition, incomplete data records and contradicting/conflicting casualties are discovered and incorrect data records are identified. At the end of this first inventory, there is usually an understanding that 70% to 95% of all data are correct. Suitable measures are needed, such as data cleansing, to define and eliminate the errors [18].

The detection of duplicates with graphic support is typically performed by a dashboard for data control and optimisation, which, according to our experience, provides many advantages, such as (a) presentation, analysis, and control of the results after diagnoses, (b) delivery of important insights, (c) intuitively understandable data visualisations, (d) traceability by the means of the clear warnings in the case of insufficient data quality, (e) development of a narrative around the data, (f) improved communication since visual data are easier to comprehend and process compared to any other form of communication, (g) accelerated decision-making processes, (h) optimising the success of institutions.

Record Linkage Framework

Before data from internal and external data sources can be integrated into a database for their further use, duplicate detection should be carried out.

All data records should be checked according to predefined rules and corrected (if needed). Then, the cleaned and consolidated data are merged into “golden records” that are defined as "a single, well-defined version of all the data entities in an organisational ecosystem providing single version of the truth, where "truth" is understood to mean the reference to which data users can turn when they want to ensure that they have the correct version of a piece of information" [27]. This golden record is a master data record that combines the relevant attributes from all data sources. The resulting golden records form the basis for further data analysis or data migration, as shown in Figure 2.

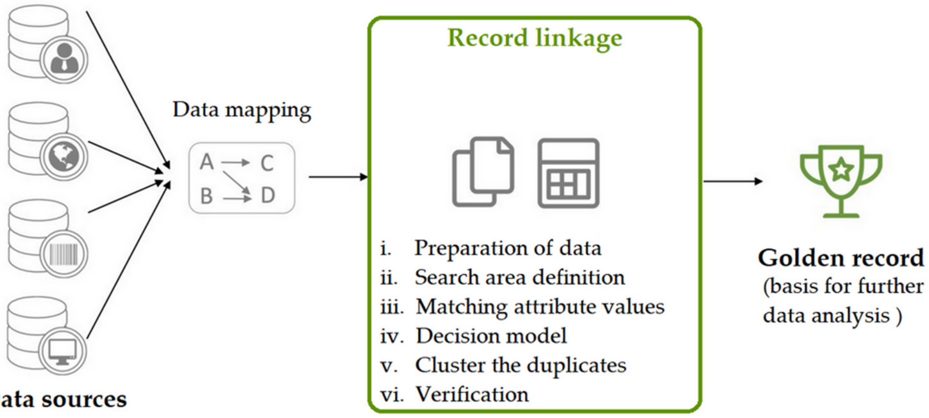


Figure 2. Framework for using Record Linkage.

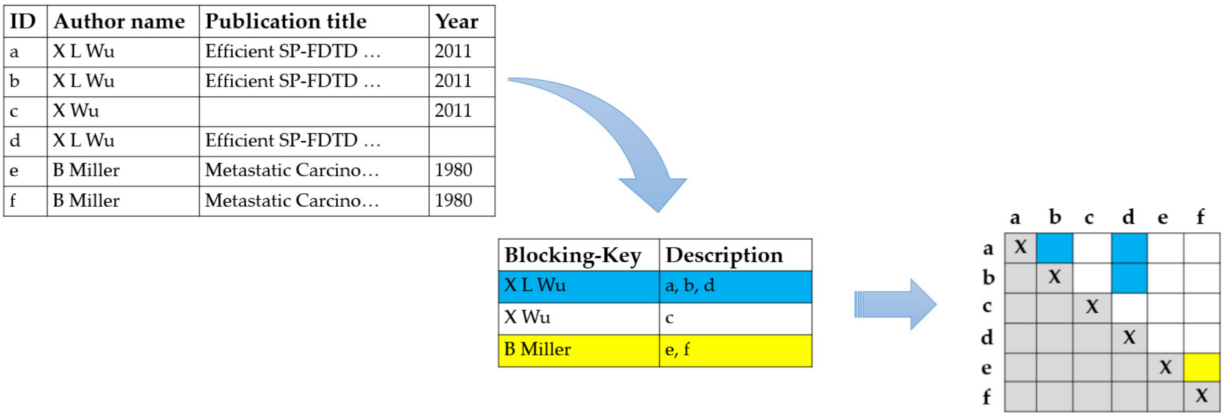
We divide the process of duplicate detection into the six steps as described below, where for the sake of simplicity, thus ensuring better understandability of the proposed framework, and the topicality of the covered issue for the internal research institution, we demonstrate the framework on the real-world example of the RIS and research data. More detail on the testing settings and data constituting the database under testing will be provided in the next section. Although we do not treat this case as an example of the IoT, the above definition can be compared to it, given that data of different structures come to the single data storage from different data sources (where it is a subject of processing, calculations, etc.) and then affects further data processing and sometimes triggers further actions and decisions.

Step 1 Prepare the Data: Firstly, data is brought in a standardised form so that the duplicates can be identified more easily. This standardisation may vary widely depending on the domain concerned. For instance, when processing data as strings, all uppercase letters are often transformed into lowercase letters, e.g., both "year of publication", "Year of Publication", "YEAR OF PUBLICATION", etc. would be transformed to "year of publication". In general, the goal is to replace possible different spelling with a single and uniform form. If there is enough domain knowledge, synonyms, and acronyms, as well as abbreviations, can be dealt with. As an example, in the IT area, these can be the use of both "AI" for "Artificial Intelligence", "ML" for "Machine Learning" or even more exotic examples such as "Big Data and Cognitive Computing", "B.D. Cogn. Comput." ⇒ "bdcc", while in biomedicine, "omics" or "-omics" stands for biological sciences that end with -omics, i.e., genomics, transcriptomics, proteomics, or metabolomics. The nature of these examples depends heavily on the domain. There may be other standardisations for other types of fields. One of the most widely used examples is dates that should be brought in a uniform format, e.g., 24/9/20, September 24, 2020, ⇒ 09/24/2020. The choice of a specific format may depend on the localisation pattern if the data are more likely to be used internally within the organisation and/or within the country, or a specific commonly accepted standard such as ISO. The first case poses a risk of making it more difficult to integrate and exchange data with other systems if each system makes a choice according to its own vision, but the latter case is a better choice to make these data more uniform when dealing with other institutions and/or national partners. Overall, the more domain knowledge is, the more data can be standardised. Much of what is being used here is analogous to what happens before an index for information retrieval is created. Standardisation has advanced significantly in recent times.

Step 2 Search Area Definition: The next stage is devoted to the definition of the search area. Let us imagine that we have two data records, namely publication A and publication B, that should be checked for duplicates. As mentioned above, the number of comparisons to be made correspond to [$publication\ |A| \times publication\ |B|$]. The task of the search area definition is to reduce the number of comparisons, without making further steps, thereby limiting the scope and minimising resources to be spent. Two methods to achieve this are Blocking and the Sorted Neighbourhood Method (SNM).

Blocking is a method according in which the search area is divided into blocks where duplicate detection is then carried out within these blocks. Splitting into blocks can take place in different ways. One option is to generate or use a block key, where all tuples that receive the same block key are combined in a block for processing. This can also be performed by using a hash function. Blocking techniques to reduce the computational complexity associated with record linkage have widely been used [28].

Figure 3 shows an example of a standard blocking procedure demonstrated in a simple example, in which we dealt with a table containing data about the author, the title of the publication and the year of its publishing, where a unique identifier ID was assigned to each record. With standard blocking, each description is assigned to exactly one block based on its blocking key.



ID	Author name	Publication title	Year
a	X L Wu	Efficient SP-FDTD …	2011
b	X L Wu	Efficient SP-FDTD …	2011
c	X Wu		2011
d	X L Wu	Efficient SP-FDTD …	
e	B Miller	Metastatic Carcino…	1980
f	B Miller	Metastatic Carcino…	1980

Blocking-Key	Description
X L Wu	a, b, d
X Wu	c
B Miller	e, f

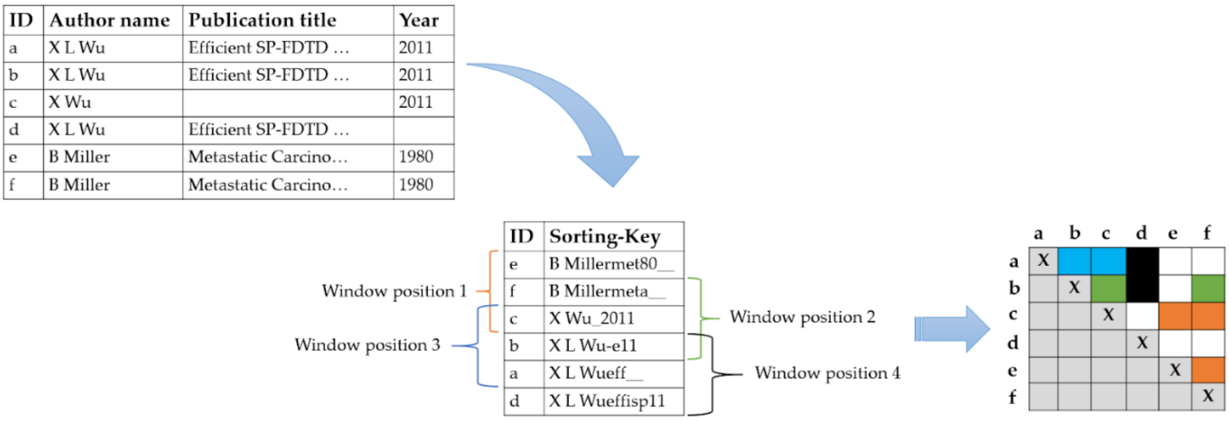
	a	b	c	d	e	f
a	X	🟦		🟦		
b	▨	X		🟦		
c	▨	▨	X			
d	▨	▨	▨	X		
e	▨	▨	▨	▨	X	🟨
f	▨	▨	▨	▨	▨	X

Figure 3. Standard blocking procedure.

All "descriptions" that are formed by a list of IDs (refer to the second column) with the same key are in the same block. In the example, author names form the blocking key. As a result, the descriptions a, b, and d are assigned to the first block and e, and f are assigned to the third block. Comparisons are made with colour-coded description pairs. Calculations for the pairs marked in white save the standard blocking. Grey fields stand for redundant and self-comparisons. Description c is assigned to a separate block due to the incorrect name of the author and can no longer be compared with the actual matching descriptions a, and b during the remainder of the duplicate detection process.

Sorted Neighbourhood is a method in which the tuples are sorted according to a suitable key. Then a fixed-size window is moved over the tuples and only the tuples in the window are considered for comparison. With this method, the choice of the key is important; it must ensure that duplicates are close together according to the order of the key. The better the key, the smaller the window can be. The sorted neighbourhood method can also be used to demonstrate how, by adaptively changing the size of a fixed window, accuracy and performance can be improved [29].

Figure 4 shows an example where the sort-key is formed by the three-character prefixes of the author's name, publication title and publication year attributes. The window size is $w = 3$. The window is pushed over the sorted keys. All pairs of entity descriptions whose keys fall into the common position of the sliding window form the so-called candidate pairs. The elements *e*, *f*, and *c* form a block for the first position of the sliding window. *f*, *c*, and *b* fall into one block for the second position of the window.



ID	Author name	Publication title	Year
a	X L Wu	Efficient SP-FDTD …	2011
b	X L Wu	Efficient SP-FDTD …	2011
c	X Wu		2011
d	X L Wu	Efficient SP-FDTD …	
e	B Miller	Metastatic Carcino…	1980
f	B Miller	Metastatic Carcino…	1980

ID	Sorting-Key
e	B Millermet80__
f	B Millermeta__
c	X Wu_2011
b	X L Wu-e11
a	X L Wueff__
d	X L Wueffisp11

	a	b	c	d	e	f
a	X	🟦	🟦	⬛		
b	▨	X	🟩	⬛		🟩
c	▨	▨	X		🟧	🟧
d	▨	▨	▨	X		
e	▨	▨	▨	▨	X	🟧
f	▨	▨	▨	▨	▨	X

Figure 4. Sorted neighborhood procedure.

Each time the window is moved, the newly added candidate pairs are marked with the colour of the respective window position. Unlike the standard blocking, incorrect description *c* can be compared to the matching elements *a* and *b* in sorted neighbourhood blocking. Several description pairs appear in multiple blocks, which would lead to unnecessary redundant comparisons.

Step 3 Matching Attribute Values: The values of the individual attributes of the comparable tuples are compared with similarity functions. These similarity functions return a similarity of two values A and B from $s = sim\ (A, B)$. Depending on the type of attribute value, the similarity function can vary. For numbers, a normalised form of the difference could be the most appropriate option. For dates, it could be the number of days/hours/minutes in between, while for coordinates—the distance would make sense. When matching the attribute values of tuples *a* and *b*, the tuples are vectors of their attributes:

$$\begin{aligned} a &= (a_1, a_2, \ldots, a_i), \\ b &= (b_1, b_2, \ldots, b_i), \end{aligned} \tag{1}$$

i—number of attributes.

An individual similarity function $sim_n\ (a_n, b_n)$ is applied to each attribute pair $(a_\mathrm{i}, b_\mathrm{i})$. This similarity function may consist of several similarity functions, the results of which are properly combined/aggregated. These calculations result in a similarity vector for a pair of tuples, which contains a similarity value per attribute. The similarity vector that is the input for the subsequent decision model is:

$$sim_{ab} = (sim_1\ (a_1, b_1), sim_2\ (a_2, b_2), \ldots, sim_i\ (a_i, b_i)) = (sim_1, sim_2, \ldots, sim_i) \tag{2}$$

In the iterative record linkage process, it is possible to link blocking and matching. New attribute values from a merge-based entity matching step can be used to change the block assignment of the descriptions. The changed blocks lead to new comparisons at the matching step. The blocking and matching are repeated alternately until the blocks cease to change or another termination criterion is fulfilled [30]. Figure 5 shows the similarity graph output by entity matching for the research data, resulting from using sorted neighbourhood blocking and a calculation of weighted similarity, according to the example in Figure 1.

Step 4 Decision Model: Based on the similarity vector, a decision model is used to decide whether compared tuples are duplicates or not. A decision model is a function that assigns a matching type. We distinguish three matching types. One of the values (*M*, ***P***, *U*) for a pair of tuples (t_1, t_2), where M stands for matching tuples, $\boldsymbol{P}$—for possible matching tuples, and U—for non-matching tuples is assigned. In other words:

$$\mu(t_1, t_2) \in \{M, \boldsymbol{P}, U\} \tag{3}$$

There are many ways in which such a decision model can be designed. The two most common options for the general decision model variants in the context of research data are: (1) domain knowledge-based decision, (2) probability-based decision.

The domain knowledge-based decision model depends on the domain expert. It defines the conditions and rules for considering two tuples as duplicates. These rules work with the similarity vector values and assign a matching type to a pair of tuples. Such a rule could be, "two records are considered duplicates, if the authors' names match for more than 80% and the publication's title matches for more than 60%". This rule is defined as:

$$\textit{IF Author names} > \textit{0.8 AND publication title} > \textit{0.6 THEN MATCHING TYPE match} \tag{4}$$

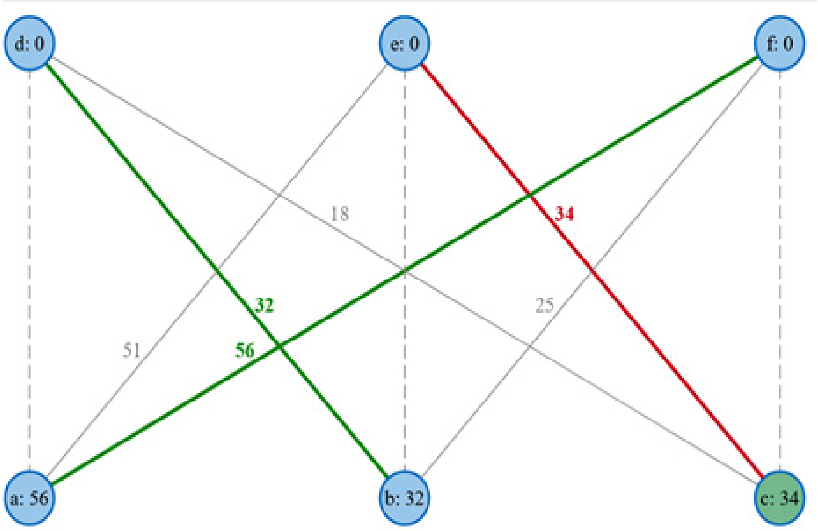


Figure 5. Similarity graph (bipartite).

While the rate assigned to "Author names" might seem questionable, it is particularly important because of the specificities of different languages and the countries to which the author belongs. More precisely, there are languages that use so-called diacritic marks, so their surnames can be written differently depending on the language. One of the authors of this paper could serve as an example, namely, the given surname, written in Latvian, is "Ņikiforova" although in non-Latvian and, for instance, the surname form used for scientific publications is "Nikiforova" ("Ņ" vs. "N"). Therefore, if the rule would not include a part on the allowable partial matching of two records, depending on the database and the article, two identical articles could not be considered duplicates, although they are such.

In addition, this rule may depend heavily on the expert′s knowledge of the quality of the database. It is therefore conceivable that a general statement can be made on which attributes have a higher value, a higher discriminant for the duplicate detection. However, in the knowledge-based rules, it is crucial that the relevant domain knowledge is expressed by combining the limit values and attributes used in the rules. Finally, either the final limit value is used to decide whether a pair of tuples are assigned M or U, or a matching type is assigned directly by a rule [31]. Figure 6 shows matching type varies, depending on the case, either M or U.

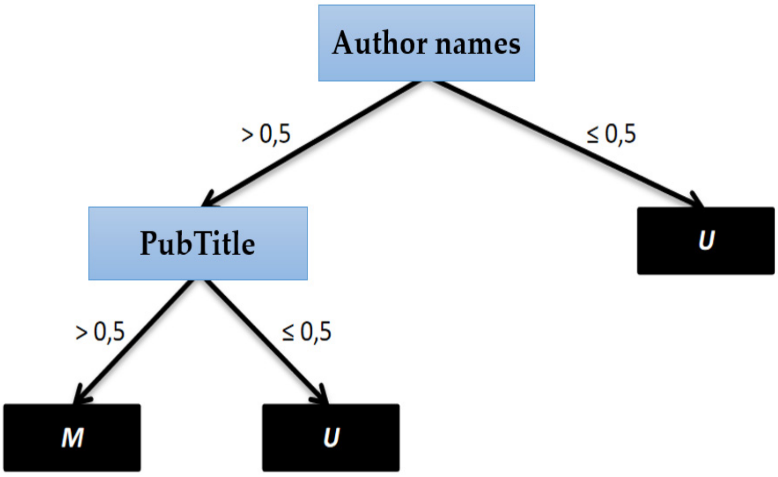


Figure 6. Decision tree for record linkage.

A probability-based decision model was originally presented by Fellegi and Sunter [32] in 1969 and has since been expanded and improved by Batini and Scannapieco [33], Panse et al. [31]. This model initially determines two conditional probabilities using a similarity vector. According to this model, the probability that the pair of tuples under consideration is a duplicate such as:

$$m(c) = \boldsymbol{P}(c \mid (t_1, t_2) \in M) \tag{5}$$

At the same time, the probability that they are not duplicates is:

$$u(c) = \boldsymbol{P}(c \mid (t_1, t_2) \in U) \tag{6}$$

It is not specified how functions u and m determine probability. Generally, they are weighted and normalised combinations of similarity vector values. The resulting conditional probabilities are then put in the relation and a new value, the matching weight $\boldsymbol{R}$, is determined as:

$$\boldsymbol{R} = m(c)/u(c) \tag{7}$$

With two limit values T_y and T_z, the tuple of a matching type M, $\boldsymbol{P}$, U is assigned to the tuple based on the matching weight. More precisely:

$$\begin{gathered} M\text{: } if\ \boldsymbol{R} > T_y \\ \boldsymbol{P}\text{: } if\ T_y \leq \boldsymbol{R} \leq T_z \\ U\text{: } if\ \boldsymbol{R} < T_z \end{gathered} \tag{8}$$

It becomes clear that the T_y and T_z limit values are essential to this model. There are different ways to determine this. Limit values can be set and optimised using appropriate training data. However, machine learning (ML) methods can also be used. Pairs of tuples that were declared as possible matches are then manually classified as match or non-match by a domain expert. Then, depending on the application, duplicates are further processed in an appropriate manner.

Step 5 Cluster the duplicates: In most use cases, finding duplicate pairs is not enough. The goal is to find all the tuples that represent real-world objects. Thus, the duplicates found must be combined into clusters. This clustering is usually implemented using certain forms of transitivity. A very simple way is to create a transit envelope over duplicates [2]. Finally, recognised clusters are properly merged/fused or the decision on the remaining tuples and those to be deleted are made. In Figure 7, we provide the code that we used to group duplicates by means of a transitory envelope.

```

public class TransitiveClustering extends AbstractConfigurable
implements Clustering {

public Collection<XTuple> cluster(TupleSearcher searcher, TuplePair
pair) {
if (this.finished.contains(pair.getValue1().getTid())) {
log.debug("XTuple " + pair.getValue1().getTid() + " allready
processed, skip it");
return null;
 }
final Queue<XTuple> queue = new LinkedList<XTuple>();
final Set<XTuple> duplicates = new HashSet<XTuple>();

queue.add(pair.getValue1());
XTuple t1 = null;
while ((t1 = queue.poll()) != null) {
if (!this.finished.contains(t1.getTid())) {
final List<TuplePair> pairs = searcher.
findPairsByMatchingTypeAndTID(
MatchingType.Match, t1.getTid());
log.debug("Pairs found for tid: " + t1.getTid() + "- Num: "+ pairs.
    size());
duplicates.add(t1);
this.finished.add(t1.getTid());
for (TuplePair tuplePair : pairs) {
final XTuple t2 = tuplePair.getNot(t1);
if (!duplicates.contains(t2)) {
log.debug("add to queue: " + t2.getTid());
queue.add(t2);
    }
  }
 }
}
log.debug("Duplicates: " + duplicates.size());
return duplicates;
 }
}
```

Figure 7. Transitive clustering code.

Step 6 Verification: The detection of duplicates is measured and evaluated in accordance with appropriate parameters to allow optimisation of the model selected. Two important parameters when assessing duplicate detection are (a) *precision* and (b) *recall* [34]. However, to define these parameters, other values such as false-non-match and false-match error rates are required. A tuple pair that is not a duplicate but was declared as a duplicate by the selected model is called a false match. Contrary to this, false-non-match denotes a pair of tuples, which is a duplicate but not recognised by the mode as such:

	Duplicates	**Not a Duplicate**
match	true-match	false-match
non-match	false-non-match	true-non-match

Now, the precision $\boldsymbol{P}$ can be defined as the ratio between the total number of duplicates found and pairs of tuples incorrectly declared as duplicates:

$$\boldsymbol{P} = |true - match|/(|true - match| + |false - match|) \tag{9}$$

The recall $\boldsymbol{R}$, however, represents the ratio between the found duplicates and the total number of existing duplicates. When a recall of 100% is achieved, it means that all pairs are declared as duplicates. Such a ratio, however, is considered too trivial and should be avoided.

$$\boldsymbol{R} = |true - match|/(|true - match| + |false - non - match|) \tag{10}$$

The precision and recall of two parameters have a mutual impact on each other; it is important to find an appropriate balance between them by optimising the model. In general, it can also be said that, as a rule, an increase in the range of the possible match reduces the number of false-non-matches and false-matches [33]. In return, the number of possible matches to be examined by an expert is increasing, which is also not desirable.

In the following section, a practical implementation of the idea set out in this section is discussed, extending the existing and widely used data quality tool.

4. Results

Deduplicate DataCleaner Extension

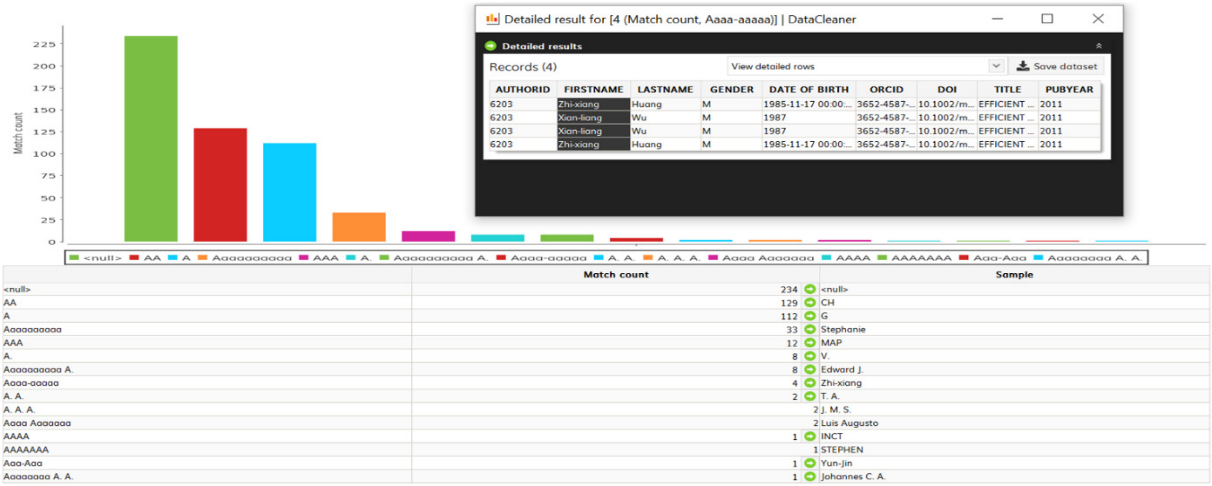
Data duplicates can be recognised by means of the use of sophisticated processes, algorithms and the human eye. However, people's capabilities to work on this are very limited. Manual methods fail with data larger than 5000 data records. This is what we experienced for the current system in use. Then, machine and algorithmic processes are required. Excel can be considered as an option but duplicate searches with Excel quickly reach their limits [18].

Duplicates recognition could be achieved using the framework proposed in the previous section. However, to make the proposed framework usable, it is preferable to integrate it into a tool that is already used in practice, rather than imposing a set of new tools where each tool is intended for only one task, i.e., by providing a tool that will only be used for deduplication and will be used in addition to other tools used. Therefore, we developed an extension for DataCleaner [https://datacleaner.org/ (accessed on 15 January 2022)], which is considered one of the most popular open-source platforms for data quality analysis. Although it provides a duplicate detection opportunity, this function is available only in the commercial version, where duplicate detection works only with raw data. However, if data are dirty, the tool suggests standardising the data before searching for duplicates. This means that deduplication is not user-friendly and could not be used by many and requires improvement. We argue that this could be achieved by following the procedure that we defined in the previous section.

Therefore, within this study, we developed the extension for DataCleaner as a powerful data quality tool to allow duplicate data to be found automatically, quickly, and easily, in more than thousands of data entries. Such an extension allowed to increase the overall quality of the deduplicate process and thus emphasing the suitability of the tool for concerned purposes. The extension is available in an open repository to be both available and accessible to any stakeholder [https://github.com/OtmaneAzeroualDZHW/Record-Linkage-Based-Data-Deduplication-Framework (accessed on 15 January 2022)].

To validate the effectiveness of the developed extension and proposed framework in general, as well as its suitability for use in the real world, we used it for analysing the system we are dealing with. The system under test is a database of the German Centre for Higher Education Research and Science Studies (DZHW), where scientific publications produced by academic and scientific staff are collected and further processed for multiple purposes, together with the data on the research project mostly inserted manually by research institutions' employee, combined with automated data retrieval from other external research institutions' information systems, complemented by data entries manually performed by researchers maintaining the completeness of the data on them and their research activity. Currently, the system stores 5000+ records. When DataCleaner, extended according to the proposed framework, was applied to this system, it has managed to easily recognise 660 duplicate records with 545 records automatically corrected within a minute.

Figure 8 shows an example of automatic duplicate detection. In the first step, the various algorithms used are optimally configured for the specific data situation. In addition, the allowed error rate is set. What is important here is that the area of application impacts the setting. It makes a difference whether duplicates are sought in the addresses of the institutions or in the titles of the author's publication. In the first case, some errors are allowed. In the second case, no data records may be recognised as duplicates if there is no firm assurance/clarity that they belong to the same person (author) (zero error tolerance). This setting is important because automatic duplicate detection works with statistical methods and there is uncertainty. The process consists of different steps, in which the group formation is a core. This way, the process can identify duplicate data.


Match count
225
200
175
150
125
100
75
50
25
0
🟩 <null> 🟥 AA 🟦 A 🟧 Aaaaaaaaaa 🟪 AAA 🟦 A. 🟩 Aaaaaaaaaa A. 🟥 Aaaa-aaaaa 🟦 A. A. 🟧 A. A. A. 🟪 Aaaa Aaaaaaa 🟦 AAAA 🟩 AAAAAAA 🟥 Aaa-Aaa 🟦 Aaaaaaaa A. A.
Detailed result for [4 (Match count, Aaaa-aaaaa)] | DataCleaner
Detailed results
View detailed rows
Save dataset

Records (4)

AUTHORID	FIRSTNAME	LASTNAME	GENDER	DATE OF BIRTH	ORCID	DOI	TITLE	PUBYEAR
6203	Zhi-xiang	Huang	M	1985-11-17 00:00:...	3652-4587-...	10.1002/m...	EFFICIENT ...	2011
6203	Xian-liang	Wu	M	1987	3652-4587-...	10.1002/m...	EFFICIENT ...	2011
6203	Xian-liang	Wu	M	1987	3652-4587-...	10.1002/m...	EFFICIENT ...	2011
6203	Zhi-xiang	Huang	M	1985-11-17 00:00:...	3652-4587-...	10.1002/m...	EFFICIENT ...	2011

	Match count	Sample
<null>	234	<null>
AA	129	CH
A	112	G
Aaaaaaaaaa	33	Stephanie
AAA	12	MAP
A.	8	V.
Aaaaaaaaaa A.	8	Edward J.
Aaaa-aaaaa	4	Zhi-xiang
A. A.	2	T. A.
A. A. A.	2	J. M. S.
Aaaa Aaaaaaa	2	Luis Augusto
AAAA	1	INCT
AAAAAAA	1	STEPHEN
Aaa-Aaa	1	Yun-Jin
Aaaaaaaa A. A.	1	Johannes C. A.

Figure 8. Duplicate detection of more than hundreds of thousands of publication data with DataCleaner.

The result of duplicate detection is a list of unique items that contain all the author's data records identified as unique. There are a variety of options recognised by the staff (data operators etc.) as duplicates or unique items according to the following criteria. Some data errors are assessed more reliably by the user based on plausibility considerations than by an algorithm. Misspelled institution names can be recognised with the human eye or by comparing the compliance of all institution names with an external authoritative dataset, which has the list of such institutions, i.e., registry. In addition, the user can immediately recognise the unstructured values of database entries as equivalent. Such entries can be reliably assessed with DataCleaner. Research data analysis with DataCleaner can take place in real-time and can be integrated into the (research) information systems. The solution also shows its strengths when it comes to mass processing in batches, such as cleaning up or merging large amounts of data from publication databases. Using a scalable architecture of the solution, many data sets can be processed with higher performance in a short time. Thus, other processes are provided with the required data more quickly and efficiently.

This study places particular emphasis on detecting duplicates in research data, where it is important to discover duplicates as part of a larger process, such as data cleaning or data integration. The developed prototype is intended to facilitate research in the field of data quality with a focus on RIS and research data, respectively. Since research in this area has not yet been advanced, the framework should be sufficiently flexible to allow it to easily integrate changes and extensions. It is also important to take into account the fact that a final version of the framework should be able to process large amounts of data. The aim is to implement an adaptive and flexible framework based on the presented model. The model consists of six phases: (1) data preparation, (2) search space definition, (3) attribute value comparison, (4) decision model, (5) clustering of the duplicates, and (6) verification. Particular emphasis is placed on the phases of comparing attribute values used for the decision model and clustering duplicates. These phases perform the essential steps of duplicate detection (other phases, such as searching the space definition, are trivial or not implemented (data preparation, verification)). When implementing the attribute value comparison, it should be noted that all alternatives are expected to be compared with each other when comparing the two x-tuples. When comparing alternatives, all attributes and all versions of attribute values should be compared. In addition, it should be possible to define several similarity functions for an attribute and to aggregate the resulting similarities (per attribute per similarity function) in a weighted manner.

When implementing the framework in the DataCleaner tool, there should be some similarity functions, but it also should be able to add new ones if necessary. The similarity functions should be normalised, i.e., always return a value between 0 and 1, otherwise, the probabilities can be falsified. The configuration takes place on two levels: on the one hand, the components are assembled and made known, this happens before the program is loaded. The second part of the configuration takes place at runtime. To assess the performance of the framework, the relevance of the results and the required runtime are considered. A high relevance in the duplicate detection results if the number of actual duplicates found is as high as possible ("hit rate"), while as few data records as possible should be falsely recognised as duplicates ("accuracy"). In scientific literature, only these factors are known as "recall" and "precision". It is easy to see that both factors are interdependent and that the hit rate generally decreases with increasing accuracy. A framework that simply marks each record as a duplicate will actually "detect" all existing duplicates and would have a 100% hit rate. However, most of the results will be irrelevant and the precision will be low. A counterexample of the solution, which only marks duplicates as such if it is absolutely certain, will have a high level of accuracy but may miss many duplicates and thus has a low hit rate. Depending on the application, a high hit rate or high accuracy can be considered more important, so it makes sense to allow the user to configure it. In the framework presented here, this is possible by setting the threshold value from which a data record is recognised as a duplicate of another data record.

The evaluation of "recall" and "precision" requires relevant test data. Ideally, they should be authentic data from practical applications, i.e., representing the real world, but in order to ensure relevance, it is important that all actual duplicates are known at the same time, as this is the only way to calculate the exact hit rate and accuracy. This can be achieved in small, while it is practically no longer possible in large databases in non-artificial environments. For the area we focused on, i.e., RIS and research data (e.g., third-party funded project data, patent data, etc.), they are usually confidential data, which are difficult to access and which are not suitable for publication. The alternative is artificial/synthetic/generated data, the structure of which should reflect reality as closely as possible. Since the duplicates are also generated in this case, it is known which data substitute duplicates, and the hit rate and accuracy can be determined precisely.

5. Discussion and Future Work

In addition to outdated and incomplete data, duplicates are one of the three most common symptoms of poor data quality as recognised both in scientific literature and in observations of the practitioners [35,36]. The quality of data is not pure IT, but rather a data management task. Therefore, it is often linked to "master data management". Detecting duplicates to ensure data quality is an important part of a comprehensive data strategy. Various measures are required, including both initial, one-off measures and activities to be carried out continuously. The research data and RIS should not be an exception and they should be considered a critical resource that needs to be properly managed if it has not already been. In other words, a comprehensive data quality management system is needed to ensure high data quality in research data.

To avoid duplicates, binding guidelines should be laid down in a written form, specifying how data (e.g., authors' names, institution addresses, and others in the RIS domain) should be recorded. These guidelines must be available to all stakeholders that input or further process the data. This can be also done by means of controlled input, where applicable. In addition, it is particularly recommended to clean up duplicates using IT technology. This is particularly useful in large data sets or when data from different, already existing systems are merged.

Duplicate search and elimination can be conducted both in research and in practice by means of different algorithms. Our proposal attempts to emulate a sense of human similarity and find duplicates, even if the spelling is very different. Another search method is called matchcode [37], where only the zip code or the first letter of the names are compared. In the future, the study can be extended with grouping algorithms. This can be achieved by using character-based symmetric metrics that introduce typographical error, and token-based symmetric metrics where the error is introduced because of rearrangement of words. Both metrics are focused on the representation of the database records in a string-based form. In addition, machine learning approaches can be developed for more sophisticated matching techniques for character-based and token-based symmetric metrics.

In addition, while we have discussed the research data as a domain of our interest, the issue of duplicates is evident in all kinds of databases, i.e., in almost every Information System and Database. Recent studies by Krasikov et al. [35] and Nikiforova et al. [36] also pointed to the issue of duplicates in open data. Thus, one potential direction could be the application of our framework to other data. It also supposes the use of other databases, covering IoT-related examples in addition to more conventional information systems and databases underlying them. This is even more the case given that data duplicates as a data quality issue is considered to be one of the key domain-specific data quality dimensions defined in the context of different IoT application domains, where smart grids and health sectors are found to be dominating [38]. This should allow us to conclude on its appropriateness for these purposes or limitations to be potentially resolved, thus making it a more universal solution.

In the future, usability testing will be conducted on the proposed framework and DataCleaner extension to find its usability with every stakeholder involved. This means that data managers and operators responsible for data management and other types of users should be involved to apply the proposed solution and testing its user-friendliness and ease of use. Although we expect a positive reaction, as this criterion was a prerequisite for us, including a choice of the tool that is not only one of the most widely used tools but also one that supports its users with a graphical representation of data processing results, this could also reveal improvements to be made in the future.

Furthermore, similar solutions used in the real world are expected to be identified, a list of which is expected to be received as user feedback, comparing them to our solution. This has not been achieved at the moment because the comparison is intended to be carried out with the solutions actually used, not the solutions presented in the scientific literature, which have often remained unimplemented or their support was terminated when the financial support has expired, or have not been widely used.

6. Conclusions

In this paper, we presented a Record Linkage framework consisting of six interconnected steps, i.e., (1) data preparation, (2) search space definition, (3) attribute value comparison, (4) a decision model, (5) clustering of the duplicates and (6) verification. It was then transformed into an extension for the DataCleaner tool that was followed by validation through its application to the real-world RIS. This allowed us to easily and automatically identify duplicates stored in the system of which data holders and operators were not aware, with their subsequent elimination and enrichment of original records thereby contributing to their completeness, thus preparing data for further processing that should provide more accurate results.

Cleaning duplicates improves the data quality and allows for optimising processes that are based on duplicate-free data. The assignment and consolidation of (research) data across all channels enable a more complete understanding of the current situation, i.e., a 360° research view of unique data only. These results could also be of high importance for stakeholders of research data systems and both academic and research institutions in general, since the issue of duplicates covered in the document is a problem that most institutions are familiar with, affecting both internal processes within institutions and external when funding is planned based on past achievements, which is insufficiently addressed.

At the same time, the framework proposed is domain-agnostic and can be applied in domains not related to the scientific or academic community. Low-quality and duplicate data have a significant impact on business processes and may pose a negative impact at various levels that can lead to bad decisions and potential loss of opportunities [36,39], including financial ones. This approach can also be used at the data preparation stage for data further use as an input for modelling and simulation processes, where the quality of the data and duplicates, in particular, may affect the model itself and the produced output.

The proposed framework is only one possible way of resolving this issue and should therefore serve as a call for other researchers to deal with it as the "golden record" could be achieved in full and broad meaning when all data quality issues are recognised and resolved, thus moving towards absolute data quality.

Author Contributions: Conceptualization, O.A., M.J. and A.N.; methodology, O.A.; software, O.A.; validation, O.A., M.J. and A.N.; formal analysis, O.A., M.J. and A.N.; investigation, O.A., M.J. and A.N.; resources, O.A., M.J. and A.N.; data curation, O.A., M.J. and A.N.; writing—original draft preparation, O.A., M.J. and A.N.; writing—review and editing, O.A., M.J., A.N., K.S., M.A. and S.J.; supervision, O.A., M.J., A.N. and K.S.; project administration, O.A., M.J. and A.N.; funding acquisition, O.A., M.J., A.N., K.S., M.A. and S.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data available on request from the authors.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Benson, P.R. Identifying and Resolving Duplicates in Master Dats. White Paper ISO 8000. 2021. Available online: https://eccma.org/what-is-iso-8000/ (accessed on 19 January 2022).
2. Naumann, F.; Herschel, M. An Introduction to Duplicate Detection. *Synth. Lect. Data Manag.* **2010**, *2*, 1–87. [CrossRef]
3. Periasamy, J.; Latha, B. Efficient hash function–based duplication detection algorithm for data Deduplication deduction and reduction. *Concurr. Comput. Pract. Exp.* **2019**, *33*, e5213. [CrossRef]
4. Nikiforova, A. Definition and Evaluation of Data Quality: User-Oriented Data Object-Driven Approach to Data Quality Assessment. *Balt. J. Mod. Comput.* **2020**, *8*, 391–432. [CrossRef]
5. Hoy, M.B. The "Internet of Things": What It Is and What It Means for Libraries. *Med. Ref. Serv. Q.* **2015**, *34*, 353–358. [CrossRef] [PubMed]
6. Miorandi, D.; Sicari, S.; De Pellegrini, F.; Chlamtac, I. Internet of things: Vision, applications and research challenges. *Ad Hoc Networks* **2012**, *10*, 1497–1516. [CrossRef]
7. Balaji, M.; Roy, S.K. Value co-creation with Internet of things technology in the retail industry. *J. Mark. Manag.* **2016**, *33*, 7–31. [CrossRef]
8. Bail, R.d.F.; Kovaleski, J.L.; da Silva, V.L.; Pagani, R.N.; Chiroli, D.M.D.G. Internet of things in disaster management: Technologies and uses. *Environ. Hazards* **2021**, *20*, 493–513. [CrossRef]
9. Pawar, A.; Kolte, A.; Sangvikar, B. Techno-managerial implications towards communication in internet of things for smart cities. *Int. J. Pervasive Comput. Commun.* **2021**, *17*, 237–256. [CrossRef]
10. Samih, H. Smart cities and internet of things. *J. Inf. Tehcnol. Case Appl. Res.* **2019**, *21*, 3–12. [CrossRef]
11. Kaupins, G.; Stephens, J. Development of Internet of Things-Related Monitoring Policies. *J. Inf. Priv. Secur.* **2018**, *13*, 282–295. [CrossRef]
12. Ziegler, P.; Dittrich, K.R. *Data Integration-Problems, Approaches, and Perspectives. Conceptual Modelling in Information Systems Engineering*; Krogstie, J., Opdahl, A.L., Brinkkemper, S., Eds.; Springer: Berlin/Heidelberg, Germany, 2007. [CrossRef]
13. Christen, P. A Survey of Indexing Techniques for Scalable Record Linkage and Deduplication. *IEEE Trans. Knowl. Data Eng.* **2011**, *24*, 1537–1555. [CrossRef]
14. Leitão, L.; Calado, P.; Herschel, M. Efficient and Effective Duplicate Detection in Hierarchical Data. *IEEE Trans. Knowl. Data Eng.* **2012**, *25*, 1028–1041. [CrossRef]
15. Daniel, C.; Serre, P.; Orlova, N.; Bréant, S.; Paris, N.; Griffon, N. Initializing a hospital-wide data quality program. The AP-HP experience. *Comput. Methods Programs Biomed.* **2018**, *181*, 104804. [CrossRef] [PubMed]

16. Elmagarmid, A.K.; Ipeirotis, P.; Verykios, V. Duplicate Record Detection: A Survey. *IEEE Trans. Knowl. Data Eng.* **2006**, *19*, 1–16. [CrossRef]
17. Chen, Q.; Britto, R.; Erill, I.; Jeffery, C.J.; Liberzon, A.; Magrane, M.; Onami, J.-I.; Robinson-Rechavi, M.; Sponarova, J.; Zobel, J.; et al. Quality Matters: Biocuration Experts on the Impact of Duplication and Other Data Quality Issues in Biological Databases. *Genom. Proteom. Bioinform.* **2020**, *18*, 91–103. [CrossRef] [PubMed]
18. Kwon, Y.; Lemieux, M.; McTavish, J.; Wathen, N. Identifying and removing duplicate records from systematic review searches. *J. Med. Libr. Assoc.* **2015**, *103*, 184–188. [CrossRef]
19. Winkler, W.E. Matching and record linkage. *WIREs Comput. Stat.* **2014**, *6*, 313–325. [CrossRef]
20. Baxter, R.; Christen, P.; Churches, T. A Comparison of Fast Blocking Methods for Record Linkage. In Proceedings of the Workshop on Data Cleaning, Record Linkage and Object Consolidation at the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 24–27 August 2003.
21. Weis, M.; Naumann, F.; Jehle, U.; Lufter, J.; Schuster, H. Industry-scale duplicate detection. *Proc. VLDB Endow.* **2008**, *1*, 1253–1264. [CrossRef]
22. Newcombe, H.B.; Kennedy, J.M.; Axford, S.J.; James, A.P. Automatic Linkage of Vital Records. *Science* **1959**, *130*, 954–959. [CrossRef]
23. Wang, R.Y.; Strong, D.M. Beyond Accuracy: What Data Quality Means to Data Consumers. *J. Manag. Inf. Syst.* **1996**, *12*, 5–33. [CrossRef]
24. Conrad, J.G.; Guo, X.S.; Schriber, C.P. Online Duplicate Document Detection: Signature Reliability in a Dynamic Retrieval Environment. In Proceedings of the twelfth international conference on Information and knowledge management (CIKM '03), Association for Computing Machinery, New York, NY, USA, 3–8 November 2003; pp. 443–452. [CrossRef]
25. Burdick, D.; Hernández, M.A.; Ho, C.T.; Koutrika, G.; Krishnamurthy, R.; Popa, L.; Stanoi, I.; Vaithyanathan, S.; Das, S.R. Extracting, Linking and Integrating Data from Public Sources: A Financial Case Study. *IEEE Data Eng. Bull.* **2011**, *34*, 60–67. [CrossRef]
26. Khtira, A. Detecting Feature Duplication in a CRM Product Line. *J. Softw.* **2020**, *15*, 30–44. [CrossRef]
27. TechTarget, WhatIs.Com. Available online: https://whatis.techtarget.com/definition/golden-record (accessed on 20 September 2021).
28. Steorts, R.C.; Ventura, S.L.; Sadinle, M.; Fienberg, S.E. A Comparison of Blocking Methods for Record Linkage. In *International Conference on Privacy in Statistical Databases*; PSD 2014 Lecture Notes in Computer Science; Domingo-Ferrer, J., Ed.; Springer: Cham, Switzerland, 2014; Volume 8744, pp. 253–268. [CrossRef]
29. Yan, S.; Lee, D.; Kan, M.Y.; Giles, L.C. Adaptive Sorted Neighborhood Methods for Efficient Record Linkage. In Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries (JCDL '07). Association for Computing Machinery, New York, NY, USA, 18–23 June 2007; pp. 185–194. [CrossRef]
30. Christophides, V.; Efthymiou, V.; Palpanas, T.; Papadakis, G.; Stefanidis, K. An Overview of End-to-End Entity Resolution for Big Data. *ACM Comput. Surv.* **2021**, *53*, 1–42. [CrossRef]
31. Panse, F.; Van Keulen, M.; De Keijzer, A.; Ritter, N. Duplicate detection in probabilistic data. In Proceedings of the IEEE 26th International Conference on Data Engineering Workshops (ICDEW2010), Long Beach, CA, USA, 1–6 March 2010; pp. 179–182. [CrossRef]
32. Fellegi, P.; Sunter, A.B. A theory for record linkage. *J. Am. Stat. Assoc.* **1969**, *64*, 1183–1210. [CrossRef]
33. Batini, C.; Scannapieco, M. *Data Quality: Concepts, Methodologies and Techniques*; Springer: New York, NY, USA, 2006. [CrossRef]
34. Manning, C.D.; Raghavan, P.; Schütze, H. *Introduction to Information Retrieval*; Cambridge University Press: New York, NY, USA, 2008. [CrossRef]
35. Krasikov, P.; Obrecht, T.; Legner, C.; Eurich, M. Open Data in the Enterprise Context: Assessing Open Corporate Data's Readiness for Use. In *International Conference on Data Management Technologies and Applications*; Springer: Cham, Switzerland, 2020; pp. 80–100. [CrossRef]
36. Nikiforova, A.; Kozmina, N. Stakeholder-centred Identification of Data Quality Issues: Knowledge that Can Save Your Business. In Proceedings of the 2021 Second International Conference on Intelligent Data Science Technologies and Applications (IDSTA), Tartu, Estonia, 15–17 November 2021; pp. 66–73. [CrossRef]
37. Premtoon, V.; Koppel, J.; Solar-Lezama, A. Semantic code search via equational reasoning. In Proceedings of the 41st ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI 2020), Association for Computing Machinery, New York, NY, USA, 15–20 June 2020; pp. 1066–1082. [CrossRef]
38. Karkouch, A.; Mousannif, H.; Al Moatassime, H.; Noel, T. Data quality in internet of things: A state-of-the-art survey. *J. Netw. Comput. Appl.* **2016**, *73*, 57–81. [CrossRef]
39. Vetrò, A.; Torchiano, M.; Mecati, M. A data quality approach to the identification of discrimination risk in automated decision making systems. *Gov. Inf. Q.* **2021**, *38*, 101619. [CrossRef]