ZfdG

Zeitschrift für digitale Geisteswissenschaften

Beitrag aus:
Zeitschrift für digitale Geisteswissenschaften

Titel:
Automatisiertes Record Linkage in prosopographischen Datenbeständen am Beispiel historischer Quellen Leipzigs

Autor*in:
Jan Michael Goldberg

Kontakt: jan.goldberg@wiwi.uni-halle.de
Institution: Martin-Luther-Universität Halle Wittenberg, Lehrstuhl für empirische Makroökonomik
GND: 1240406630 ORCID: 0000-0002-4817-4283

Autor*in:
Marcel Mernitz

Kontakt: marcel.mernitz@informatik.uni-halle.de
Institution: Martin-Luther-Universität Halle Wittenberg, Institut für Informatik
GND: 1275436560 ORCID: 0000-0001-6464-2844

DOI des Artikels:
10.17175/2023_001_v2

Nachweis im OPAC der Herzog August Bibliothek:
185844733X

Erstveröffentlichung:
26.01.2023

Version 2.0:
29.09.2023

Lizenz:

Sofern nicht anders angegeben (cc) BY

Medienlizenzen:
Medienrechte liegen bei den Autor*innen

Letzte Überprüfung aller Verweise:
31.08.2023

Format:
PDF ohne Paginierung, Lesefassung

GND-Verschlagwortung:
Duplikaterkennung | Datenverknüpfung | Personenbezogene Daten | Algorithmus | Genealogie | Geschichtswissenschaft |

Empfohlene Zitierweise:
Jan Michael Goldberg / Marcel Mernitz: Automatisiertes Record Linkage in prosopographischen Datenbeständen am Beispiel historischer Quellen Leipzigs. In: Zeitschrift für digitale Geisteswissenschaften 8 (2023). 26.01.2023. Version 2.0 vom 29.09.2023. HTML / XML / PDF. DOI: 10.17175/2023_001_v2

Änderungen in Version 2.0 (29.09.2023):
Inhaltliche Ergänzungen an mehreren Stellen gemäß Gutachten.

Jan Michael Goldberg, Marcel Mernitz

Automatisiertes Record Linkage in prosopographischen Datenbeständen am Beispiel historischer Quellen Leipzigs

Abstracts

In dieser Studie wird ein automatisierter Ansatz zum *Record Linkage* in prosopographischen Datenbeständen vorgestellt. In ihm sind zahlreiche genealogische Regeln zur Verknüpfung von Personen implementiert. Dadurch ist er besonders für Datenbestände geeignet, die zu den abgebildeten Individuen viele genealogisch relevante Informationen bereithalten. Dazu wird eine normierte Datenstruktur definiert, in die die Eingangsdaten einzuordnen sind. Der Algorithmus erkennt innerhalb dieser Datenstruktur Einträge zu gleichen Personen und führt diese automatisch zusammen. In diesem Zuge wird eine Formalisierung von genealogischen Heuristiken vorgenommen. Die Funktionsfähigkeit des Algorithmus wird am Beispiel historischer Datenbestände aus Leipzig erfolgreich dargestellt. Der Programmcode ist in Python realisiert worden und frei verfügbar.

In this study, an automated approach to *record linkage* in prosopographic datasets is presented. It implements numerous genealogical rules for linking individuals. This makes it particularly suitable for datasets that contain a lot of genealogically relevant information about the represented individuals. For this purpose, a standardized data structure is defined into which the input data is to be arranged. The algorithm recognizes entries pertaining to the same persons within this data structure and merges them automatically. In this process, a formalization of genealogical heuristics is performed. The functionality of the algorithm is successfully demonstrated using historical datasets from the city of Leipzig as an example. The program code has been realized in Python and is freely available.

1. Einleitung

Gleiches mit Gleichem zu verbinden, stellt überall dort eine besondere Herausforderung dar, wo keine eindeutigen Identifikationsmerkmale vorliegen. Dieses Problem tritt in wissenschaftlichen Untersuchungen insbesondere dann auf, wenn historische Personendaten Forschungsgegenstand sind. Immer größere Datenmengen sorgen zudem zunehmend dafür, dass eine manuelle Bearbeitung erschwert wird. Dadurch besteht ein Bedarf an automatisierten *Record-Linkage*-Lösungen. Neben den klassischen wissenschaftlichen Anwendungen betrifft das unter anderem auch Projekte wie *Time-Machine*-Anwendungen.[1] Im deutschen Sprachraum sind derzeit beispielsweise die Projekte in Leipzig, Jena und Köln zu nennen.[2] Perspektivisch ist denkbar, dass in vielen deutschsprachigen Städten solche Time Machines initiiert werden. Eine besondere Herausforderung dabei ist es, viele unterschiedliche Quellen zusammenzuführen. Aus dieser Perspektive heraus besteht ein erhöhter Bedarf an Record-Linkage-Lösungen, die die Besonderheiten der deutschen Sprache berücksichtigen.

Um Lebensläufe von Individuen oder Familienentwicklungen nachvollziehen zu können, greifen Historiker*innen sowie Wirtschafts- und Sozialwissenschaftler*innen auf verschiedene Daten zur persönlichen Identifikation zurück. Hierfür gibt es eine Vielzahl verschiedener Record-Linkage-Ansätze. Schon in der Antike wurde über die Bevölkerung Buch geführt, beispielsweise zur Übersicht über die zur Musterung heranzuziehende Bevölkerung, zur Wahlrechtverteilung oder zur Erhebung von Steuern.[3] Die meisten historischen Informationen über Individuen der Neuzeit befinden sich in prosopographischen Quellen wie Kirchenbüchern. Die historische Datenerhebung kennt dabei zur eindeutigen Personenerkennung keine eindeutigen Identifikatoren wie Steuer-, Personalausweis- oder Sozialversicherungsnummer. Daher muss auf andere Daten einer Person zurückgegriffen werden, beispielsweise den Namen, Geburts- und Sterbedaten oder die Namen der Eltern. Diese Daten allerdings sind nicht geschützt vor Fehlern oder Verlust. Daraus ergibt sich eine enorme Ungenauigkeit ebendieser Daten.[4] Zudem sind große Datenbestände unübersichtlich oder gar nicht überschaubar. Das zeigt sich beispielsweise, wenn Personen in einem Zensus händisch im darauffolgenden Zensus anhand der Angaben zur Stadt oder Gegend beziehungsweise zum Land gesucht werden.[5] Problematisch an diesem Ansatz ist, dass verzogene Menschen in dem folgenden Zensus aufgrund des Ortswechsels

[1] *Time Machines* sind Konstrukte, in denen historische Daten verschiedenster Quellen zusammengeführt werden. Dadurch werden beispielsweise individuelle Biografien, politisch-städtische Dynamiken und die Veränderung der Bausubstanz verknüpft auf einer Plattform sichtbar. Diese werden öffentlich zur Verfügung gestellt und können zur Forschung und Bildung genutzt werden. Vgl. Kaplan 2015, S. 73.
[2] Vgl. Time Machine Organisation 2022.
[3] Vgl. Hin et al. 2016, S. 50.
[4] Vgl. Feigenbaum 2016; Hin et al. 2016, S. 50, 52; Massey 2017, S. 129, 131.
[5] Vgl. Massey 2017, S. 130.

nicht gefunden werden. Die Aussagekraft der Ergebnisse ist hierbei also durch die geografische Mobilität der Bevölkerung gefährdet. Neben der Qualität der Quellen haben zudem knapp bemessene personelle, finanzielle und zeitliche Ressourcen in der Forschung Einfluss auf die Qualität der Record-Linkage-Ergebnisse.[6] Unter anderem aus diesem Umstand heraus wurden neben einer händischen Verknüpfung halb- und vollautomatisierte Verfahren entwickelt.[7] Welche Herangehensweise hierbei die richtige ist, ist abhängig vom Projektziel. Da es oftmals keine offiziellen Regeln für das Verbinden der Records gibt, existieren zahlreiche Heuristiken für die Verknüpfung der Daten. Eine Grundvoraussetzung für ein automatisiertes Record Linkage ist die Formalisierung der Heuristiken, die dem Verbinden der Daten zugrunde liegen.

Ziel des hier vorgestellten Ansatzes ist es, Heuristiken zum Record Linkage in prosopographischen Datenbeständen mit vielen genealogisch relevanten Informationen zu formalisieren und in einem automatisierten Algorithmus umzusetzen. Genealogisch relevante Informationen sind dabei Lebensdaten wie Geburts- oder Sterbedatum, Berufe oder Informationen über die Eltern einer Person. Dieser Algorithmus soll dazu geeignet sein, ein Record Linkage in deutschsprachigen Datenbeständen zu ermöglichen. Zu diesem Zweck wird im nächsten Abschnitt zunächst ein Überblick über den Stand der Forschung gegeben. Darauffolgend findet die Beschreibung des entwickelten Algorithmus statt, bevor sich dieser einer Validierung anhand von historischen Leipziger Quellen unterzieht. Abschließend wird das Ergebnis zusammengefasst. Der Algorithmus selbst wird in der Programmiersprache Python 3.8 umgesetzt und ist im Online-Repositorium zu finden.

2. Forschungsstand

Zunächst wird auf verschiedene Methoden des Record Linkage eingegangen. Danach findet eine Betrachtung der Besonderheiten prosopographischer Datenbestände mit umfangreichen genealogisch relevanten Daten statt.

2.1 Methoden des Record Linkage historischer Datenbestände

Wie eingangs erwähnt, gibt es unterschiedliche Ansätze, wie Datensätze zusammengeführt werden können. Diese Darstellung fokussiert sich explizit auf den Stand der Forschung bei der Anwendung auf historische Daten.[8] Zweck ist es, einen Überblick über verschiedene Verfahren und Ideen zu geben, ohne dabei jedoch einen Anspruch auf Vollständigkeit zu erheben. Das Record Linkage historischer Daten hat sich in den vergangenen Jahrzehnten stetig verändert, wie beispielsweise Massey aufzeigt.[9] Übergreifend werden von Gellatly als wesentliche Herausforderungen zum einen die Skalierbarkeit auf große Datenbestände, zum anderen die Genauigkeit und Effizienz der Algorithmen identifiziert.[10] Als dritte große Herausforderung werden Datenschutzaspekte genannt.[11] Der Datenschutzaspekt wird im Weiteren vernachlässigt, da der Algorithmus auf Daten ausgelegt werden soll, die aufgrund ihres Alters vom deutschen Datenschutzrecht nicht tangiert werden. Die Analyse von Daten aus verschiedenen Zeiträumen weist dabei unterschiedliche Herausforderungen auf, beispielsweise in der Standardisierung von Namensschreibweisen oder der generellen Datenerfassung.[12]

Zum Record Linkage können verschiedenste Variablen herangezogen werden. Grundlegend dabei ist, dass Variablen / Attribute zur Verfügung stehen, die einen identischen Schlüssel aufweisen.[13] Dies kann beispielsweise der Name, das Geburtsdatum oder die Sozialversicherungsnummer sein. Auch können Graphen genutzt werden, um die Ähnlichkeit der Records untereinander darzustellen.[14] Um die Daten zu vergleichen, ist eine vorhergehende Bereinigung notwendig.[15]

[6] Vgl. Massey 2017, S. 129f.
[7] Bei einem halbautomatisierten Ansatz unterbreitet ein Programm dem Forschungspersonal Vorschläge zu möglichen Treffern. Jedoch bestimmt das Forschungspersonal und kein Algorithmus über die Verknüpfung.
[8] Als Einführung in die Grundlagen des Themas vgl. Gu et al. 2003.
[9] Sie selbst prüft verschiedene Record-Linkage-Verfahren und kommt beispielsweise zu dem Schluss, dass Ergebnisse besser werden, wenn die Altersangaben zwischen zwei zeitlich auseinanderliegenden Quellen in Bezug auf die zeitliche Differenz zwischen diesen umgerechnet werden. Die besten Resultate erzielt sie mit probabilistischen Matching-Techniken. Vgl. Massey 2017, S. 129, 140.
[10] Vgl. Gellatly 2015, S. 114, 122.
[11] Vgl. Christen et al. 2015, S. 87.
[12] Vgl. Georgala et al. 2015, S. 173.
[13] Vgl. Baxter et al. 2003, S. 2.
[14] Die Qualität der Verknüpfungen wird dabei besser, wenn man zeitliche Restriktionen einbeziehe, beispielsweise des möglichen Schwangerschaftszeitraums der Frau. Vgl. Nanayakkara et al. 2018.
[15] Vgl. Gellatly 2015, S. 116.

Gellatly testet einen Ansatz, bei dem er verschiedene Variablen kombiniert und im Folgenden analysiert, welche Kombinationen die besten Ergebnisse erzielen. Diese erreicht er bei einer Kombination von Geburtsjahr (nicht das exakte Datum), Geschlecht, Nachname, einer Variable, die sich aus der Anzahl von Brüdern und Schwestern zusammensetzt, und den ersten drei Buchstaben des Vornamens.[16]

Efremova et al. nutzen dahingegen ein ›disjunctive blocking‹.[17] Darin werden die ersten Buchstaben eines Namens einer phonetischen Analyse unterzogen. Nur, wenn diese einen gewissen Grad an Ähnlichkeit aufweisen, wird das Record Linkage fortgesetzt. Im folgenden Schritt wird die Similarität zwischen verschiedenen Records berechnet. Die besten Ergebnisse erhalten sie unter Hinzuziehung der Namenshäufigkeit innerhalb der untersuchten Datenbank sowie der geografischen Distanz.

Statt einer binären Verknüpfung (Zuordnung / keine Zuordnung) gibt es auch Systeme, die Abstufungen verwenden. Sichere Verknüpfungen werden darin anders bewertet als unsichere.[18] Thorvaldsens automatisierte Anwendung auf norwegische Daten nimmt viele Verknüpfungen aufgrund von Ungewissheit nicht automatisch vor und lässt einen beträchtlichen Spielraum für die (nachfolgende) manuelle Verknüpfung.[19]

Anhand englischer Daten zeigen Georgala et al., dass String-Metriken wie die Levenshtein- oder Jaro-Winkler-Distanz besser als phonetische Ähnlichkeitsanalysen funktionieren, diese jedoch wiederum deutlich bessere Ergebnisse aufweisen als eine absolute Gleichheit der Namen.[20]

Zur Unterstützung des Record Linkage existieren verschiedene Programme. In diese wird hier nicht im Detail eingeführt. Lediglich beispielhaft genannt werden drei Lösungen. Eine Lösung, die explizit auf das Record Linkage von genealogischen GEDCOM-Dateien (GEnealogical Data COMmunication, siehe unten) ausgelegt ist: *GedTool*.[21] Zur Identifizierung von Dateneinträgen zu gleichen Personen können darin bis zu acht Kriterien wie der Vorname, der Nachname oder eine ID bestimmt werden. Stimmen diese überein, kann ein Record Linkage stattfinden. Alle Einträge, die den definierten Kriterien entsprechen, werden gemeinsam angezeigt und können nachfolgend manuell zusammengeführt werden. Eine phonetische Suche mit den Algorithmen Soundex, Kölner Phonetik und Double Metaphone kann ebenfalls ausgeführt werden.[22] Hierbei handelt es sich demnach um eine semi-automatisierte Lösung.

Ein weiteres Record-Linkage-Programm stellt *Demolink* dar. Eli Fure evaluiert dieses anhand norwegischer Daten. Sie kommt zu dem Schluss, dass für die Anwendung eine Vorstellung über den historischen Kontext einer Quelle notwendig ist, um bessere Ergebnisse als eine automatisierte Lösung zu erzielen. Damit meint sie, dass die Forschenden u. a. Wissen darüber haben müssen, welche Namen im untersuchten Gebiet gleich sind, ohne dass ein Algorithmus sie zuordnen kann. Hierzu seien menschliche Eigenschaften notwendig.[23] Ein Beispiel dafür sind die Namen Goldberg und Goldbrich, die in Nordböhmen und der südlichen Oberlausitz bis etwa zur zweiten Hälfte des 18. Jahrhunderts synonym verwendet werden.

Zuletzt genannt wird *OpenRefine*. Zwar hat *OpenRefine* ein breiteres Anwendungsgebiet, kann jedoch auch zum Rekord Linkage verwendet werden. Ein Vorteil ist, dass hierdurch eigene Daten mit Referenzressourcen wie Wikidata abgeglichen und verbunden werden können. Auch unterstützt *OpenRefine* die *Reconciliation Service API*, ein Protokoll zum Datenmatching im Web.[24]

Abramitzky et al. zeigen jedoch auf, dass auch automatisierte Vorgehensweisen zufriedenstellende Ergebnisse erzielen können.[25] Da nie mit Sicherheit bestimmt werden kann, ob zwei Records tatsächlich dieselbe Entität beschreiben, sind solche Vorgehen probabilistisch. Bei einem Vergleich verschiedener Methoden durch Abramitzky et al. erreichen auch automatisierte Ansätze Falschpositivraten von unter fünf Prozent. Zudem zeigen sie, dass auch Menschen nicht frei von Fehlern sind und ebenfalls falschpositive Ergebnisse erzeugen.[26] In ihrem automatischen Ansatz demonstrieren Abramitzky et al. ein dreischrittiges Verfahren: Zunächst sind (1.) Variablen für die Verknüpfung auszuwählen, dann setzen sie (2.) mit dem Expections-Maximization-Algorithmus einen Algorithmus zur Berechnung der Wahrscheinlichkeit der Übereinstimmung von zwei Datensätzen ein,

[16] Vgl. Gellatly 2015, S. 122f.
[17] Vgl. Efremova et al. 2015.
[18] Vgl. Thorvaldsen et al. 2015, S. 163f.
[19] Vgl. Thorvaldsen et al. 2015, S. 168.
[20] Vgl. Georgala et al. 2015, S. 187.
[21] Vgl. Schulz 2017.
[22] Die Programmierung dieser Funktionen ist jedoch nicht nachvollziehbar, da es sich um ein kommerzielles Produkt handelt und der Code des Programms (es handelt sich um Excel-Makros) nicht einsehbar ist.
[23] Vgl. Fure 2000.
[24] Vgl. Delpeuch et al. 2023.
[25] Vgl. Abramitzky et al. 2021.
[26] Vgl. Abramitzky et al. 2021, S. 865.

schließlich wird (3.) die Wahrscheinlichkeit der Übereinstimmung bewertet.[27] Die hohe Verlässlichkeit ihrer Vorgehensweise zeigt sich darin, dass sie bei der Berechnung der beruflichen und intergenerationalen Mobilität aus ihren Verknüpfungen ihrer Daten ähnliche Resultate wie in bereits bestehenden, manuellen Verknüpfungen erhalten.[28]

Da der Algorithmus mit der Programmiersprache Python umgesetzt wird, liegt auch die Verwendung Python-spezifischer Bibliotheken nahe (z. B. *RecordLinkage* von Jonathan de Bruin). Zunächst jedoch wird der Algorithmus fernab von den Möglichkeiten oder Restriktionen programmiersprachenspezifischer Bibliotheken entwickelt. Deswegen findet keine Vorfestlegung auf solche statt. Zugleich aber sind solche Bibliotheken sinnvolle Werkzeuge, um Record-Linkage-Herausforderungen praktisch zu begegnen; auch zur Umsetzung des Algorithmus in diesem Fall.

Grundsätzlich ist es zudem möglich, Methoden des maschinellen Lernens auf Record-Linkage-Herausforderungen anzuwenden. So könnte beispielsweise die Ähnlichkeit manuell verknüpfter Datensätze ausgewertet werden, um die Systematik der Verknüpfungen zu erkennen auch auf weitere Daten anzuwenden. Solchen Ansätzen gemein ist jedoch, dass das erzeugte Modell – und somit das Ergebnis – von den Trainingsdaten abhängig ist. Aus diesem Grund wird in diesem Algorithmus bewusst darauf verzichtet, da bekannte genealogische Heuristiken zunächst in einem statischen Modell formalisiert werden sollen. Darauf aufbauend kann nachfolgende Forschung diese Ergebnisse nutzen, Verfahren maschinellen Lernens zu implementieren.

2.2 Format genealogisch-prosopographischer Datenbestände

Besonders interessant erscheint die Anwendung eines automatisierten Record Linkage auf große Datenbestände mit genealogisch relevanten Daten. Das Record Linkage muss dabei jedoch immer auch die besondere Struktur der Daten betrachten. Genealogisch relevante Datenbestände weisen andere Besonderheiten auf als einfache Listen, beispielsweise Notenlisten von Schulen. Oftmals stehen dabei in genealogischen Datenbeständen im deutschsprachigen Raum fünf Lebensereignisse im Zentrum: Geburten, Taufen, Heiraten, Todesfälle und Beerdigungen. Die Erfassung dieser Aspekte bildet ein Grundgerüst zur Beschreibung eines individuellen Lebensverlaufs. Daneben werden oft weitere Informationen wie Wohnorte oder Berufsangaben, vor allem aber die Verknüpfung zu den Eltern und Kindern ergänzt.

Quellen, die genealogisch relevante Daten enthalten, sind sehr unterschiedlich strukturiert. Die zugrundeliegenden Primärquellen sind oftmals Manuskripte. Hier sind vorwiegend Kirchenbücher zu nennen. Verschiedene prosopographische Quellen enthalten dabei unterschiedliche Informationen.[29] Allerdings existiert auch eine große Menge an Sekundärquellen, die bereits aufgearbeitete Daten präsentieren. Solche Daten können dabei unterschiedlich und höchst individuell strukturiert sein, beispielsweise als Fließtext in Chroniken vorliegen oder in Stammtafeln abgedruckt sein. Auch im digitalen Raum existieren mannigfaltige Formate. Hier haben sich allerdings auch spezielle Austauschformate für genealogische Daten entwickelt.

Für diese Studie wird davon ausgegangen, dass einzelne Quellen so aufgearbeitet werden können, dass sie in einer Tabelle vorliegen. Jeder Eintrag der Quelle entspricht einer Zeile (i. d. R. eine Person), jede Spalte hingegen einem Datenfeld in der Quelle. Die in einer Zeile enthaltenen Informationen werden im Weiteren als Record bezeichnet. Herausforderung hierbei ist, dass die Datenfelder / Spalten tatsächlich vergleichbare Informationen enthalten müssen. Die Zuordnung von Informationen aus einer Quelle in die korrekten Datenfelder ist dadurch schwierig, dass trotz gleicher Bezeichnung in den Originalquellen unterschiedliche Informationen gemeint sein können. Zum Beispiel kann mit dem »Stand« in einer Quelle der Beruf (z. B. »Müller«) gemeint sein oder aber der Familienstand (z. B. »verheiratet«). Für ein Record Linkage zwischen verschiedenen Datenbeständen ist also die Definition des Inhalts der Datenfelder unerlässlich.

Als wesentlicher Standard zum Austausch genealogischer Informationen hat sich das GEDCOM-Format herausgebildet.[30] In diesem werden einzelne Informationen sogenannten Tags zugewiesen, die eine ähnliche Funktion wie Datenfelder / Spalten haben (z. B. beschreibt der Tag OCCU eine Berufsangabe). Aber auch aus GEDCOM-Daten ergeben sich Probleme: Zwar sind diese strukturiert, doch gibt es nicht für alle Informationen eigene Tags. Auch wenn mit GEDCOM 5.5.1 ein Standard existiert,[31] legt dieser nicht immer fest, welcher Inhalt den Tags zugeordnet werden darf. Im Standard ist beispielsweise für die Nennung

[27] Vgl. Abramitzky et al. 2020, S. 94.
[28] Dieses stellt zugleich ein geeignetes Beispiel für die Anwendung und den Nutzen von Record-Linkage-Algorithmen in der ökonomischen Forschung dar. Vgl. Abramitzky et al. 2020, S. 106f.
[29] Efremova et al. nennen beispielsweise Variablen, die sie aus der Analyse von Geburts-, Todes- und Heiratsdokumenten erhalten. Vgl. Efremova et al. 2015, S. 132.
[30] Vgl. Gellatly 2015, S. 112; Harviainen / Björk 2018, S. 4.
[31] Vgl. The Church of Jesus Christ of Latter-day Saints 2019.

von Ortsangaben eine Trennung der administrativen Gliederungsebenen durch ein Komma vorgesehen. Nutzer*innen jedoch müssen sich daran nicht halten, sondern können diese ›Freitextfelder‹ ausfüllen, wie es ihnen beliebt und wie sie diese interpretieren.

Einen weiteren Standard stellt Gedbas4all dar.[32] Anders als GEDCOM, in der die einzelnen Informationen zu einer Person zwar zusammengeführt, die zugrundeliegenden Quellen aber schlecht nachvollziehbar sind, basiert dieses Modell auf einer Verknüpfung von Records, die im Nachhinein wieder voneinander gelöst werden können. In dem Datenmodell gibt es einige Variablen, die auch konkret definiert wurden. Besonders für die Zeitangaben gibt es eine detaillierte Normierung.[33] Das Datenmodell enthält jedoch nicht zu allen möglichen Variablen eine detaillierte Erläuterung. Zudem hat es noch keine weite Verbreitung gefunden.

Es zeigt sich, dass kein allgemeingültiges und ausreichend detailliertes System zur Definition vieler möglicher Schlüssel für ein Record Linkage auf Basis zahlreicher Variablen existiert. Darum werden im Folgenden mögliche Datenfelder im Rahmen der Entwicklung des Algorithmus definiert.

3. Algorithmus zum Record Linkage

Die oben aufgeführten Algorithmen scheinen auf ihre jeweiligen Anwendungen bezogen zwar effektiv zu sein, doch können sie nicht auf alle prosopographischen Quellen übertragen werden. Eine allgemeingültige Lösung für alle deutschsprachigen Quellen kann auch hier nicht entwickelt werden. Die aufgezeigte Lösung aber bildet viele mögliche Fälle bereits ab und stellt eine geeignete Grundlage zur weiteren Anpassung dar. Das Ergebnis ist also nicht nur auf eine einzelne Anwendung angepasst, sondern kann für verschiedene prosopographische Quellen (speziell solche mit einer hohen Dichte genealogisch relevanter Informationen) adaptiert werden. Auch wird einem weiteren bestehenden Nachteil der dargestellten Ansätze begegnet, welche vorwiegend auf englischsprachige, norwegische oder niederländische Datensätze angewendet wurden: Es gibt in jeder Sprache Besonderheiten, die es zu berücksichtigen gilt, auch die deutsche Sprache stellt keine Ausnahme dar. Der im Folgenden vorgestellte Algorithmus ist daher nur mit deutschsprachigen Daten kompatibel und nimmt Rücksicht auf die phonetischen Besonderheiten der deutschen Sprache. Auch hier kann jedoch eine Anpassung vorgenommen werden, indem Regeln weiterer Sprachen integriert werden. Da es einen in dieser Art ausgestalteten Algorithmus bislang nicht gab, wird hier eine Forschungslücke geschlossen. Aufbauend auf dem Forschungsstand verwendet dieser besonders solche Metriken und Verfahren, die sich in den dargestellten Lösungen als tauglich erwiesen haben.

Der Algorithmus wird im Folgenden textuell erklärt. Die Erläuterung orientiert sich am Aufbau der programmtechnischen Umsetzung. Es ist insbesondere auch ein Anspruch, den Quellcode zugänglich zu machen und so eine Anpassung an die jeweilige Herausforderung zu ermöglichen. Hierzu wird der Algorithmus in der Programmiersprache Python 3.8 umgesetzt. Dieser ist im Online-Repositorium verfügbar.

Wesentliche Herausforderungen bestehen in der Normierung, Strukturierung und Bereinigung von Eingangsdaten sowie der Prüfung einer Similarität zwischen verschiedenen Records. Die Bereinigung der Daten ist eine Voraussetzung für die Prüfung der Similarität der Datensätze; letztere wiederum stellt eine notwendige Bedingung zur Verknüpfung im Zuge des Record Linkage dar. Im folgenden Abschnitt wird zunächst eine detaillierte Übersicht über den Algorithmus gegeben. Danach wird eine Normalform der Daten definiert (im Weiteren Normform), in die die Eingangsdaten gebracht werden müssen. Dies geschieht, damit die Datenfelder / Spalten gleichartige Daten enthalten. Daran anschließend wird die Datenbereinigung und -strukturierung behandelt, bevor die genealogischen Heuristiken, die dem Vergleich zweier Records dienen, formalisiert werden. Abschließend wird bestimmt, in welcher Form die Zusammenführung der Records geschieht.

3.1 Funktionsweise des Algorithmus

Der Algorithmus ist auf prosopographische Quellen angepasst, die genealogisch relevante Daten enthalten. Es ist denkbar, dass es viele prosopographische Quellen gibt, die Daten enthalten, welche durch die Normform nicht adäquat abgebildet werden (z. B. Immatrikulationslisten). Hier wird deutlich, dass nicht alle erdenklichen (und praktisch auch irgendwo vorkommenden) Attribute prosopographischer Quellen Einbindung finden können. Nichtsdestotrotz wird mit jeder Information, die nicht genutzt wird, eine Möglichkeit verworfen, das Record Linkage positiv zu beeinflussen. In Fällen besonderer Relevanz spezieller Variablen für eine Aufgabenstellung sollten diese im Algorithmus ergänzt werden.

[32] Vgl. Verein für Computergenealogie 2016a.
[33] Vgl. Verein für Computergenealogie 2016b.

Der grundlegende Ablauf zur Verarbeitung der Daten ist in Abbildung 1 ersichtlich. Um den Algorithmus ausführen zu können, müssen die Daten aufbereitet werden. Das kann manuell, aber auch durch ein gesondertes Programm geschehen.[34] Der Algorithmus ist darauf ausgelegt, zwei in der Normform vorliegende Datensätze dem Record Linkage zu unterziehen.[35] Nach der Zusammenführung kann der entstandene, verknüpfte Datensatz dann in weitere, übliche Formate wie z. B. GEDCOM übertragen werden. Zur Erstellung einer GEDCOM-Datei aus dem Ergebnis des Algorithmus kann beispielsweise das bereits im Forschungsstand erwähnte Programm *GedTool* genutzt werden. Die konkrete Umwandlung des Ergebnisses in eine GEDCOM-Datei findet hier jedoch keine weitere Erläuterung, sondern ist der Bedienungsanleitung des Programms zu entnehmen.[36]

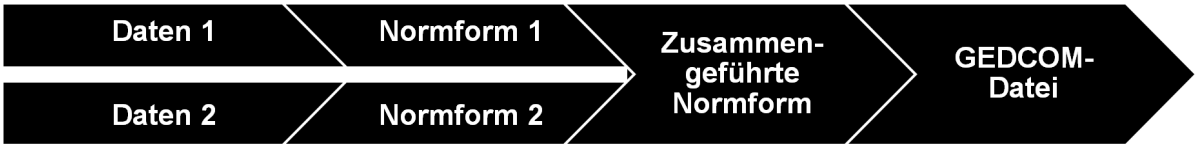

Abb. 1: Ablauf der Datenverarbeitung. [Goldberg / Mernitz 2023]

Nach der Transformation in die Normform wird eine Bereinigung und weitere Strukturierung der Informationen vorgenommen. Dieser Schritt ist notwendig, beispielsweise um Abkürzungen zu entfernen und Schreibfehler zu korrigieren.

Nachfolgend wird ein Vergleich zwischen einzelnen Records erzeugt. Für jede Zeile in der ersten Tabelle wird dazu geprüft, ob die einzelnen Records der zweiten Tabelle disjunkt sind, also nicht dieselbe Person abbilden. Hierzu sind verschiedene genealogische Regeln implementiert, die eine Zusammenführung ausschließen sollen (z. B. ist eine Taufe nach dem Tod nicht möglich).

Danach wird für die nichtdisjunkten Records eine Similaritätsprüfung durchgeführt. Hierdurch soll herausgefunden werden, ob die Personen similär sind – also diese beiden Records dieselbe historisch existierende Person beschreiben und die Informationen entsprechend zu verknüpfen sind. Hierzu werden die Namen verglichen. Bei einem Wert von 1 wird eine vollständige Similarität der verglichenen Personen indiziert, bei 0 eine Abwesenheit dieser. Daneben können bei uneindeutiger Similarität auch Zwischenwerte erreicht werden. Dadurch wird ein graphbasierter Ansatz implementiert, in dem jeder Record im ersten Datensatz zu jedem im zweiten eine gewichtete Beziehung aufweist. Zudem ist dieser Ansatz probabilistisch, da oftmals nicht mit Sicherheit von einer Similarität ausgegangen werden kann.

Der grundlegende Ablauf ist in Abbildung 2 dargestellt. Eine ausführliche Erläuterung der einzelnen Schritte findet in den folgenden Abschnitten statt.

[34] In vielen Fällen werden die Spaltenüberschriften anzupassen und deren Inhalt entsprechend zuzuordnen sein. Mit tabellarisch vorliegenden Informationen ist die Umsetzung dieses Schrittes vergleichsweise einfach durchführbar. Liegen die Daten als Fließtext vor, so müssen diese zunächst in ein tabellarisches Format überführt werden. Anders sieht das jedoch bei GEDCOM-Dateien aus, die zwar auch Fließtext darstellen, jedoch gut genug strukturiert sind, um sie in ein entsprechendes tabellarisches Format zu überführen. Dazu bietet sich ein GEDCOM-Parser an, welcher in gängigen Genealogieprogrammen enthalten ist.
[35] Sollten mehr als zwei Datensätze verglichen werden, so sind zunächst zwei auszuwählen und zusammenzuführen. Da das aus dem Record Linkage resultierende Ergebnis ebenfalls der Normform entspricht, kann das Ergebnis mit weiteren Dateien verglichen werden. Dadurch können theoretisch unendlich viele Datensätze miteinander verbunden werden.
[36] Vgl. Schulz 2017.

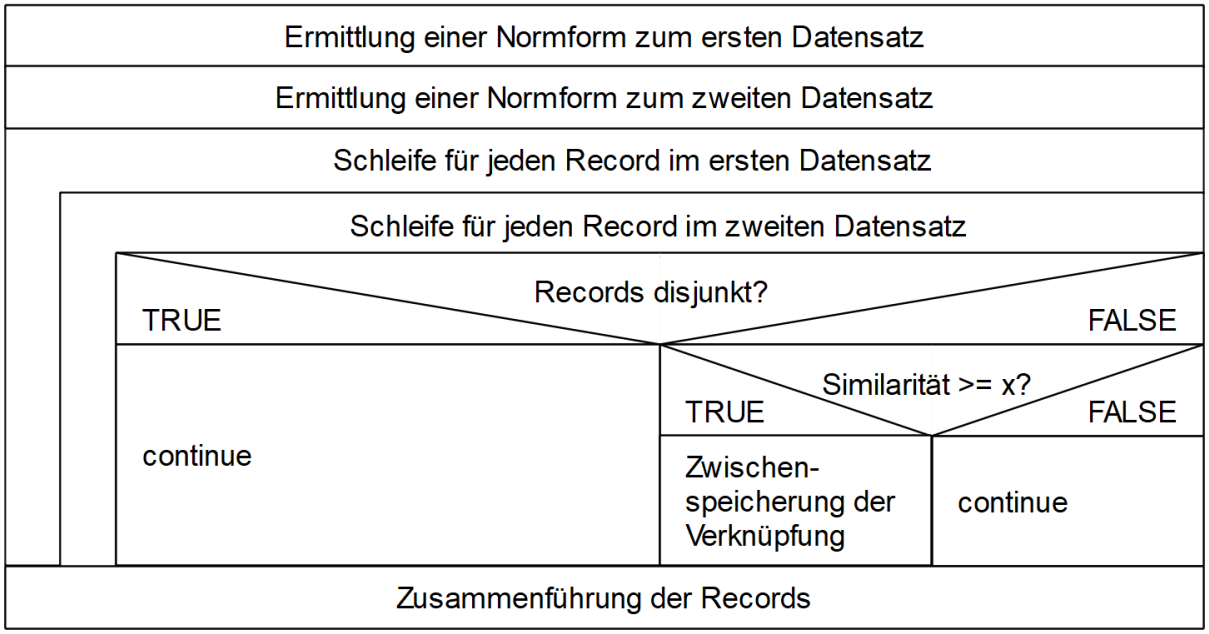

Abb. 2: Funktionsweise des Algorithmus als Nassi-Shneiderman-Diagramm. [Goldberg / Mernitz 2023]

3.2 Definition der Normform

Um Daten in eine Normform zu überführen, ist die Definition einer solchen notwendig. Das umfasst (1.) die Definition eines Formats und (2.) die Definition des Inhalts (die möglichen Schlüssel der Variablen / Attribute). Zum Format wird festgelegt, dass es sich bei der Normform um eine CSV-Datei handelt. Dies stellt ein gängiges Format zur Darstellung von tabellarischen Informationen dar. Als Trennzeichen wird der Tabstopp festgelegt. Jede Zeile stellt einen Record dar. Bei der Definition des Spalteninhalts ist darauf zu achten, dass sie bestmöglich einem intuitiven Verständnis entspricht (vgl. Tabelle 1). Auch wenn der Inhalt zwar definiert wird, ist nicht davon auszugehen, dass in jedem Fall vor einer Eintragung von Daten zunächst die Beschreibung studiert wird.

Bezeichnung	**Inhalt**
id	Diese Spalte enthält eine Abfolge von Zeichen, die innerhalb des Datensatzes einmalig je Eintrag ist. Falls die Spalte in einem Datensatz nicht vorhanden ist, so wird diese nachträglich erzeugt und allen Einträgen wird eine eindeutige ID zugeordnet. Es ist darauf zu achten, dass Tabellen aus unterschiedlichen Quellen auch unterschiedliche IDs aufweisen.
firstnameGiven	Diese Spalte enthält die Vornamen. Sind mehrere Vornamen vorhanden, so sind diese mit einem Leerzeichen voneinander zu trennen.
firstnameChange	Diese Spalte enthält Informationen über die Änderung des Vornamens. Es handelt sich also um einen alternativen Vornamen.
sex	Diese Spalte enthält eine Information über das Geschlecht (›F‹ für weiblich, ›M‹ für männlich und eine leere Zelle für unbestimmte Geschlechter).
surnameGiven	Diese Spalte enthält die Information über den Nachnamen bei der Geburt.
surnameChange	Diese Spalte enthält die Information über eine Änderung des Nachnamens nach der Geburt, aber vor der Heirat. Das kann beispielsweise dadurch erfolgen, dass eine Person adoptiert wird oder aber die Eltern nach der Geburt heiraten.

Tab. 1: Definition von Datenfeldern. [Goldberg / Mernitz 2023]

surnameMarriage1, surnameMarriage2, surnameMarriage3	Diese Spalte enthält die Änderung des Nachnamens im Zuge einer ersten, zweiten oder dritten Hochzeit. Wenn im Zuge der Heirat keine Namensänderung stattgefunden hat, bleibt sie leer.
surnameUnknown	Diese Spalte enthält den Nachnamen, wenn nicht klar ist, zu welchem Ereignis diesen jemand erlangt hat.
birthday	Diese Spalte enthält den Tag der Geburt. Hier ist nur der Tag in dem Format DD.MM.YYYY einzutragen, ohne eine weitere Spezifikation der Uhrzeit. Die GEDCOM-Systematik zur Beschreibung ungenauer Zeitpunkte ist anzuwenden (z. B. ›BET … AND …‹ für ein Ereignis in einer Zeitspanne).
birthplace	Diese Spalte enthält den Ort der Geburt. Hier ist nur die Stadt anzugeben, keine weiteren Adressen.
birthplaceGOV	Diese Spalte enthält die GOV-Kennung (Geschichtliches Orts-Verzeichnis, siehe unten) des Geburtsortes.
growthUpPlace	Diese Spalte enthält Informationen über die Herkunft einer Person, wenn der Geburtsort nicht näher zu bestimmen ist. Beispielhaft dafür sind Angaben wie »aus […]«. Auch kann der Geburtsort von dem Wohnort der Eltern abweichen. Letzterer ist hier einzutragen.
growthUpPlaceGOV	Diese Spalte enthält die GOV-Kennung des Herkunftsortes.
baptismday	Diese Spalte enthält den Tag der Taufe. Hier ist nur der Tag in dem Format DD.MM.YYYY einzutragen, ohne eine weitere Spezifikation der Uhrzeit. Die GEDCOM-Systematik zur Beschreibung ungenauer Zeitpunkte ist anzuwenden (z. B. ›BET … AND …‹ für ein Ereignis in einer Zeitspanne).
baptismplace	Diese Spalte enthält den Ort der Geburt. Hier ist ein Ort einzutragen und nicht die entsprechende Kirche. Hier ist nur die Stadt anzugeben, keine weiteren Adressen.
baptismplaceGOV	Diese Spalte enthält die GOV-Kennung des Taufortes.
marriageday1, marriageday2, marriageday3	Diese Spalte enthält den Tag der ersten, zweiten oder dritten Hochzeit. Hier ist nur der Tag in dem Format DD.MM.YYYY einzutragen, ohne eine weitere Spezifikation der Uhrzeit. Die GEDCOM-Systematik zur Beschreibung ungenauer Zeitpunkte ist anzuwenden (z. B. ›BET … AND …‹ für ein Ereignis in einer Zeitspanne).
marriageplace1, marriageplace2, marriageplace3	Diese Spalte enthält den Ort der ersten, zweiten oder dritten Heirat. Hier ist nur die Stadt anzugeben, keine weiteren Adressen.
marriageplaceGOV1, marriageplaceGOV2, marriageplaceGOV3	Diese Spalte enthält die GOV-Kennung des ersten, zweiten oder dritten Heiratsorts.
ageAtMarriage1, ageAtMarriage2, ageAtMarriage3	Diese Spalte enthält Angaben zum Alter bei der ersten, zweiten oder dritten Hochzeit in Jahren.
idSpouse1, idSpouse2, idSpouse3	Diese Spalte enthält die ID des*der ersten, zweiten oder dritten Ehepartner*in in dem gleichen Datensatz.
divorceday1, divorceday2, divorceday3	Diese Spalte enthält den Tag der ersten, zweiten oder dritten Scheidung. Hier ist nur der Tag in dem Format DD.MM.YYYY einzutragen, ohne eine weitere Spezifikation der Uhrzeit. Die Gedbas4All-Systematik zur Beschreibung ungenauer Zeitpunkte ist anzuwenden.
deathday	Diese Spalte enthält den Tag des Todes. Hier ist nur der Tag in dem Format DD.MM.YYYY einzutragen, ohne eine weitere Spezifikation der Uhrzeit. Die GEDCOM-Systematik zur Beschreibung ungenauer Zeitpunkte ist anzuwenden (z. B. ›BET … AND …‹ für ein Ereignis in einer Zeitspanne).
deathplace	Diese Spalte enthält den Ort des Todes. Hier ist nur die Stadt anzugeben, keine weiteren Adressen.
deathplaceGOV	Diese Spalte enthält die GOV-Kennung des Todesorts.

Tab. 1: Definition von Datenfeldern. [Goldberg / Mernitz 2023]

causeOfDeath	Diese Spalte enthält die Todesursache. Verschiedene Todesursachen sind mit Komma und nachfolgendem Leerzeichen oder einem ›und‹ mit vor- und nachstehendem Leerzeichen abzugrenzen.
maritalStatusAtDeath	Diese Spalte enthält eine Information über den Familienstand beim Tod. Eine Benennung als Witwer beispielsweise kann darauf hindeuten, dass die Frau früher verstorben sein muss.
ageAtDeath	Diese Spalte enthält eine Information über das Lebensalter beim Tod.
burialday	Diese Spalte enthält den Tag der Beerdigung. Hier ist nur der Tag in dem Format DD.MM.YYYY einzutragen, ohne eine weitere Spezifikation der Uhrzeit. Die GEDCOM-Systematik zur Beschreibung ungenauer Zeitpunkte ist anzuwenden (z. B. ›BET … AND …‹ für ein Ereignis in einer Zeitspanne).
burialplace	Diese Spalte enthält den Ort der Beerdigung. Hier ist nur die Stadt anzugeben, keine weiteren Adressen.
burialplaceGOV	Diese Spalte enthält die GOV-Kennung des Beerdigungsortes.
occupation	Diese Spalte enthält Informationen zum Beruf. Verschiedene Berufsangaben sind mit Komma und nachfolgendem Leerzeichen oder einem ›und‹ mit vor- und nachstehendem Leerzeichen abzugrenzen.
idFather	Diese Spalte enthält die ID des Vaters innerhalb dieses Datensatzes.
idMother	Diese Spalte enthält die ID der Mutter innerhalb dieses Datensatzes.

Tab. 1: Definition von Datenfeldern. [Goldberg / Mernitz 2023]

Die Normform enthält dabei nicht alle möglichen Bestandteile prosopographischer Quellen. Daneben sind weitere Charakteristika denkbar, die sich auf das Leben von Personen beziehen und in prosopographischen Quellen vorkommen (u. a. Taufpaten, Trauzeugen, Täufer, weitere Bezugspersonen, Adressen zu bestimmten Zeitpunkten, Quellenangaben, Angaben zu weiteren religiösen Feierlichkeiten wie der Firmung oder Konfirmation). Da es hier aber unwahrscheinlich ist, dass diese Informationen in zwei Datensätzen vorkommen, die auf verschiedenen Quellen basieren und diese teilweise zudem automatisiert schwer zu vergleichen wären, finden diese keinen Einzug. Es kann im Einzelfall jedoch essenziell sein, diese Informationen zu ergänzen und den Algorithmus dahingehend zu erweitern.

3.3 Datenbereinigung und -strukturierung

Trotz der Normform können die Daten nicht immer direkt miteinander in einen Vergleich gesetzt werden. Es ist eine weitere Bereinigung des Inhalts notwendig. Darunter gehört z. B. die Veränderung des Datumsformats. Ferner betrifft die Bereinigung insbesondere die Vornamen (siehe Abschnitt 3.3.1, ›Aufbereitung der Namen‹). Sind mehrere Vornamen vorhanden, so werden diese in einer Liste voneinander separiert. Ebenso werden die Berufsangaben aufbereitet (siehe Abschnitt 3.3.2, ›Aufbereitung der Berufsangaben‹). Auch hier werden mehrere Berufe voneinander getrennt. In einem folgenden Schritt werden die Datumsfelder zur Geburt, Taufe, Heirat, dem Tod oder der Beerdigung korrigiert (siehe Abschnitt 3.3.3, ›Aufbereitung der Zeitangaben‹). Die Bereinigung von Ortsangaben dahingegen ist derzeit nicht implementiert, kann aber ergänzt werden.[37]

[37] Ortsangaben unterliegen einer breit gefächerten Variation. Insbesondere, ob und wie übergeordnete administrative Einheiten in die Angabe mit eingebunden werden, ist in der Praxis uneinheitlich. Hierbei ist die Verwendung von eindeutigen Identifikatoren für Orte sehr hilfreich. Als Identifikatoren für Orte sind die IDs des Geschichtlichen Orts-Verzeichnis (GOV) zu empfehlen. Vgl. Verein für Computergenealogie 2021. Die Datenbank des Vereins für Computergenealogie bildet hier insbesondere für den deutschen Sprachraum eine geeignete Repräsentation tatsächlich (vormals) vorhandener Orte. Aufgrund einer langen Zeit geringer Mobilität insbesondere der ländlichen Bevölkerung ist es wahrscheinlicher, dass Lebensereignisse in einer begrenzten geografischen Distanz stattgefunden haben. Vgl. Bähr et al. 1992; Kocka et al. 1980. Für den Erfolg eines Record Linkage kann es also auch relevant sein, ob Orte geografisch nah beieinander zu finden sind. Vgl. Efremova et al. 2015, S. 135, 139–141. Die Aufbereitung der Ortsangaben kann an den von Goldberg definierten, auf den deutschen Sprachraum abgestimmten Kriterien orientiert sein. Vgl. Goldberg 2022. Über das von Goldberg beschriebene Programm kann auch eine automatische Zuweisung der GOV-IDs stattfinden.

3.3.1 Aufbereitung der Namen

Namensbezeichnungen können verschiedene Eigenschaften besitzen, die ein Record Linkage erschweren. Ein Beispiel dafür sind Abkürzungen (unvollständige Bezeichnungen, die mit einem Punkt abschließen). Abkürzungen können dabei sehr individuell ausgestaltet sein, aber auch eine große Intersubjektivität besitzen. Der Algorithmus enthält eine Reihe üblicher Abkürzungen für Namen. Hier zeigt sich ein weiterer Aspekt der Anpassung der Lösung an die deutsche Sprache. Je nach zu bearbeitenden Quellen kann es der Qualität des Ergebnisses dienlich sein, diese Liste zu erweitern oder anzupassen. Zur weiteren Aufbereitung werden auch Klammern entfernt, die in Vornamensnennungen vorkommen können, beispielsweise um Aliasnamen wie etwa »Hans Joseph (Franz)« darzustellen. Mehrere Vornamen werden durch Leerzeichen separiert als Liste gespeichert.

Um den Nutzen der Vornamen für das Record Linkage zu erhöhen, wird aus den Angaben zum Vornamen das Geschlecht erkannt – sofern diese Information nicht gesondert vorliegt. Hierzu werden die Vornamen, die auf ein A oder E enden, als weiblich erkannt. Dazu wird jeweils der erste Vorname herangezogen.[38] Etliche Ausnahmen sind gesondert definiert (z. B. Ingeborg, Elisabeth).

3.3.2 Aufbereitung der Berufsangaben

Ähnlich wie bei den Namen können auch Berufsangaben eine Abkürzung erfahren. Auch diese werden mit Hilfe einer initial definierten Liste aufgelöst und ausgeschrieben. Die uneindeutige Verwendung von Abkürzungen stellt hier im Vergleich zu den Vornamen jedoch ein größeres Problem dar. Das betrifft besonders sehr allgemeine Kürzel, beispielsweise die Abkürzung »K.«, die sowohl auf einen Knaben als auch einen Kaufmann hindeuten oder möglicherweise auch eine andere Bedeutung haben kann. Auch kann die Berufsangabe nicht nur Angaben zur beruflichen Tätigkeit, sondern weitergehende Informationen über den Rechtsstatus, Wohnsitz oder einen Zeitbezug enthalten.[39] Mehrere Berufsangaben werden anhand des Kommas oder eines ›und‹ aufgesplittet als Liste gespeichert.

3.3.3 Aufbereitung der Zeitangaben

Zeitangaben können verschiedene Formate aufweisen. Das liegt vor allem in dem Umstand begründet, dass Zeitangaben nicht immer ein konkretes, taggenaues Datum bezeichnen, sondern zum Beispiel auch einen Zeitraum benennen können. Im Algorithmus wird davon ausgegangen, dass die Datumsangaben bereits in die Normform umgewandelt und im Format DD.MM.YYYY vorliegen. Eine Ausnahme betrifft Zeiträume, die im GEDCOM-Format ›BET … AND …‹ formatiert werden. Hier wird die vordere Grenze des Zeitraumes für die weitere Berechnung herangezogen.

3.4 Formalisierung von Heuristiken zum Vergleich von Records

Genealogische Heuristiken helfen dabei, die Records zu identifizieren, die dieselbe Entität beschreiben. Ihre Formalisierung führt zu Logikoperationen, die programmtechnisch realisiert werden können. Dabei basieren diese Vergleiche auf den vorhandenen Variablen. Jedoch können schon bei einem Datensatz mit 30 verschiedenen zu vergleichenden Variablen (Variable vorhanden / nicht vorhanden) insgesamt etwa eine Milliarde mögliche Kombinationen auftreten.[40] Der Vergleich von zwei Datensätzen erhöht diese Zahl der möglichen Kombinationen auf mehr als eine Trillion.[41] Für diese Anzahl an Kombinationen ist eine manuelle Definition von Verarbeitungsfolgen nicht vorstellbar. Vielmehr muss diese sinnvoll reduziert werden. Dieses wird erreicht, indem Kombinationen von Variablen ausgeschlossen werden. Beispielhaft lässt ein Vergleich zwischen Sterbeort und Berufsangabe allein voraussichtlich keinen Schluss auf den Zusammenhang von Records zu.

Hierzu können zunächst verschiedene Variablen zusammengefasst werden, die ähnliche Merkmale aufweisen (z. B. Datumsangaben, Ortsangaben, Namen). Vergleiche sind nur innerhalb dieser Gruppen sinnhaft. Diese Definition geschieht im ersten Unterabschnitt (›Definition zu vergleichender Variablen‹). Im zweiten Unterabschnitt (›Disjunktionen‹) werden

[38] In der deutschen Sprache enden Frauennamen traditionell auf A oder E. Zwar tragen auch vereinzelte Männer Frauennamen, häufig Maria, diesen jedoch kaum als ersten Vornamen. Auf die moderne Namensgebung passt dieses Muster nicht mehr. Da sich dieser Algorithmus aber auf historische Daten bezieht, stellt das an dieser Stelle kein entscheidendes Problem dar.
[39] Zur Separierung solcher berufsfernen Angaben kann auf Goldberg / Moeller 2022 hingewiesen werden, die Kriterien zur Bereinigung von Berufsangaben aufstellen.
[40] 2^{30} = 1.073.741.824.
[41] $1.073.741.824^2$ = 1.152.921.504.606.850.000.

Disjunktionsregeln beschrieben: Wenn z. B. eine Taufe nach dem Tod stattfindet, dann ist eine Similarität auszuschließen.[42] Es bleibt eine deutlich minimierte Anzahl an Variablenkombinationen übrig, bei denen ein genauerer Vergleich sinnhaft erscheint. Im dritten Unterabschnitt (›Similaritätsprüfung‹) wird dann der Similaritätsvergleich zwischen zwei Records beschrieben, die nicht disjunkt sind.

3.4.1 Definition zu vergleichender Variablen

Eine Gruppe von Vergleichen kann vorgenommen werden, wenn in beiden Records gleichartige Variablen vorliegen. Dazu ist ein Wissen über die Beziehungen der Variablen untereinander relevant. Hiervon sind insbesondere Zeit- und Nachnamensangaben betroffen. Bei Zeitangaben sind die zeitlichen Relationen zwischen Geburts-, Tauf-, Heirats-, Sterbe- und Beerdigungsdatum relevant. Hierbei ist auch ein Vergleich zu den Lebenszeitangaben der potenziellen Eltern von Interesse. Nachnamen sind von der Schwierigkeit betroffen, dass sie im Lebensverlauf starken Veränderungen unterliegen können. Besonders Frauen wechselten häufig bei Hochzeiten ihre Namen, sodass es keine Seltenheit darstellt, wenn Personen im Lebensverlauf mit drei oder vier verschiedenen Nachnamen erscheinen. Deshalb ist ein Vergleich sowohl mit dem Geburtsnamen als auch mit den Ehenamen relevant. Auch Ortsangaben können relevant sein, weil es wahrscheinlicher ist, dass verschiedene Lebensereignisse in einem begrenzten geografischen Radius stattfinden. Da es sich hierbei jedoch um eine vergleichsweise ungenaue Bestimmung handelt, ist diese im bisherigen Algorithmus nicht eingebunden. Sie ist dennoch aufgeführt, um eine Hilfestellung für eine Erweiterung zu bieten. Im Folgenden werden die Vergleichsgruppen dargestellt und grundsätzliche Vergleiche eingegrenzt:

- Vornamensvergleiche: firstnameGiven, firstnameChange
    - Die (teilweise) Übereinstimmung von Vornamen kann Aufschluss über die Zusammenführung der Personen liefern.[43]
- Geschlechtsvergleiche: sex
    - Gleiche Personen weisen das gleiche Geschlecht auf.
- Nachnamensvergleiche: surnameUnknown, surnameGiven, surnameChange, surnameMarriage1, surnameMarriage2, surnameMarriage3
    - Die (teilweise) Übereinstimmung von Nachnamen kann Aufschluss über die Zusammenführung von Personen liefern, wobei die Übereinstimmung von Nachnamen in unterschiedlichen Kategorien nur bei surnameUnknown ein Indiz für eine Übereinstimmung ist.[44]
- Datumsvergleiche: birthday, baptismday, marriageday1, ageAtMarriage1, divorceday1, marriageday2, ageAtMarriage2, divorceday2, marriageday3, ageAtMarriage3, divorceday3, deathday, ageOfDeath, burialday
    - birthday und baptismday: Taufdatum und Geburtsdatum liegen oft nah beieinander.[45] Eine Person kann nicht vor ihrer Geburt getauft werden.
    - ageAtMarriage1, ageAtMarriage2, ageAtMarriage3 und birthday, marriageday1, marriageday2, marriageday3: Das Alter bei der Heirat und das errechnete Alter sollten nahe beieinanderliegen.
    - marriageday1, marriageday2, marriageday3 und birthday: Eine Person muss bei einer Heirat ein Mindestalter erreicht haben.
    - divorceday1, divorceday2, divorceday3 und birthday: Eine Person muss bei einer Scheidung ein Mindestalter erreicht haben.
    - ageAtDeath und birthday, deathday: Das beim Tod errechnete Alter und das Geburtsdatum dürften nur endlich weit auseinanderliegen. Eine Person kann nicht vor ihrer Geburt sterben. Totgeburten und schnelle Todesfälle nach der Geburt können am Geburtstag auftreten.
    - birthday, deathday und ageOfDeath: Die Differenz zwischen einem errechneten Alter und dem angegebenen Alter bei Tod muss gering sein.
    - birthday, burialday und ageOfDeath: Die Differenz zwischen einem errechneten Alter und Berücksichtigung der Angabe des Beerdigungsdatums und dem angegebenen Alter bei Tod muss gering sein.
    - ageAtMarriage1, ageAtMarriage2, ageAtMarriage3 und baptismday, marriageday1, marriageday2, marriageday3: Das Alter bei der Heirat und das errechnete Alter sollten nahe beieinanderliegen.
    - marriageday1, marriageday2, marriageday3 und baptismday: Eine Person muss bei einer Heirat ein Mindestalter erreicht haben.
    - divorceday1, divorceday2, divorceday3 und baptismday: Eine Person muss bei einer Scheidung ein Mindestalter erreicht haben.

[42] Sonderformen bei einzelnen Glaubensgemeinschaften, z. B. die Totentaufe der Mormonen, bleiben unberücksichtigt.
[43] Der Vergleich darf sich aber nicht nur auf einzelne Vornamen oder die Reihenfolge der Vornamen beziehen. Beispielsweise können »Johann« und »Johann Christoph« dieselbe Person sein, »Johann Christoph« und »Christoph Johann« können dieselbe Person sein, »Johann Christoph« und »Christoph Heinrich« sind aber eher unwahrscheinlich dieselbe Person.
[44] Beispielsweise ist eine Person, die als surnameGiven »Schwarzenberg« aufweist, nur in seltenen Fällen mit einer Person übereinstimmend, die diesen Namen durch die erste Heirat (surnameMarriage1) erhalten hat.
[45] Die hier definierten Regeln passen nur auf solche Religionsgemeinschaften, die die Kleinkindtaufe praktizieren.

    - ageAtDeath und baptismday, deathday: Das beim Tod errechnete Alter und das Taufdatum dürften nur endlich weit auseinanderliegen. Eine Person kann nicht vor ihrer Taufe sterben. Allerdings sind Nottaufen möglich, die am Todestag erfolgen.
    - baptismday, deathday und ageOfDeath: Die Differenz zwischen einem errechneten Alter und dem angegebenen Alter bei Tod muss gering sein.
    - baptismday, burialday und ageOfDeath: Die Differenz zwischen einem errechneten Alter und Berücksichtigung der Angabe des Beerdigungsdatums und dem angegebenen Alter bei Tod muss gering sein.
    - marriageday1, marriageday2, marriageday3 und deathday: Die Hochzeit erfolgt vor dem Tod.
    - divorceday1, divorceday2, divorceday3 und deathday: Die Scheidung erfolgt vor dem Tod.
    - marriageday1, marriageday2, marriageday3 und burialday: Die Hochzeit erfolgt vor der Beerdigung.
    - divorceday1, divorceday2, divorceday3 und burialday: Die Scheidung erfolgt vor der Beerdigung.
    - divorceday1, divorceday2, divorceday3, deathday: Die Scheidung erfolgt vor dem Tod.
    - deathday und burialday: Eine Person kann nicht vor ihrem Tod beerdigt werden. Beerdigungsdatum und Todesdatum liegen nah beieinander.
- Ortsstringvergleiche: birthplace, growthUpPlace, baptismplace, marriageplace1, marriageplace2, marriageplace3, deathplace, burialplace
    - Gleiche oder ähnliche Ortsangaben weisen auf gleiche Personen hin. Das kann durch eine exakte Übereinstimmung der Strings oder eine starke Ähnlichkeit erkannt werden.
    - Die Wahrscheinlichkeit für ein Match ist höher, wenn beispielsweise Geburtsort und Heiratsort der gleiche sind.
- Ortsentfernungsvergleiche: birthplaceGOV, growthUpPlaceGOV, baptismplaceGOV, marriageplaceGOV1, marriageplaceGOV2, marriageplaceGOV3, deathplaceGOV, burialplaceGOV
    - growthUpPlaceGOV, birthplaceGOV: Wenn Herkunft und Geburtsort nah beieinanderliegen, erhöht dieses die Wahrscheinlichkeit, dass es sich um die gleiche Person handeln kann. Das wird über die Koordinaten in den GOV-Elementen ermittelt.
    - growthUpPlaceGOV, baptismplaceGOV: Wenn Herkunft und Taufort nah beieinander liegen erhöht dieses die Wahrscheinlichkeit, dass es sich um die gleiche Person handeln kann. Das wird über die Koordinaten in den GOV-Elementen ermittelt.
- Variablen, die nur mit sich selbst verglichen werden können:
    - causeOfDeath: Wenn in zwei Quellen die Todesursache angegeben ist und diese gleich oder ähnlich ist, erhöht dieses die Wahrscheinlichkeit, dass es sich um dieselbe Person handelt.
    - occupation: Wenn in zwei Quellen eine Berufsangabe gegeben ist und diese gleich oder ähnlich ist, erhöht dieses die Wahrscheinlichkeit, dass es sich um dieselbe Person handelt. Berufsangaben können sich dabei im Verlauf eines Lebens jedoch ändern. Auch kann derselbe Beruf unter Bezeichnungen angegeben werden, die sich nicht ähnlich sind und dadurch nur schwer über String-Matching-Methoden erkannt werden können (z. B. »Feuerwehrmann« und »Hauptbrandmeister«).
- source: Wenn zwei Personen in derselben Quelle genannt werden, wird hier angenommen, dass es sich nicht um dieselbe Person handelt. Dabei sind detaillierte Quellen gemeint (z. B. ein konkreter Heiratseintrag mit laufender Nummer in einem Heiratsregister).

3.4.2 Disjunktionen

Sind im vorigen Abschnitt mögliche Vergleiche zwischen Variablen beschrieben worden, findet nun eine Definition konkreter Kriterien statt, die ein Record Linkage verhindern. Dazu wird zunächst erkannt, ob zwei Records disjunkt sind, also nicht dieselbe Entität beschreiben. In dem Fall erhalten sie einen Similaritätswert von 0. Disjunkte Einträge werden vom Algorithmus nicht weiter behandelt. Die Disjunktionsregeln werden hier oberflächlich textuell beschrieben und dann stärker formalisiert und übersichtlicher dargestellt. In der programmtechnischen Umsetzung wird darauf geachtet, jene Regeln, die besonders viele Kombinationen ausschließen, an den Beginn zu setzen. Dies ändert zwar das Ergebnis nicht, führt jedoch zu einer erheblichen Verbesserung der Laufzeit.

Die meisten hier vorgestellten Regeln sind in Hinblick auf die kulturelle Praxis und den Ablauf von Lebensereignissen logisch. So kann eine Person beispielsweise vor ihrer Geburt nicht sterben. Bisher wurden solche Regeln für den deutschsprachigen Raum wissenschaftlich noch nicht beschrieben. Vielmehr finden sich zahlreiche Publikationen zur Genealogie, die insbesondere Privatpersonen einen Zugang ermöglichen, aber wissenschaftlichen Standards nicht entsprechen und auf die deshalb hier kein Bezug genommen wird. Die ›kulturelle Praxis‹ für den deutschsprachigen Raum basiert dabei vielmehr auf der jahrelangen Erfahrung der Autoren im Umgang mit genealogischen Daten.

Zunächst sind Records disjunkt, wenn sie auf demselben Eintrag in einer Quelle basieren. Das kann beispielsweise in Taufeinträgen der Fall sein, bei denen Vater und Sohn die gleichen Namen haben, niemals aber dieselbe Person darstellen. Auch wenn bei einem Eintrag kein Vorname oder kein Nachname vorhanden ist, wird für diesen Algorithmus definiert, dass kein Record Linkage erfolgen kann und die Einträge werden so behandelt, als wären sie disjunkt. Alle Kinder, die vor dem Alter von 13 Jahren verstorben sind, erhalten ebenfalls eine 0. Hier besteht die Annahme, dass diese vor diesem Alter noch nicht in anderen Einträgen vorkommen können und ein weiterer Vergleich aus Laufzeitgründen deshalb nicht notwendig ist.[46] Wenn beide Records ein Geschlecht aufweisen, dieses aber nicht dasselbe ist, so sind sie disjunkt. Personen können nicht vor ihrer Geburt getauft oder beerdigt werden, heiraten oder sterben. Sie können auch nicht vor ihrer Heirat sterben oder beerdigt werden. Auch können sich Personen nicht scheiden lassen, bevor sie geheiratet haben. In der programmtechnischen Umsetzung existieren Variablen für bis zu drei Eheschließungen. Dies kann jedoch beliebig erweitert werden. Eine Hochzeit kann nicht nach dem Tod oder der Beerdigung stattfinden. Ebenso kann eine Person maximal ein Alter von 120 Jahren erreichen. Wenn kein Geburtsdatum vorhanden ist, wird jeweils das Taufdatum für den Vergleich herangezogen. Auch ersetzt das Beerdigungsdatum den Sterbetag, sofern dieser fehlt. Im Übrigen muss eine Person erst sterben, bevor sie beerdigt werden kann.

Wenn die Geburtsdaten beider Personen vorhanden und trotzdem unterschiedlich sind, so beschreiben sie nicht dieselbe Person. Ebenso verhält es sich mit den Sterbedaten. Bei den Taufzeitpunkten sind die Einträge nicht disjunkt, solange die Taufdaten eine Differenz von drei Jahren nicht überschreiten. Die drei Jahre stellen dabei eine Annahme dar, die genügend Platz für Abweichungen lässt.

Aus dem Vergleich mit den Eltern ergeben sich einige Zustände, die ein ausschließendes Kriterium darstellen. So kann der Tod des eigenen Vaters maximal neun Monate vor der eigenen Geburt stattfinden, der Tod der Mutter nicht vor der Geburt. Da die Taufen in den historischen Daten oftmals wenige Tage nach der Geburt vollzogen worden sind, gilt die gleiche Regel auch für die Taufdaten (der Tod der Mutter kann jedoch vor der Taufe des Kindes eintreten, wenn sie bei der Geburt verstirbt). Es wird zudem ein Mindestalter für eine Elternschaft von 13 Jahren angenommen. Diese Grenze wird auch als Mindestalter für eine Hochzeit oder Scheidung gewählt. Zudem wird definiert, dass Frauen maximal mit 60 Jahren noch Mutter werden können.

Folgende Regeln führen zur Ungleichheit der Records (similarity = 0):

- Wenn sex ! = sex
- Wenn source = = source
- Wenn Differenz von birthday von id und deathday von idFather > 9 Monate
- Wenn Differenz von baptismday von id und deathday von idFather > 9 Monate
- Wenn Differenz von birthday von id und burialday von idFather > 9 Monate
- Wenn Differenz von baptismday von id und burialday von idFather > 9 Monate
- Wenn birthday von id > deathday von idMother[47]
- Wenn birthday von id > burialday von idMother
- Wenn Differenz von birthday von id und birthday von idFather > 13 Jahre
- Wenn Differenz von baptismday von id und birthday von idFather > 13 Jahre
- Wenn Differenz von birthday von id und baptismday von idFather > 13 Jahre
- Wenn Differenz von baptismday von id und baptismday von idFather > 13 Jahre
- Wenn Differenz von birthday von id und birthday von idMother > 13 Jahre
- Wenn Differenz von baptismday von id und birthday von idMother > 13 Jahre
- Wenn Differenz von birthday von id und baptismday von idMother > 13 Jahre
- Wenn Differenz von baptismday von id und baptismday von idMother > 13 Jahre
- Wenn Vornamen vorhanden und kein Vorname mit einem anderen übereinstimmt
- Wenn Differenz baptismday und birthday > 3 Jahre
- Wenn Differenz ageAtMarriage und errechnetes Alter durch birthday, marriageday > 5 Jahre
- Wenn Differenz ageAtMarriage und errechnetes Alter durch baptismday, marriageday > 5
- Wenn errechnetes Alter durch birthday, marriageday < 13 Jahre oder > 100 Jahre
- Wenn errechnetes Alter durch birthday, divorceday < 13 Jahre oder > 100 Jahre
- Wenn errechnetes Alter durch baptismday, marriageday < 13 Jahre oder > 100 Jahre
- Wenn errechnetes Alter durch baptismday, divorceday < 13 Jahre oder > 100 Jahre
- Wenn Differenz ageAtDeath und errechnetes Alter durch birthday, deathday > 10
- Wenn Differenz ageAtDeath und errechnetes Alter durch baptismday, deathday > 10

[46] Wenn für die zu vergleichenden Quellen jedoch insbesondere dieser Aspekt relevant ist, kann die Altersgrenze auch variiert oder entfernt werden. Das kann zum Beispiel der Fall sein, wenn Geburtsangaben aus Zeitungen mit denen aus Kirchenbüchern verglichen werden sollen.

[47] Auf diese Regel unter Einbeziehung des Taufdatums wird hier verzichtet, weil die Mutter bei der Geburt sterben und das Kind erst danach getauft werden kann.

- Wenn Differenz ageAtDeath und errechnetes Alter durch birthday, burialday > 10
- Wenn Differenz ageAtDeath und errechnetes Alter durch baptismday, burialday > 10
- Wenn birthday > baptismday
- Wenn birthday > marriageday1
- Wenn birthday > divorceday1
- Wenn birthday > marriageday2
- Wenn birthday > divorceday2
- Wenn birthday > marriageday3
- Wenn birthday > divorceday3
- Wenn birthday > deathday
- Wenn birthday > burialday
- Wenn baptismday > marriageday1
- Wenn baptismday > divorceday1
- Wenn baptismday > marriageday2
- Wenn baptismday > divorceday2
- Wenn baptismday > marriageday3
- Wenn baptismday > divorceday3
- Wenn baptismday > deathday
- Wenn baptismday > burialday
- Wenn marriageday1 > marriageday2
- Wenn marriageday1 > marriageday3
- Wenn marriageday1 > divorceday1
- Wenn marriageday1 > deathday
- Wenn marriageday1 > burialday
- Wenn marriageday2 > marriageday3
- Wenn marriageday2 > divorceday2
- Wenn marriageday2 > deathday
- Wenn marriageday2 > burialday
- Wenn marriageday3 > divorceday3
- Wenn marriageday3 > deathday
- Wenn marriageday3 > burialday
- Wenn divorceday1 > marriageday2
- Wenn divorceday1 > marriageday3
- Wenn divorceday1 > deathday
- Wenn divorceday1 > burialday
- Wenn divorceday2 > marriageday3
- Wenn divorceday2 > deathday
- Wenn divorceday2 > burialday
- Wenn divorceday3 > deathday
- Wenn divorceday3 > burialday
- Wenn Differenz deathday und burialday > 1 Jahr
- Wenn Differenz birthday und deathday > 120 Jahre
- Wenn Differenz birthday und burialday > 120 Jahre
- Wenn Differenz baptismday und deathday > 120 Jahre
- Wenn Differenz baptismday und burialday > 120 Jahre
- Wenn Differenz birthday und birthday > 1 Jahr
- Wenn Differenz baptismday und baptismday > 1 Jahr
- Wenn Differenz deathday und deathday > 1 Jahr
- Wenn Differenz burialday und burialday > 1 Jahr
- Wenn marriageday1 > deathday von idSpouse1
- Wenn marriageday2 > deathday von idSpouse2
- Wenn marriageday3 > deathday von idSpouse3
- Wenn divorceday1 > deathday von idSpouse1
- Wenn divorceday2 > deathday von idSpouse2
- Wenn divorceday3 > deathday von idSpouse3

Programmtechnisch sind die Vergleiche mit IF-ELSE-Anweisungen umgesetzt. Ferner ist ergänzend eine optionale Variable (sortingBySurnameGiven) angelegt, mit der im Fall identischer zu vergleichender Tabellen nur solche Personen zusammengeführt werden, deren surnameGiven mit demselben Anfangsbuchstaben beginnt. Diese Implementierung dieser optionalen Funktion erfolgt vorwiegend aus Laufzeitgründen für große Tabellen mit hunderttausenden Datensätzen.

3.4.3 Similaritätsprüfung

Kann nicht erkannt werden, dass zwei Records disjunkt sind, so wird die Similarität dieser weiter geprüft. Dazu wird ein Fuzzy-Vergleich der Vor- und Nachnamen vorgenommen. Zum Vergleich dieser Strings wird die Jaro-Winkler-Distanz ausgewählt, weil diese bei Georgala et al. zu guten Ergebnissen führt.[48] Georgala et al. erzielen mittels einer ROC-Kurve[49] ein optimales Ergebnis bei einem Grenzwert von 0,70.[50] Um die Anzahl der falschpositiven Zuordnungen zu verringern, wird in unserem Ansatz jedoch ein Grenzwert von 0,95 definiert. Nur wenn der Wert für die Nachnamen höher ist, wird davon ausgegangen, dass die Personen similär sind. Die Auswahl dieses Maßes und dieser Grenze ist jedoch keineswegs alternativlos, sondern kann im Programmcode verändert und ggf. auch an die Bedürfnisse der jeweiligen Anwendung angepasst werden. Alternativ zur reinen Jaro-Winkler-Distanz ist im Programmcode derzeit die phonetische Übereinstimmung auf Basis der Kölner Phonetik in Kombination mit einem anderen Grenzwert der Jaro-Winkler-Distanz implementiert. Diese wird getestet, wenn die Jaro-Winkler-Distanz den gewählten Grenzwert nicht überschreitet. Die Kölner Phonetik wird ausgewählt, da diese speziell auf den deutschen Sprachraum ausgerichtet ist. Buchstaben werden dabei in Zahlen codiert.[51] Ist der Wert der Kölner Phonetik gleich und liegt die Jaro-Winkler-Distanz bei über 0,60, wird hier ebenfalls von einer Similarität ausgegangen. Der Wert der Kölner Phonetik wird im Programmcode über die Bibliothek *kph* ermittelt. Für die Berechnung der Jaro-Winkler-Distanz wird hingegen die Bibliothek *distance* genutzt.

Nach dem Test der Nachnamen wird zudem die Similarität der Vornamen überprüft. Überschreitet die Jaro-Winkler-Distanz auch bei einem Vergleich der Vornamen einen Wert von 0,95, oder 0,60 in Kombination mit der Gleichheit der phonetischen Werte, wird als Similarität der arithmetische Mittelwert der Jaro-Winkler-Distanzen von Vor- und Nachnamen genutzt, um die Ähnlichkeit beider Records auszudrücken. Anderenfalls wird die Hypothese, dass die Records dieselbe Entität beschreiben, verworfen. Die Similarität erhält dann einen Wert von 0.

Die Similaritätsprüfung stützt sich im Algorithmus damit nur auf die Ähnlichkeit von Vor- und Nachnamen. Dabei können perspektivisch auch weitere Vergleiche integriert werden. So ist es denkbar, die Ähnlichkeit der Zeiten, der Ortsnamen, der Ortsentfernungen, der Berufe oder Todesursachen sowie eine Kombination dieser zu implementieren.

Wenn mehrere Matches vorhanden sind, wird geprüft, welches über die größte Übereinstimmung verfügt. Nur das passendste wird zusammengeführt. Es wird das mit dem besten Similaritätswert ausgewählt. Bestehen mehrere Matches mit dem gleichen Similaritätswert, so werden die Einträge ausgewählt, die zuerst zusammengeführt worden sind. Für die nicht ausgewählten Matches werden programmintern jedoch trotzdem globale IDs vergeben, weswegen nicht jede globale ID nachher auch in der Ergebnistabelle erscheint. Sollen mehr als zwei Matches zusammengeführt werden, muss das Programm mit der Ergebnistabelle wiederholt ausgeführt werden.

Neben der Similaritätsprüfung gibt es noch einen sogenannten Prioritätswert. Dieser wird ermittelt, um nicht nur Disjunktionsregeln und die Ähnlichkeit der Namen in der Similaritätsprüfung zu integrieren. Nur, weil zwei Records einen hohen Similaritätswert innehaben, bedeutet das nämlich noch nicht, dass sie tatsächlich die gleiche Person abbilden. Wenn alle anderen Variablen leer sind, reicht hier vielmehr die reine Namensgleichheit für einen hohen Similaritätswert aus. Records nur auf dieser Basis zusammenzuführen, ist nicht sinnvoll. Deswegen werden diese nur zusammengeführt, wenn sie zugleich verschiedene Variablenkombinationen aufweisen (z. B. beide ein Geburts- und Taufdatum), die die Disjunktionsprüfung überstanden haben. Darunter fallen folgende Ereignisse:

- Eine gleiche Berufsangabe (ausgenommen die Angabe »Bürger«)
- Einer der Hochzeitstage ist identisch
- Geburtsdatum oder Taufdatum bei beiden vorhanden
- Geburtsdatum oder Taufdatum und Todes- oder Beerdigungsdatum vorhanden
- Todesdatum oder Beerdigungsdatum bei beiden vorhanden

[48] Vgl. Georgala et al. 2015, S. 187.
[49] Receiver Operating Characteristic, vgl. Fan et al. 2006.
[50] Vgl. Georgala et al. 2015, S. 185.
[51] Vgl. Postel 1969, S. 928.

3.5 Zusammenführung von Records

Wird erkannt, dass zwei Records dieselbe Entität beschreiben, sind diese zusammenzuführen. Es wird ein neuer Record in einer neuen Tabelle kreiert, die ebenfalls die Normform besitzt. Dazu ist festzulegen, wie Daten zusammengeführt werden. Wenn jeweils gleiche Informationen vorhanden sind, wird die gemeinsame Information übernommen. Ist eine Variable in nur einem bekannten Datensatz beschrieben, so ist dieser Inhalt für den neuen Eintrag auszuwählen. Sind unterschiedliche Informationen vorhanden, so ist entweder die Information mit der höheren Aussagekraft zu übernehmen oder die Informationen ergänzen sich gegenseitig. Eine höhere Aussagekraft wird angenommen, wenn es beispielsweise statt einer Jahresangabe ein konkretes Datum gibt. Bei Namen oder Ortsangaben stellt der längere String die weitergehende Information dar. Bei Berufen und Quellenangaben werden beide Informationen beibehalten und mit einem Komma separiert zusammengeführt.

Die neue Tabelle enthält neben allen (wie oben beschrieben zusammengeführten) Variablen zudem die Spalte idGlobal. Diese globale ID stellt eine neu erzeugte ID dar, auf die sich alle weiteren ID-Verweise des zusammengeführten Datensatzes beziehen. Die Spalte ›id‹ der Normform wird ergo nicht zusammengeführt, sondern in der neuen Tabelle jeweils als ›idSource1‹ und ›idSource2‹ übernommen. Dies dient der erleichterten manuellen Qualitätskontrolle des Record Linkage. Tabelle 2 enthält die Beschreibung dieser Variablen.

Solche Records, zu denen kein Pendant im jeweils anderen Datensatz gefunden wird, werden unverändert in die neue Tabelle überführt. Ausnahme ist allerdings auch hierbei die Verwendung einer neuen ›globalId‹.

Bezeichnung	**Inhalt**
globalId	Diese Spalte enthält eine eindeutige, globale ID. Jede natürliche Person soll nur eine ID erhalten, die mit den einzelnen Einträgen der Datensätze verknüpft ist.
idSource1	Diese Spalte enthält die Angabe über die ID des ersten Eintrags in der ersten Quelle.
idSource2	Diese Spalte enthält die Angabe über die ID des zweiten Eintrags in der zweiten Quelle.

Tab. 2: Zusätzliche Variablen eines zusammengeführten Datensatzes. [Goldberg / Mernitz 2023]

4. Validierung am Beispiel Leipzigs

Leipzig ist eine Stadt, an der sich zwei große historische Handelsrouten Europas kreuzen: die Via Regia von Ost nach West sowie die Via Imperii von Nord nach Süd.[52] Diese geografische Lage bot für die Entwicklung Leipzigs, vor allem als Messe- und Handelszentrum, lange Zeit eine fruchtbare Grundlage. Mit der wirtschaftlichen Bedeutung Leipzigs ging auch ein Wachstum der Bevölkerung einher, zu dem noch heute in verschiedenen Quellen Zeugnisse erhalten sind. Aufgrund der vorhandenen prosopographischen Datenbestände mit umfangreichen genealogisch relevanten Informationen bietet Leipzig ein geeignetes Beispiel zur Validierung des beschriebenen Algorithmus. Innerhalb dieser Validierung werden zwei Quellen / Datenbestände betrachtet: die Kartei Leipziger Familien (KLF) und die Kartei Leipziger Kreisamtstestamente (KLK). Diese Datenquellen verbindet, dass sie zumindest teilweise Daten über dieselben Personen enthalten. Aufgrund des unterschiedlichen Gegenstands,[53] vor allem aber wegen unterschiedlicher Zeiträume, sind nicht alle Personen in beiden Datenbeständen zu finden. Zum Teil spielt auch eine unterschiedliche geografische Reichweite eine Rolle. Während die KLF auf den Innenstadtkern von Leipzig beschränkt ist, bezieht die KLK das Amt Leipzig mit ein.

In dem folgenden Abschnitt wird zunächst die Struktur der hier verwendeten Datenbestände beschrieben, bevor der Algorithmus auf sie angewendet wird. Die Validierung geschieht zum einen zwischen den Datenbeständen, aber auch innerhalb eines Datensatzes mit sich selbst. Das ist notwendig, da dieselben Personen auch dort doppelt erscheinen können und zunächst zusammengeführt werden müssen. Danach werden die Resultate dargestellt.

[52] Vgl. Schönfelder / Börngen 2015, S. 39.
[53] Bei der KLK ist vor allem relevant, dass nur ein Teil der Bevölkerung überhaupt Testamente hinterlegt hat.

4.1 Daten und Ermittlung der Normform

Im Folgenden wird zunächst auf die KLF eingegangen. Danach folgt die KLK.

4.1.1 Kartei Leipziger Familien (ca. 1550–1850)

In der KLF sind viele Informationen über in Leipzig ansässige Familien enthalten. Die Kartei wurde von einer Mitarbeiterin der Deutschen Zentralstelle für Genealogie, Helga Moritz, ab den 1950er Jahren erstellt. Als Grundlage nutzte sie die Leipziger Kirchen- und Bürgerbücher. Die Daten umfassen in etwa den Zeitraum von der Mitte des 16. bis zur Mitte des 19. Jahrhunderts. Auf 20.000 Karteikarten sind dort etwa 200.000 Personen(einträge) dokumentiert.[54] Die Karteikarten enthalten jeweils Angaben zu einem Ehemann, seiner Ehefrau und deren Kindern. Falls ein Mann zweimal heiratete, so sind beide Ehen auf einer Karte verzeichnet. Die Karteikarten sind untereinander nicht über eindeutige Identifikatoren wie Kartennummern verknüpft.[55]

Im Rahmen eines Datenerfassungsprojekts durch den Verein für Computergenealogie wurde die Kartei digitalisiert.[56] Dazu wurden die Scans der Karteikarten manuell abgetippt. Datenfelder im genutzten Datenerfassungssystem (DES) sind der Nachname (mit akademischen Titeln), die Vornamen, der Beruf, der Ort samt GOV-ID, das Geburtsdatum oder wahlweise Alter bei Tod, das Taufdatum, Heiratsdatum, Sterbedatum, Beerdigungsdatum und eine Bemerkung sowie ein Feld für weitere Ortsangaben und die ID der Karteikarte (die automatisch vergeben wird). Des Weiteren existieren besondere, KLF-spezifische Angaben zur Rolle, zur Bezugsperson und zur Art der Beziehung zur Bezugsperson.[57] Es gibt die Rollen Familienoberhaupt, Kind, Ehefrau und Drittperson. Ersteres beschreibt einen Mann, der die Karteikarte begründet, die Ehefrau ist seine Frau. Kinder einer Ehe sind als »Kind« klassifiziert. Drittpersonen können Ehepartner*innen von Kindern darstellen. Auch können Eltern von Personen, die nicht Kinder sind, als Drittpersonen auftauchen (insbesondere die Eltern der Ehepartner*innen). Jede Drittperson ist jeweils einer Bezugsperson zugeordnet. Ein*e Ehepartner*in eines Kindes beispielsweise ist diesem Kind zugeordnet. Die Art der Beziehung beschreibt dahingegen das Verhältnis zur Drittperson (Ehemann / Ehefrau / Vater). Damit sind die Felder nicht direkt der definierten Normform zuzuordnen, sondern müssen zunächst umgewandelt werden. Dieses wurde automatisiert durch ein Programm realisiert, das im Online-Repositorium einsehbar ist. Es zeigt sich hier auch beispielhaft, dass die Umwandlung in die Normform aufwendig sein kann.

Ein Schwerpunkt dieses Programms besteht dabei in der Umwandlung von Altersangaben: Dabei wird im Algorithmus der Sonderfall abgedeckt, dass in den Datumszellen Altersangaben stehen. So kann dort statt dem Geburtsdatum eine Angabe zum Alter gemacht werden. Die hier enthaltenen Altersangaben werden während der Bereinigung im Algorithmus erkannt und zu Datumsangaben verarbeitet. Aus diesem Grunde findet an dieser Stelle keine Separierung in die Normform-Variablen ›birthday‹ und ›ageAtDeath‹ statt. Eine solche Separierung wäre ein alternativ mögliches Vorgehen.

Da Altersangaben nur in Beziehung mit anderen Variablen interpretiert werden können, bezieht die Aufbereitung dieser Daten weitere Informationen eines Records mit ein (z. B. das Alter bei Tod und das Todesdatum zur Berechnung des Geburtszeitpunkts). Für die Aufbereitung ist aufgrund der relativen Beziehung der Variablen untereinander eine Betrachtung sämtlicher Datumsangaben des Records notwendig.

Es wird zunächst geprüft, ob die Zeitangabe einer normierten Schreibweise entspricht. Diese wird hier als D.M.YYYY definiert und darüber ermittelt, ob sich der String in ein datetime-Objekt umwandeln lässt. Wenn das der Fall ist, ist die Zuordnung erfolgreich. In dem Fall, dass das Geburtsdatum nicht der Schreibweise D.M.YYYY entspricht, soll die Art und Weise der Zeitangabe identifiziert werden. Es sind verschiedene Ursachen möglich:

- Das Datum enthält nur eine Jahresangabe.
- Statt einem Datum wird eine Altersangabe in Jahren gemacht.
- Statt einem Datum wird eine Altersangabe in Tagen gemacht.
- Die Zeitangabe enthält nur eine Info darüber, dass das Ereignis überhaupt eingetroffen ist (»ja«). Das deutet bei Todesangaben auf einen frühen Tod des Kindes hin.
- Die Zeitangabe enthält keine Information, die Rückschlüsse auf die zeitliche Einordnung zulässt.

[54] Munke 2019, S. 118. Personen innerhalb der KLF können also doppelt vorkommen, indem sie auf einer Karteikarte in der Rolle des Kindes erscheinen, auf einer anderen als Familienoberhaupt oder Ehefrau. Auch Drittpersonen können in den anderen Rollen vorkommen. Dadurch reduziert sich im Zuge eines Record Linkage die Anzahl der Personeneinträge.
[55] Für eine detaillierte Erklärung des Aufbaus der Karteikarten vgl. Verein für Computergenealogie 2018–2019.
[56] Online durchsuchbar, vgl. Verein für Computergenealogie 2018–2019.
[57] Erwähnenswert ist, dass nicht jedes Feld einen Eintrag enthält, sondern vieles optional ist. Dadurch stehen im Zweifel bei jedem Eintrag andere Daten zur Verfügung.

Bei den ersten vier der fünf Fälle kann eine Zeitangabe abgeleitet werden. Im fünften Fall besteht die Herausforderung darin, zu erkennen, dass es sich nicht um eine Angabe mit zeitlichem Bezug handelt. Zunächst werden solche Angaben erkannt, die nur aus einer Jahreszahl bestehen. Hier wird zunächst geprüft, ob das zu prüfende Datum in einen Integerwert umgewandelt werden kann. Das ist der Fall, wenn es sich um eine reine Jahresangabe handelt. Ist nicht nur eine Jahreszahl Inhalt der Angabe, so wird geprüft, ob sie ein »J.« (für Jahr), ein »T.« (für Tag) oder ein »ja« (für das generelle Eintreten des Ereignisses) enthält. Je nach Typ des Datums werden darauffolgend unterschiedliche Berechnungsschritte durchgeführt. Beispielsweise wird bei einer Angabe »64 J.« in einem Feld zum Sterbedatum versucht, das Sterbedatum anhand der Geburts- oder Taufangabe zu ermitteln. Diese Verarbeitungsschritte haben zur Folge, dass am Ende ein Gedbas4all-konformes Datumsformat vorliegt.

Die grundsätzliche Zuordnung der KLF zu den Datenfeldern der Normform wird wie in Tabelle 3 ersichtlich realisiert. Dabei werden die Datumsangaben wie zuvor beschrieben behandelt.

Variable KLF	**Variable der Normform**
page [ID der Karteikarte]	source
lastname	lastnameGiven
firstname	firstnameGiven
Beruf	occupation
Rolle	---
Ort	---
GOV-Id	---
Bezugsperson	---
Art der Beziehung	---
Geburtsdatum/Alter	birthday
Taufdatum	baptismday
Heiratsdatum	marriageday1
Sterbedatum	deathday
Beerd.Datum	burialday
Bemerkung	---
weiterer Ort	---

Tab. 3: Direkte Umwandlung der KLF-Struktur in die Normform. [Goldberg / Mernitz 2023]

Die KLF-Variablen Rolle, Bezugsperson, Art der Beziehung und ID werden zudem herangezogen, um weitere Variablen der Normform zu füllen (vgl. Tabelle 4).

Variable der Normform	**Verknüpfung der KLF-Variablen**
idSpouse1, idSpouse2, idSpouse3	Ein Familienoberhaupt erhält die ID der Ehefrau auf derselben Karteikarte. Eine Ehefrau erhält die ID des Familienoberhauptes auf derselben Karteikarte. Eine Drittperson vom Typ Ehefrau / Ehemann führt dazu, dass bei der Drittperson wie auch bei der Bezugsperson eine ID für den*die Ehepartner*in ergänzt wird.
idFather, idMother	Bei Kindern werden die IDs der Eltern jeweils ergänzt. Tritt eine Drittperson als Vater auf, so wird diese bei dem Kind ergänzt.
idGlobal	Wird ohne Bezug zur KLF fortlaufend vergeben.

Tab. 4: Indirekte Umwandlung der KLF-Struktur in die Normform. [Goldberg / Mernitz 2023]

4.1.2 Kartei Leipziger Kreisamtstestamente (1696–1829)

Für das Amt Leipzig liegen für die Zeit von 1696 bis 1829 Testamente innerhalb von 120 Bänden im Sächsischen Staatsarchiv vor.[58] Zum Auffinden von Testamentsvorgängen existiert eine Kartei – die KLK. Auch die KLK ist im Rahmen eines Datenerfassungsprojektes des Vereins für Computergenealogie mit Hilfe des DES erfasst worden und online einsehbar.[59] Sie umfasst 4.800 Karteikarten, auf denen jeweils zu einer Person die entsprechenden Vorgänge zum Testament erfasst sind. Ehepartner*innen erhalten jeweils eigene Karten. Jedoch können auch Drittpersonen auf den Karten erscheinen. Dazu gibt es in der KLK die Variable ›Rolle‹, in der zwischen Erblasser*innen und Drittpersonen / Verwandten unterschieden wird. Dies führt dazu, dass ca. 6.500 Personendatensätze entstehen. Zu den Erblasser*innen sind jeweils entsprechende Informationen über die Testierung vorhanden. Bei einer Drittperson dahingegen ist die Art der Beziehung zur testierenden Person dokumentiert.

Auch die Variablen der KLK-Erfassung lassen sich in die Normform umwandeln. Wie bei der KLF gibt es dabei Variablen, die sich direkt auf die Normform übertragen lassen (vgl. Tabelle 5) oder auch indirekt hergeleitet werden können (vgl. Tabelle 6).

Variable KLK	**Variable der Normform**
page	---
firstname	firstnameGiven
Stand/Beruf	occupation
Rolle	---
Ort	---
Band und Blatt	source
Familienstand	---
Ereignis 1, ..., Ereignis 8	---
Geschlecht	sex
Bezugsperson ID	---
Bezugsperson Name	---
Art der Beziehung	---
Sterbedatum	deathday
Datum von [erster Vorgang]	---
Datum bis [letzter Vorgang]	---
idGlobal	›A‹ + id, bzw. neue ID bei zusammengeführten Personen.

Tab. 5: Direkte Umwandlung der KLK-Struktur in die Normform. [Goldberg / Mernitz 2023]

Die indirekte Herleitung betrifft vor allem die Nachnamen. In der KLK sind nämlich die vorherigen Nachnamen mit abgebildet. Wenn der Teilstring »geb.« im Nachnamen vorhanden ist, dann ist der Name danach der Geburtsname, der Name davor ist ein Heiratsname. Bei dem Teilstring »verw.« dahingegen ist der folgende Name der Ehename einer früheren Verbindung, der davorstehende der aktuelle Ehename. Wird im Nachnamen dahingegen der Begriff »verehel.« verwendet, ist der erste Teil der Geburtsname, der letztere der Heiratsname. Sind bei einer Frau keine Hinweise enthalten, von wem der Nachname stammt, wird dieser der Variable ›surnameUnknown‹ zugeordnet. Bei Männern wird angenommen, dass der angegebene Nachname immer der Geburtsname ist.

Auch bei den IDs findet eine indirekte Zuordnung statt. Wenn eine Drittperson definiert ist und diese den Typ ›Ehemann‹ oder ›Ehefrau‹ aufweist, dann wird die ID des Ehepartners / der Ehepartnerin hinzugefügt. Gleiches erfolgt bei Müttern und Vätern, Söhnen und Töchtern bei den Variablen ›idFather‹ und ›idMother‹. Bei der eigenen ID einer Person wird die ID der KLK grundsätzlich übernommen. Ihr wird ein ›A‹ vorangestellt, um die IDs eindeutig von den IDs der KLF zu unterscheiden. Die ID wird jedoch überschrieben, wenn Dubletten in der KLK vorhanden sind. Das kommt vor, wenn Ehepartner*innen jeweils eigene

[58] Sächsisches Staatsarchiv. Bestand 20009 Amt Leipzig.
[59] Verein für Computergenealogie 2019–2021.

Karteikarten haben. Schlüssel zur Erkennung von Dubletten ist hierbei die Quellenangabe (Band und Blatt) der Testamente. Wenn nur die ID eines Ehepartners / einer Ehepartnerin verändert wird, deutet es darauf hin, dass in einem Eintrag der*die Ehepartner*in der Verweis auf den*die andere*n Ehepartner*in als Drittperson fehlt.

Des Weiteren wird angenommen, dass die Testamentseröffnung kurz nach dem Tod vorgenommen wird. Liegt also kein Todestag vor, so wird das Jahr der Testamentsöffnung auch als Todesjahr verwendet. Die Umwandlung in die Normform wurde automatisiert durch ein Programm realisiert, das im Online-Repositorium einsehbar ist.

Variable der Normform	**Verknüpfung der KLF-Variablen**
idSpouse1, idSpouse2, idSpouse3	Wenn eine Drittperson (›Rolle‹ = = Drittperson / Verwandter) vom Typ Ehefrau oder Ehemann vorhanden ist (›Art der Beziehung‹), dann wird ihre ID (›Bezugsperson ID‹) entsprechend ergänzt.
idFather, idMother	Wenn eine Drittperson vom Typ Vater / Mutter / Sohn / Tochter vorhanden ist, dann wird die ID entsprechend ergänzt.
idGlobal	id
lastname	surnameGiven, surnameUnkown, surnameMarriage1, surnameMarriage2, surnameMarriage3
deathday	Eröffnung

Tab. 6: Indirekte Umwandlung der KLK-Struktur in die Normform. [Goldberg / Mernitz 2023]

4.2 Resultate des Record Linkage

Da sowohl in der KLK und KLF Personen mehrfach genannt werden können, ist zunächst ein Vergleich der beiden normformatierten Datentabellen mit sich selbst sinnvoll. Erst darauffolgend werden die Ergebnisse miteinander verglichen und zusammengeführt. Dabei stellt sich die Frage nach der Qualität der Zusammenführung. Zur Validierung der Resultate bietet sich eine Identifizierung von falschpositiven und falschnegativen Ergebnissen an. Eine solche Identifizierung ist an dieser Stelle nur begrenzt möglich, da auch mit einer manuellen Überprüfung nicht zweifelsfrei festgestellt werden kann, ob eine Verknüpfung nun richtig oder falsch ist. Diese Einschätzung nämlich basiert vielmehr auf den Heuristiken, die zuvor definiert, formalisiert und auch umgesetzt worden sind.

Dennoch wird eine manuelle Überprüfung der zusammengeführten Records vorgenommen. Da nicht alle Records überprüft werden können, werden nur die Personen behandelt, deren Geburtsname mit ›A‹ beginnt.[60] Von diesen 4.251 Records werden 651 zusammengeführt (15,3 Prozent). Dabei konnten einige falschpositive Ergebnisse identifiziert werden: 1585 und 1586 sind zwei Elisabeth Albrechts in Leipzig getauft worden (IDs 14505990 und 14506456). Hier liegt das Taufdatum weniger als ein Jahr auseinander. Da zu beiden die Angabe des Vaters vorliegt, hätte über den Vergleich der Väter erkannt werden können, dass es sich nicht um dieselbe Person handelt. Hier ist Potenzial für eine Erweiterung des Algorithmus. Gleiches trifft auf Maria Arnoldt (14558811 und 14558853), Maria Albrecht (14499274 und 14505976), Barbara Abitzsch (14457480 und 14458315), Thomas Abitzsch (14457495 und 14458366), Maria Arnst (14556375 und 14556424) und Paul Arnst (14556496 und 14560610). Bei dem / den Bäcker(n) Anton Arnoldt (14554173 und 14554184) wird es sich möglicherweise um unterschiedliche Personen handeln. Helga Moritz hat diese beiden auch nicht auf derselben Karteikarte erfasst; die Heiratsdaten liegen 28 Jahre auseinander. Möglicherweise ist die Implementierung einer maximalen Distanz von Heiratsdaten notwendig, wenngleich diese dann jedoch nicht bei 28 Jahren, sondern deutlich höher liegen sollte. Andere Beispiele für weit auseinander liegende Heiratsdaten stellen Joachim Arnst (14556335 und 14560573) oder zwei weitere Personen namens Thomas Abitzsch (14457397 und 14458332) dar. Wird angenommen, dass es sich bei diesen elf Fällen tatsächlich um falschpositive Ergebnisse handelt, liegt die Rate an Falschpositiven bei 1,7 Prozent.

Weiterhin ist auffällig, dass bei vielen Personen ein positiver Prioritätswert aufgrund gleicher Heiratsdaten oder gleicher Berufsangaben zustande kommt. Gleiche Berufsangaben sind in solchen Orten problematisch, in denen es viele namensgleiche Personen gibt und bestimmte Berufe aufgrund der nichtdiversifizierten Wirtschaftsstruktur dominant sind. In diesen Fällen

[60] Hierdurch werden nicht alle Aspekte des Algorithmus in gleicher Weise geprüft. Insbesondere die intergenerationalen Elemente der Plausibilitätsprüfung entfallen, da insbesondere Mütter Geburtsnamen mit anderen Anfangsbuchstaben haben.

scheint eine Anwendung des Algorithmus nur sinnvoll, wenn weitere Lebensdaten vorhanden sind. In Leipzig gibt es bis auf wenige Ausnahmen im von den Daten abgedeckten Zeitraum eine große Diversität an Namen und Berufen, sodass dieser Umstand hier kein Problem darstellt.

Die Relevanz von Berufsangaben für den Prioritätswert führt auch dazu, dass etwas mehr Männer (58,7 Prozent) als Frauen zusammengeführt werden. Um mehr Frauen zusammenzuführen, kann es eine Option sein, über die Übereinstimmung einer seltenen Kombination aus Vornamen einen positiven Prioritätswert zu erreichen: Die Übereinstimmung von zwei Personen namens »Maria« ist weniger wahrscheinlich als die von zwei Personen namens »Johanna Maria Henriette Friederike«, die von »Johann« anders als die von »Immanuel Friedlieb«. Auch die Seltenheit der Namen kann hier integriert werden. Ebenso kann die Übereinstimmung seltener Berufe priorisiert werden.

Bemerkenswert ist auch, dass Vor- und Nachname bei den zusammengeführten Personen in 90,6 Prozent der Fälle exakt übereinstimmen. Das liegt auch darin begründet, dass die Erstellerin der KLF die Namensschreibweise normiert hat. Für eine Bewertung der Ähnlichkeitsanalyse der Namensstrings sind die Daten darum nicht besonders gut geeignet. Es kann zudem sinnvoll sein, eine Synonymerkennung der Namen zu implementieren (»Hans« und »Johann«, »Xine« als schriftliche Abkürzung für »Christine« etc.).

Zudem ist zu vermuten, dass es im gesamten Datensatz eine nicht näher bekannte Anzahl von falschnegativen Zuordnungen gibt – also Records, die zusammengeführt werden müssten, es aber nicht wurden. Für diesen Abgleich wäre eine genealogische Übersicht der Leipziger Familien als Goldstandard notwendig, die jedoch nicht existiert. Darum kann dieser Abgleich nicht vorgenommen werden. Auffällig bei der manuellen Überprüfung ist, dass es einige wenige Fälle gibt, in denen eine Person sogar vier Mal im Datensatz auftaucht (und dann zweimal zusammengeführt wird). Um die Anzahl an Falschnegativen zu verringern, kann eine mehrfache Iteration also hilfreich sein.

Dass mit dem hier vorgestellten Algorithmus jedoch ein erheblicher Teil der tatsächlich zusammenzuführenden Records auch zusammengeführt wird, zeigt ein Vergleich mit der Personenzusammenführung des Genealogie-Programms *Ahnenblatt* 2.99[61]: Wird die GEDCOM-Datei dort hineingeladen und werden die Vorschläge zur Zusammenführung der Personen ohne weiteren manuellen Eingriff ausgeführt, werden 25.329 von 241.466 Personen zusammengeführt.[62] Das entspricht mit 10,5 Prozent einem deutlich geringeren Anteil als im Test der mit »A« beginnenden Personen mit dem hier entwickelten Algorithmus (15,3 Prozent). Über alle Daten ist mit dem Algorithmus eine Erkennung von 13,2 Prozent zu erkennen (vgl. Tabelle 7). Bei der KLK werden mit 0,7 Prozent erwartungsgemäß wenige Personen verknüpft, da die Normform hier bereits wenige Duplikate enthält. Von den Testamentsdatensätzen konnten mit dem Algorithmus 413 Einträge einer Person zugeordnet werden, auf 5.348 Personen traf das nicht zu.

	KLF	**KLK**
KLF	31.791 von 241.465 Records zusammengeführt (Anteil: 13,2 Prozent)	---
KLK	413 zusammengeführt bei 5.761 Personen (Anteil: 7,2 Prozent)[63]	41 zusammengeführt bei 5.802 Personen (Anteil: 0,7 Prozent)[64]

Tab. 7: Übersicht über die Anzahl der verknüpften Personen aus den Normformen. [Goldberg / Mernitz 2023]

Insgesamt sind die Ergebnisse des Algorithmus also gut: Ein nicht näher zu quantifizierender, aber erheblicher Teil der tatsächlich zusammenzuführenden Records konnte auch zusammengeführt werden. Etwa 98 Prozent dieser zusammengeführten Records sind korrekt. Überall dort, wo Personen klar zusammengeführt werden können, wird dieses gemacht. Das spart besonders bei großen Datensätzen viele Ressourcen. Zugleich ist die Lösung nicht perfekt, vielmehr ist sie ein erster Ansatz, auf den aufzubauen es sich lohnt. Besonders die Formalisierung und Automatisierung genealogischer Heuristiken kann erweitert und das Record Linkage somit verbessert werden.[65]

[61] Vgl. Böttcher 2018.
[62] Die Zusammenführung basiert hierbei auf gleichen Namen und einem gleichen Ereignisdatum (z. B. das Taufdatum) und betrifft auch die nähere Verwandtschaft der betreffenden Personen wie die Eltern, Kinder oder Geschwister. Vgl. Böttcher 2018, S. 17.
[63] Hier werden die Daten genutzt, nachdem die KLF und KLK jeweils mit sich selbst abgeglichen worden sind. Von den 5.761 übrig gebliebenen Personen in der KLK konnten 413 in der KLF gefunden werden.
[64] Die KLK enthält zwar 6.524 Personendatensätze. Die Überführung in die Normform sorgt jedoch dafür, dass bereits Personen zusammengeführt werden, sodass hier 5.802 Personendatensätze übrig bleiben.
[65] Es gibt weitere, noch nicht in die Normform integrierte Informationen, die eine hohe praktische Relevanz für genealogische Verknüpfungen haben, deren maschinelle Interpretation aber sehr schwer erscheint. Dazu gehören insbesondere Angaben zu den Taufpaten.

5. Zusammenfassung

Gleiches mit Gleichem zu verbinden – darin besteht eine Herausforderung im Umgang mit historischen Personendaten. Der vorgestellte Ansatz leistet einen Beitrag, diese Herausforderung in der praktischen Forschung zu bewältigen. Im Unterschied zu vorhergehenden Studien nutzt der vorgestellte Algorithmus dafür eine Vielzahl von genealogisch relevanten Informationen eines Records, vom Beerdigungsdatum über den Beruf bis hin zu den Lebensdaten der Eltern. Die Besonderheit hier ist, dass verschiedene Variablen in Beziehung zueinander gesetzt werden. So werden zahlreiche genealogische Regeln genutzt, um zu erkennen, dass Records disjunkt sind. Die letztendliche Übereinstimmung (Similarität der Records) wird dahingegen über die Jaro-Winkler-Distanz und die Kölner Phonetik ermittelt und ist aufgrund des letzteren Aspekts vor allem an den deutschen Sprachraum angepasst. Auch die implementierten genealogischen Heuristiken sind an den deutschen historischen Sprach- und Kulturraum und die evangelische bzw. römisch-katholische Religionspraxis angepasst; so kennen diese beispielsweise keine Erwachsenentaufen oder Ehen mit mehreren Personen. Eine vergleichbare Lösung in diesem Umfang zur Automatisierung genealogischer Heuristiken existiert bisher nicht. Die Umsetzung in der Programmiersprache Python bietet die Möglichkeit der Veränderung und Anpassung an die jeweiligen Herausforderungen.

Hierbei zeigt sich sowohl ein großer Vorteil als auch ein großer Nachteil der vorgestellten Lösung: Der Vorteil besteht darin, dass der Algorithmus besonders gut ist, wenn viele Informationen (vor allem Datumsangaben) zu einer Person bekannt sind. Somit ist die Lösung sehr gut geeignet für Quellen mit vielen genealogisch relevanten Daten. Das ist beispielsweise bei dem zur Validierung genutzten Beispiel Leipziger Quellen der Fall. Hilfreich ist sie vor allem bei der Bearbeitung großer Datenbestände, die manuell nicht mehr mit vertretbarem Aufwand zu verarbeiten sind. Neben dem Einsatz in der Wissenschaft oder in Time-Machine-Projekten ist es dadurch vorstellbar, Daten aus Kirchenbüchern mit dem Algorithmus zu verknüpfen. Durch den Algorithmus ist nämlich die automatisierte genealogische Verknüpfung über mehr als zwei Generationen hinweg möglich. Der Algorithmus kann hier beispielsweise bei der Erstellung von Ortsfamilienbüchern ein nützliches Werkzeug sein. Hierzu gilt es in einem nächsten Schritt, die Nachnutzung des Programmcodes niederschwelliger möglich zu machen, beispielsweise durch ein Webinterface. Ziel ist es, dass zwei Normform-Tabellen als CSV-Dateien in einem Webbrowser hochgeladen werden können. Hier würde zudem die Möglichkeit bestehen, diverse Funktionen des Algorithmus ab- oder anzuschalten oder Grenzwerte zu variieren.

Nachteilig ist der Algorithmus dahingegen, wenn nur wenige Informationen über die durch die Records beschriebenen Personen vorhanden sind. Sind beispielsweise nur Namen vorhanden, ist es sicherlich angebrachter, verschiedene String-Matching-Algorithmen an den jeweiligen Daten zu testen. Allerdings kann das erstellte Programm auch beliebig verändert, erweitert und an die eigenen Bedürfnisse angepasst werden. Dass das Programm für verschiedene Zwecke angepasst werden muss, liegt aufgrund der Validierung mittels der Leipziger Daten nahe. Insbesondere die Herstellung der normalisierten Form (Normform) bedarf einer solchen Aufmerksamkeit. Es ist zudem eine Illusion zu glauben, dass es zurzeit eine Lösung geben kann, in der zwei völlig verschiedene Quellen ohne große Vorarbeit einem automatisierten Record Linkage zugeführt werden können. Nichtsdestotrotz stellt das entwickelte Programm ein geeignetes Grundgerüst für die Anpassung dar. Weiteres Potenzial besteht in der Evaluation und Integration von Methoden maschinellen Lernens, die hier, wie eingangs erläutert, bewusst nicht genutzt worden sind.

Bibliografische Angaben

Ran Abramitzky / Leah Boustan / Katherine Eriksson / James Feigenbaum / Santiago Pérez: Automated Linking of Historical Data. In: Journal of Economic Literature 59 (2021), H. 3, S. 865–918. DOI: 10.1257/jel.20201599 [Nachweis im GVK]

Ran Abramitzky / Roy Mill / Santiago Pérez: Linking individuals across historical sources: A fully automated approach. In: Historical Methods: A Journal of Quantitative and Interdisciplinary History 53 (2020), H. 2, S. 94–111. DOI: 10.1080/01615440.2018.1543034 [Nachweis im GVK]

Jürgen Bähr / Christoph Jentsch / Wolfgang Kuls: Bevölkerungsgeographie. Berlin u. a. 1992. (= Lehrbuch der allgemeinen Geographie, 9). [Nachweis im GVK]

Rohan Baxter / Peter Christen / Tim Churches: A Comparison of Fast Blocking Methods for Record Linkage. 2003. PDF. [online]

Dirk Böttcher: Ahnenblatt Handbuch. 2018. PDF. [online]

Peter Christen / Dinusha Vatsalan / Zhichun Fu: Advanced Record Linkage Methods and Privacy Aspects for Population Reconstruction. A Survey and Case Studies. In: Population Reconstruction. Hg. von Gerrit Bloothooft / Peter Christen / Kees Mandemakers / Marijn Schraagen. Cham u. a. 2015, S. 87–110. DOI: 10.1007/978-3-319-19884-2_5 [Nachweis im GVK]

The Church of Jesus Christ of Latter-day Saints: The GEDCOM Standard. Salt Lake City 2019. Release 5.5.1. vom 15.11.2019. PDF. [online]

Antonin Delpeuch / Adrian Pohl / Fabian Steeg / Thad Guidry Sr. / Osma Suominen: Reconciliation Service API v0.2. A Protocol for Data Matching on the Web. Final Community Group Report. 10.04.2023. HTML. [online]

Julia Efremova / Bijan Ranjbar-Sahraei / Hossein Rahmani / Frans A. Oliehoek / Toon Calders / Karl Tuyls / Gerhard Weiss: Multi-Source Entity Resolution for Genealogical Data. In: Population Reconstruction. Hg. von Gerrit Bloothooft / Peter Christen / Kees Mandemakers / Marijn Schraagen. Cham u. a. 2015, S. 129–154. DOI: 10.1007/978-3-319-19884-2_7 [Nachweis im GVK]

Jerome Fan / Suneel Upadhye / Andrew Worster: Understanding receiver operating characteristic (ROC) curves. In: Canadian Journal of Emergency Medicine 8 (2006), H. 1, S. 19–20. DOI: 10.1017/S1481803500013336 [Nachweis im GVK]

James J. Feigenbaum: Automated census record linking: a machine learning approach. 2016. Handle: 2144/27526

Eli Fure: Interactive Record Linkage: The Cumulative Construction of Life Courses. In: Demographic Research 3 (2000). 12.12.2000. DOI: 10.4054/DemRes.2000.3.11

Corry Gellatly: Reconstructing Historical Populations from Genealogical Data Files. In: Population Reconstruction. Hg. von Gerrit Bloothooft / Peter Christen / Kees Mandemakers / Marijn Schraagen. Cham u. a. 2015, S. 111–128. DOI: 10.1007/978-3-319-19884-2_6 [Nachweis im GVK]

Kleanthi Georgala / Benjamin van der Burgh / Marvin Meeng / Arno Knobbe: Record Linkage in Medieval and Early Modern Text. In: Population Reconstruction. Hg. von Gerrit Bloothooft / Peter Christen / Kees Mandemakers / Marijn Schraagen. Cham u. a. 2015, S. 173–195. DOI: 10.1007/978-3-319-19884-2_9 [Nachweis im GVK]

Jan Michael Goldberg: Kontextsensitive Entscheidungsfindung zur automatisierten Identifizierung und Clusterung deutschsprachiger Urbanonyme. In: Zeitschrift für digitale Geisteswissenschaften 7 (2022). 10.10.2022. DOI: 10.17175/2022_005

Jan Michael Goldberg / Katrin Moeller: Automatisierte Identifikation und Lemmatisierung historischer Berufsbezeichnungen in deutschsprachigen Datenbeständen. In: Zeitschrift für digitale Geisteswissenschaften 7 (2022). 08.03.2022. DOI: 10.17175/2022_002

Lifang Gu / Rohan Baxter / Deanne Vickers / Chris Rainsford: Record Linkage: Current Practice and Future Directions. In: CMIS Technical Report 03/83 (2003). PDF. [online]

J. Tuomas Harviainen / Bo-Christer Björk: Genealogy, GEDCOM, and popularity implications. In: Informaatiotutkimus 37 (2018), H. 3, S. 4–14. DOI: 10.23978/inf.76066 [Nachweis im GVK]

Saskia Hin / Dalia A. Conde / Adam Lenart: New light on Roman census papyri through semi-automated record linkage. In: Historical Methods: A Journal of Quantitative and Interdisciplinary History 49 (2016), H. 1, S. 50–65. DOI: 10.1080/01615440.2015.1071226 [Nachweis im GVK]

Frédéric Kaplan: The Venice Time Machine. In: DocEng '15: Proceedings of the 2015 ACM Symposium on Document Engineering (DocEng, Lausanne, 08.–11.09.2015). New York 2015, S. 73. DOI: 10.1145/2682571.2797071

Jürgen Kocka / Karl Ditt / Josef Mooser / Heinz Reif / Reinhard Schüren: Familie und soziale Platzierung. Studien zum Verhältnis von Familie, sozialer Mobilität und Heiratsverhalten an westfälischen Beispielen im späten 18. und 19. Jahrhundert. Wiesbaden 1980 (= Forschungsberichte des Landes Nordrhein-Westfalen, 2953). DOI: 10.1007/978-3-322-87746-8

Catherine G. Massey: Playing with matches: An assessment of accuracy in linked historical data. In: Historical Methods: A Journal of Quantitative and Interdisciplinary History 50 (2017), H. 3, S. 129–143. DOI: 10.1080/01615440.2017.1288598 [Nachweis im GVK]

Martin Munke: Citizen Science / Bürgerwissenschaft. Projekte, Probleme, Perspektiven am Beispiel Sachsen. In: Forschungsdesign 4.0. Datengenerierung und Wissenstransfer in interdisziplinärer Perspektive. Hg. von Jens Klingner / Merve Lühr (Dresden, 19.–21.04.2018). Dresden 2019, S. 107–124. DOI: 10.25366/2019.11

Charini Nanayakkara / Peter Christen / Thilina Ranbaduge: Temporal graph-based clustering for historical record linkage. In: Proceedings of 14th International Workshop on Mining and Learning with Graphs (MLG 14, London, 20.08.2018). New York 2018. PDF. [online]

Hans Joachim Postel: Die Kölner Phonetik. Ein Verfahren zur Identifizierung von Personennamen auf der Grundlage der Gestaltanalyse. In: IBM-Nachrichten 19 (1969), S. 925–931. [Nachweis im GVK]

Günther Schönfelder / Michael Börngen: Naturräumliche Grundlagen. Landschaft und Klima. In: Geschichte der Stadt Leipzig. Hg. von Uwe John / Enno Bünz. 4 Bde. Leipzig 2015–2019. Bd. 1 (2015): Von den Anfängen bis zur Reformation, S. 33–47. [Nachweis im GVK]

Peter Schulz: GEDTOOL. Makrosammlung für GEDCOM-Dateien. V. 2.7 vom 14.09.2017. PDF. [online]

Gunnar Thorvaldsen / Andersen Trygve / Hilde L. Sommerseth: Record Linkage in the Historical Population Register for Norway. In: Population Reconstruction. Hg. von Gerrit Bloothooft / Peter Christen / Kees Mandemakers / Marijn Schraagen. Cham u. a. 2015, S. 155–171. DOI: 10.1007/978-3-319-19884-2_8 [Nachweis im GVK]

Time Machine Organisation: Local Time Machines. 2022. HTML. [online]

Verein für Computergenealogie (2016a): Gedbas4all / Datenmodell. In: GenWiki. Das Genealogie-Wiki. 2016. HTML. [online]

Verein für Computergenealogie (2016b): Gedbas4all / Datumsangaben. In: GenWiki. Das Genealogie-Wiki. 2016. HTML. [online]

Verein für Computergenealogie: Kartei Leipziger Familien. In: GenWiki. Das Genealogie-Wiki. 2018–2019. HTML. [online]

Verein für Computergenealogie: Kartei Leipziger Kreisamtstestamente. 2019–2021. HTML. [online]

Verein für Computergenealogie: The Historic Gazetteer. 2021. HTML. [online]

Abbildungs- und Tabellenverzeichnis