Technische Hochschule Wildau
Technical University of Applied Sciences

Masterarbeit

zur Erlangung des akademischen Grades
Master

Technische Hochschule Wildau

Fachbereich Wirtschaft, Informatik, Recht

Studiengang Bibliotheksinformatik (M. Sc.)

Thema (deutsch): Die Deduplizierung von bibliothekarischen Metadaten am Beispiel der Datenintegration eines Institutskatalogs in den Bibliotheksverbund IDS St.Gallen

Thema (englisch): Deduplication of library metadata using the example of data integration of an institute catalogue into the library network IDS St.Gallen

Autor/in: Kathrin Verena Heim

Seminargruppe: BIM/16

Betreuer/in: Dipl.-Mathematiker Stefan Lohrum

Zweitgutachter/in: M.A. Petra Keidel

Eingereicht am: 23.09.2019

Die Deduplizierung von bibliothekarischen Metadaten

am Beispiel der Datenintegration eines Institutskatalogs in den Bibliotheksverbund IDS St.Gallen

Masterarbeit

vorgelegt von

Kathrin Heim

Technische Hochschule Wildau

2019

Betreuer: Stefan Lohrum

Gutachterin: Petra Keidel

Zusammenfassung

Dubletten gehören in Bibliotheken zum Alltag. Da Dubletten beim Retrieval und bei der Datenbankeffizienz grosse Probleme verursachen, wird viel Aufwand für deren Vermeidung betrieben.

Thema dieser Masterarbeit ist die Deduplizierung von bibliothekarischen Metadaten. Ziel ist es, im Rahmen einer Datenintegration ein eigenes Deduplizierungsverfahren nach Vorbild von bestehenden Verfahren zu entwickeln und parametrisieren.

Ausgangssituation ist die Integration eines Institutskatalogs in den Bibliotheksverbund IDS St. Gallen. Bei der Datenanalyse zeigt sich, dass die Institutsdaten sehr heterogen sind und die Datenqualität stark variiert. Daher sollen die Daten, wo immer möglich, durch qualitativ bessere Metadaten ersetzt werden.

Zunächst wird ein Kriterienkatalog für das eigene Verfahren aufgestellt. Danach werden bestehende Deduplizierungsverfahren untersucht und auf ihre Eignung für die vorliegende Situation geprüft. Aufgrund dieser Bewertung wird ein eigenes Deduplizierungsverfahren entwickelt.

Die Analyse der zu integrierenden Daten, das Schema Mapping sowie die Datenbereinigung spielen eine wichtige Rolle bei der erfolgreichen Deduplizierung der Institutsdaten. Die vorgenommenen Bereinigungen werden gezeigt und die Unterschiede in den Ergebnissen – im Vergleich mit unbereinigten Daten - präsentiert.

Die technische Umsetzung des eigenen Deduplizierungsverfahren wird dokumentiert, die Besonderheiten und die Parametrisierung des Verfahrens erläutert. Im vorliegenden Fall werden die Daten durch Abfragen in grossen Datenpools wie swissbib oder GVI dedupliziert und dabei gleichzeitig die Datenqualität verbessert.

Die vorgenommenen Tests und Ergebnisse dieses Verfahrens werden präsentiert und kommentiert. Die Ergebnisse zur Effektivität und Effizienz des Verfahrens sind zufriedenstellend und können umgesetzt werden.

Abstract

Duplicates are part of everyday life in libraries. Since duplicates cause major problems with retrieval and database efficiency, a lot of effort is put into avoiding them.

The topic of this master thesis is the deduplication of library metadata. The aim is to develop and parameterize a dedicated deduplication procedure based on existing procedures within the framework of data integration.

The initial situation is the integration of an institute catalogue into the library network IDS St. Gallen. The data analysis shows that the institute data are very heterogeneous and the data quality varies greatly. Wherever possible, the original data should therefore be replaced by better-quality metadata.

First, a catalogue of criteria is elaborated for the procedure. Existing deduplication procedures are then examined and their suitability for the present situation tested. Based on this evaluation, a dedicated deduplication procedure is developed.

The analysis of the data to be integrated, the schema mapping and the data cleansing play an important role in the successful deduplication of the institute's data. The adjustments made are shown and the differences in the results - compared to the unadjusted data - are presented.

The technical implementation of the own deduplication procedure is documented, the special features and the parameterization of the procedure are explained. In the present case, the data is deduplicated by queries in a large data pools such as swissbib or GVI, while improving the data quality at the same time.

The tests carried out and the results of this procedure are presented and commented on. The results on the effectiveness and efficiency of the procedure are satisfactory and can be implemented.

Inhaltsverzeichnis

Zusammenfassung **1**

Abstract **2**

Inhaltsverzeichnis **3**

Abbildungsverzeichnis **7**

Tabellenverzeichnis **8**

Abkürzungsverzeichnis **9**

1 Einleitung **10**

1.1 Ausgangslage und Motivation 10

1.2 Ziel der Arbeit 11

1.3 Abgrenzung des Themas 11

1.4 Aufbau der Arbeit 11

2 Datenintegration und Deduplizierung **13**

2.1 Heterogene Daten 14

2.1.1 Technische Heterogenität 15

2.1.2 Strukturelle Heterogenität 16

2.1.3 Semantische Heterogenität 16

2.2 Datenbereinigung 16

2.3 Schema Mapping 17

2.4 Dublettenerkennung 18

2.4.1 Definition von Dubletten 18

2.4.2 Entstehung und Konsequenzen von Dubletten 19

2.4.3 Ähnlichkeitsmass von Attributwerten 19

2.4.4 Gewichtung von Attributen 20

2.4.5 Sorted Neighbourhood 21

2.5 Datenfusion 22

3 Deduplizierungsverfahren **23**

3.1 Entwicklung eines Deduplizierungsverfahrens 23

3.1.1 Designziel 24

3.1.2 Anwendungsschritte 25

3.1.3 Auswahl der Felder ... 25
3.1.4 Sortierschlüssel (Matchkey) ... 26
3.1.5 Evaluation ... 27
3.1.6 Merging / Datenfusion ... 27
3.1.7 Qualitätsmessung: Effektivität und Effizienz ... 28
3.1.7.1 Effektivität ... 28
3.1.7.2 Effizienz ... 30
3.2 Kriterienkatalog für die Integration des IFF-Bestands ... 30
3.2.1 Designziel für das IFF-Verfahren ... 30
3.2.1.1 Ebene: Titelebene ... 30
3.2.1.2 Dokumenttyp: Monografien und analytische Aufnahmen ... 31
3.2.2 Anwendungsschritte: 2-Schritt-Verfahren ... 31
3.2.3 Felder: Attribute mit sinnvollen Informationen ... 31
3.2.4 Sortierschlüssel: keiner ... 32
3.2.5 Evaluation: gewichtetes Verfahren ... 32
3.2.6 Merging/Datenfusion: Offline-Verfahren ... 32
3.2.7 Qualitätsmessung für das IFF-Verfahren ... 33
3.2.7.1 Effektivität: Gutes Resultat ... 33
3.2.7.2 Effizienz: vertretbarer Aufwand ... 33
3.3 Existierende Deduplizierungsverfahren ... 34
3.3.1 Gemeinsame Eigenschaften ... 36
3.3.2 Hickey und Rypka / OCLC (1979) ... 38
3.3.2.1 DDR (1990) ... 44
3.3.3 OCLC-Verfahren auf Werkebene (sowie Spin-offs) ... 44
3.3.3.1 FRBR Work-Set (2009) ... 44
3.3.3.2 GLIMIR (2012) ... 45
3.3.3.3 Magnus Pfeffer (2012) ... 46
3.3.4 Culturegraph (2013) ... 47
3.3.5 KOBV (1999) ... 49
3.3.5.1 GVI (2019) ... 51
3.3.6 swissbib (2008) ... 53
3.3.6.1 SLSP (2019) ... 55
3.3.7 Alma und Primo / Ex Libris (2015) ... 58
3.3.7.1 Melvyl / CDL (1992/2006) ... 60
3.3.8 IDSSG (2009) ... 60
3.4 Bewertung der Verfahren ... 64

3.4.1 Allgemeine Bewertung der Verfahren ... 64
3.4.2 Effektivität der Algorithmen (Recall / Precision) ... 65
3.4.3 Effizienz der Algorithmen ... 66
3.4.4 Bewertung in Bezug auf Kriterienkatalog ... 67

4 Datenanalyse und Datenbereinigung der IFF-Daten ... 68

4.1 Datenanalyse ... 68
4.1.1 Datentypen ... 70
4.1.1.1 Monografien ... 71
4.1.1.2 Mehrteilige Monografien ... 71
4.1.1.3 Stücktitel in monografischer Reihe ... 72
4.1.1.4 Analytische Aufnahmen ... 73
4.1.1.5 Zeitschriften / Jahrbücher (Print und Online) ... 74
4.1.1.6 Loseblattsammlungen ... 75
4.1.1.7 CD-ROMs, DVDs (Audiovisuelle Medien) ... 76
4.1.2 Schema Mapping der IFF-Daten ... 77
4.2 Datenbereinigung der IFF-Daten ... 78
4.2.1 Herausfiltern von gleichlautenden Titeln ... 79
4.2.2 Strukturierung von Titel- und Autorenspalten ... 80
4.2.3 Standardisierung von Namen sowie Orten ... 81
4.2.4 Normierung nach GND ... 83
4.2.5 Normalisierung: Beispiel ISBN ... 84
4.2.6 Erkennung von Körperschaften ... 85

5 Technische Umsetzung der IFF-Daten-Deduplizierung ... 87

5.1 Beschreibung der Ausgangslage ... 87
5.1.1 Informationen zum IDSSG ... 87
5.1.2 Vorarbeiten des IDSSG ... 87
5.1.3 Konsequenzen für die Deduplizierung der IFF-Daten ... 88
5.2 Annahmen zur Datenintegration ... 89
5.2.1 Nur Original-Dokument des IFF vorhanden ... 89
5.2.2 Dublette im IDSSG vorhanden ... 90
5.2.3 Dublette in einem anderen Datenpool vorhanden ... 90
5.2.4 Mehrere Dubletten: Entscheidungstabelle ... 91
5.2.5 Problemfälle ... 93
5.2.6 Reihenfolge der Abfrage in swissbib / GVI ... 94
5.3 Entwicklung des IFF-Verfahrens ... 94

5.3.1 Überblick ... 95
5.3.2 Ablaufbeschreibung ... 97
5.3.3 Evaluation ... 101
5.3.4 Datenfusion / Export ... 103
5.4 Ergebnisse ... 104
5.4.1 Testdateien ... 104
5.4.2 Tests und Testergebnisse ... 105
5.4.3 Effektivität für swissbib ... 109
5.4.4 Effektivität für GVI ... 112
5.4.5 Effizienz ... 113
5.4.5.1 Effizienz im Vergleich mit anderen Verfahren ... 115
5.4.6 Fazit zur Qualität des Verfahrens ... 116

6 Zusammenfassung und Ausblick ... 117

6.1 Rückblick ... 117
6.1.1 Verwendete Werkzeuge ... 117
6.1.2 Zielerreichung ... 118
6.1.3 Gewonnene Erkenntnisse ... 119
6.2 Ausblick ... 120
6.2.1 Ausbaumöglichkeiten des Programms ... 120
6.2.2 Anreicherungsmöglichkeiten ... 121
6.2.3 Nutzung von Frameworks ... 122

7 Literaturverzeichnis ... 123

Anhänge ... 127

A. Rohdaten IFF ... 128

B. Konfiguration IFF ... 132

C. DVD ... 133

Selbständigkeitserklärung ... 134

Abbildungsverzeichnis

Abbildung 1: Datensatz aus Datenquelle 1 14
Abbildung 2: Datensatz aus Datenquelle 2 15
Abbildung 3: FRBR-Gruppe 1 (eigene Darstellung) 24
Abbildung 5: Deduplizierung im Fernleihe-Portal BOSS 53
Abbildung 6: Bibliothekslandschaft Schweiz ab 2021 (Mattmann, 2018) 56
Abbildung 7: Alma Extended Fuzzy Matching (eigene Darstellung) 59
Abbildung 8: Kandidatenliste IDSSG: Gruppenübersicht 62
Abbildung 9: Weboberfläche Dublettenkontrolle IDSSG 63
Abbildung 10: IFF-Katalog vor der Datenintegration 68
Abbildung 11: Einzelner Datensatz im IFF-Katalog 70
Abbildung 12: Beispiel einer Zeitschrift im IFF-Katalog 75
Abbildung 13: CD-ROMs und DVDs im IFF-Katalog 77
Abbildung 14: Schema Mapping IFF-Daten (eigene Darstellung) 78
Abbildung 15: Clustern nach Titeln mit OpenRefine 80
Abbildung 16: Aufsplitten des Titels mit OpenRefine 81
Abbildung 17: Aufsplitten der Autoren mit OpenRefine 81
Abbildung 18: Levenshtein-Distance bei Orten mit OpenRefine 82
Abbildung 19: ngram-Ähnlichkeitsmass von Autoren mit OpenRefine 82
Abbildung 20: Phonetisches Ähnlichkeitsmass mit OpenRefine 83
Abbildung 21: GND-Anreicherung eines IFF-Professors 84
Abbildung 22: Normalisierung der ISBN 85
Abbildung 23: Prüfung von Körperschaften 86
Abbildung 24: Ablauf des Programms dedup.pl (eigene Darstellung) 97
Abbildung 25: Verteilung der bestmatch-Werte 100
Abbildung 26: Subroutine getMatchValue 102
Abbildung 27: Prüfung auf Titel, Untertitel und Alternativtitel 102
Abbildung 28: Prüfung auf Bandtitel 103
Abbildung 29: Laufzeitmessungen GVI (links) und swissbib (rechts) 115
Abbildung 30: IFF-Daten, Ansicht im CSV-Format (Originaldaten) 129

Anmerkung: Bei Abbildungen ohne Verweis oder ohne Vermerk «eigene Darstellung» handelt es sich um Screenshots aus dem Web oder dem IFF-Programm (Anhang C).

Tabellenverzeichnis

Tabelle 1: Felder für den Sortierschlüssel (Sitas & Kapidakis, 2008, S. 293). 26
Tabelle 2: Ergebnisse der Dublettenerkennung (Naumann, 2007, S. 2) 29
Tabelle 3: Charakteristik einiger Algorithmen (eigene Darstellung) 36
Tabelle 4: Übersicht d. Felder einiger Verfahren (eigene Darstellung) 37
Tabelle 5: Auszug Entscheidungstabelle (Hickey & Rypka, 1979, S. 134) 41
Tabelle 6: Erstellung der Schlüsselteile (Hickey & Rypka, 1979, 133-135) ... 43
Tabelle 7: Schlüsselmuster FRBR Work-Set (Hickey & Toves, 2009, S. 4) 45
Tabelle 8: Resultate des Pfeffer-Algorithmus (Pfeffer, 2014, S. 443) 47
Tabelle 9: KOBV-Verfahren mit 2 Gewichtungen (Kuberek, 1999, S. 19) 50
Tabelle 10: Gewichtungen im GVI (KOBV) (Lohrum et al., 2019) 52
Tabelle 11: Resultate SLSP-Deduplizierung TL2 (eigene Darstellung) 58
Tabelle 12: Übersicht Recall & Precision (eigene Darstellung) 66
Tabelle 13: Monografie (eigene Darstellung) 71
Tabelle 14: Mehrteilige Monografie, Fall A (eigene Darstellung) 72
Tabelle 15: Mehrteilige Monografie, Fall B (eigene Darstellung) 72
Tabelle 16: Stücktitel in monografischer Reihe, Fall A (eigene Darstellung) 73
Tabelle 17: Stücktitel in monografischer Reihe, Fall B (eigene Darstellung) 73
Tabelle 18: Analytische Aufnahme (eigene Darstellung) 74
Tabelle 19: Loseblattwerk (eigene Darstellung) 76
Tabelle 20: swissbib-Entscheidungstabelle bei mehreren Dubletten (eigene Darstellung) 92
Tabelle 21: Unterschiede Skript 1 - Skript 2 (eigene Darstellung) 106
Tabelle 22: Effektivität mit swissbib Schnittstelle (eigene Darstellung) 110
Tabelle 23: Werte für die Effektivitätsberechnung (eigene Darstellung) 111
Tabelle 24: Effektivität mit GVI-Schnittstelle (eigene Darstellung) 112
Tabelle 25: Effizienzmessung GVI/swissbib (eigene Darstellung) 114
Tabelle 26: Vergleich Datensätze pro Minute (eigene Darstellung) 116
Tabelle 27: Auszug nicht bereinigte Daten: Datei IFF_Katalog_Full.csv (eigene Darstellung) 130
Tabelle 28: Auszug bereinigte Daten: Datei IFF_Katalog_FULL_normalized.csv (eigene Darstellung) 131

Abkürzungsverzeichnis

CDL ..... California Digital Library
CQL ..... Contextual Query Language
CSV ..... Comma-separated values
DDR ..... Duplicate Detection and Resolution
DNB ..... Deutsche Nationalbibliothek
FRBR ..... Functional Requirements for Bibliographic Records
GLIMIR ..... Global Library Manifestation Identifier
GND ..... Gemeinsame Normdatei
GREL ..... General Refine Expression Language
GVI ..... Gemeinsame Verbünde-Index
HSG ..... Universität St.Gallen - Hochschule für Wirtschafts-, Rechts- und Sozialwissenschaften sowie Internationale Beziehungen
IDS ..... Informationsverbund Deutschschweiz
IDSSG ..... Bibliotheksverbund IDS St. Gallen
IFF ..... Institut für Finanzwissenschaft, Finanzrecht und Law and Economics der Universität St.Gallen
ISBN ..... Internationale Standardbuchnummer
ISSN ..... Internationale Standardnummer für fortlaufende Sammelwerke
IZ ..... Institutional Zone
KOBV ..... Kooperativer Bibliotheksverbund Berlin Brandenburg
LCCN ..... Library of Congress Control Number
MARC ..... Machine-Readable Cataloging
MDBUPD ..... Master Database Update
NZ ..... Network Zone
OCLC ..... Online Computer Library Center
RDA ..... Resource Description and Access
SLSP ..... Swiss Library Service Platform
SRU ..... Search/Retrieve via URL
XML ..... Extensible Markup Language

1 Einleitung

> ***"Multiple, yet different representations of the same real-world objects in data, duplicates, are one of the most intriguing data quality problems."*** *(Naumann & Herschel, 2010, S. 6)*

Dubletten gehören im Bibliothekswesen zum Alltag. Oft sind sie verhasst, manchmal aber auch erwünscht. Mit rasant zunehmenden Datenmengen wachsen gleichzeitig die Datenqualitätsprobleme und somit die Schwierigkeit des Erkennens von Dubletten. Seit den siebziger Jahren gibt es Verfahren zur Deduplizierung. Das automatisierte Erkennen sowie das Verarbeiten von Dubletten in nützlicher Frist beim Zusammenführen von kleinen und grossen Datenbeständen bleibt auch ein halbes Jahrhundert später eine Herausforderung.

Eine relativ kleine Datenintegration in einen Bibliotheksverbund ist die Grundlage dieser Masterarbeit. Die Ausgangslage sowie die Ziele dieser Arbeit werden nachfolgend erläutert. Anschliessend wird der Aufbau der Arbeit kurz dargelegt.

1.1 Ausgangslage und Motivation

Ausgangslage und Motivation dieser Masterarbeit ist eine Datenintegration eines Institutskataloges in einen Verbundkatalog.

Das Institut für Finanzwissenschaft, Finanzrecht und Law and Economics der Universität St.Gallen (IFF) möchte mit seinem Bibliotheksbestand (ca. 15'000 Datensätze) dem Katalog des Bibliotheksverbunds IDS St.Gallen (IDSSG)[1] beitreten. Der IDSSG ist ein Partner des Informationsverbunds Deutschschweiz (IDS)[2].

Der IFF-Bestand enthält unterschiedliche Dokumenttypen, mehrheitlich aus dem Bereich Steuerrecht. Der IDSSG verzeichnet bereits einen umfangreichen Bestand an Wirtschafts- und Rechtsliteratur. Eine grosse Anzahl von Dubletten ist daher zu erwarten.

[1] IDSSG: unisg.ch/de/universitaet/bibliothek/recherche/bibliothekskatalog/hsgverbund

[2] IDS: informationsverbund.ch/24.0.html

Die IFF-Bibliothek wird nicht von ausgebildetem Bibliothekspersonal betreut. Die IFF-Daten entsprechen daher nicht den bibliothekarischen Standards oder Normen, mit welchen im IDSSG gearbeitet wird. Auch liegen die Daten nicht in einem bibliothekarischen Standardformat vor, sondern nur als Textdatei.

Die IFF-Daten sollen in den Bestand des IDSSG integriert werden, möglichst ohne Dubletten zu erzeugen. Dabei sollen die Anforderungen des IDSSG an die Datenqualität berücksichtigt werden. Wo immer möglich, sollen die IFF-Daten ersetzt werden mit Daten, die nach bibliothekarischen Standards erfasst wurden. Diese standardisierten Daten sollen in einem grossen Datenpool gesucht werden. Es sollen möglichst viele Dubletten erkannt werden, um die Datenqualität der Originaldaten zu verbessern.

1.2 Ziel der Arbeit

Das Ziel dieser Masterarbeit ist es, für die Datenintegration des IFF-Bestands (*Kapitel 1.1*) ein geeignetes Deduplizierungsverfahren zu finden und zu parametrisieren. Es wird ein Kriterienkatalog erstellt, welcher die Anforderungen an das Verfahren definiert. Bestehende Deduplizierungsverfahren werden untersucht und aufgrund des Kriterienkatalogs überprüft. Das verwendete Verfahren bzw. dessen Ergebnis wird im Hinblick auf den Kriterienkatalog evaluiert.

1.3 Abgrenzung des Themas

Thema der Arbeit ist die Dublettenerkennung in bibliografischen Datenbanken. Oft wird auch von Matching oder «Zusammenführen, was zusammengehört» gesprochen. Das Merging, d.h. die Datenfusion, ist nicht Teil dieser Arbeit und wird daher nur am Rande behandelt.

1.4 Aufbau der Arbeit

In den folgenden Kapiteln werden die Grundlagen der Deduplizierung im Allgemeinen beschrieben sowie bestehende Deduplizierungsverfahren speziell für bibliografische Daten analysiert, um daraus ein eigenes Verfahren zu entwickeln.

Das angewendete Verfahren für den IFF-Katalog sowie dessen technische Umsetzung wird präsentiert und die Ergebnisse kommentiert.

Kapitel 2 befasst sich mit den Grundlagen der Datenintegration und der Deduplizierung. Es werden einige Begriffe im Zusammenhang mit Dubletten definiert sowie die allgemein empfohlene Vorgehensweise einer Datenintegration dargelegt.

Im *Kapitel 3* wird ein Kriterienkatalog für die Entwicklung eines Deduplizierungsverfahren präsentiert sowie eine Auswahl bestehender Deduplizierungs-Algorithmen vorgestellt, evaluiert und auf ihre Eignung für das zu behandelnde Problem bewertet.

Kapitel 4 beinhaltet eine Datenanalyse der IFF-Daten und erläutert die Datenbereinigungen, welche vorgenommen wurden.

Kapitel 5 beschreibt die Umsetzung des gewählten Deduplizierungsverfahrens für die Datenintegration des IFF-Kataloges, im Speziellen die technische Entwicklung des Programms sowie die Ergebnisse.

Im *Kapitel 6* werden die Erkenntnisse zusammengefasst und ein Ausblick auf aktuelle und zukünftige Anwendungen von Deduplizierungsverfahren gegeben.

Anmerkung der Autorin:

In der folgenden Arbeit wird aus Gründen der besseren Lesbarkeit die männliche Form verwendet. Sie bezieht sich auf Personen beiderlei Geschlechts. Ausserdem wird die Rechtschreibung nach Schweizer Hochdeutsch[3] ohne Eszett (ß) verwendet.

[3] Siehe dazu wikipedia.org/wiki/Schweizer_Hochdeutsch sowie blog.supertext.ch/2015/07/der-unterschied-zwischen-deutschem-und-schweizerischem-hochdeutsch/

2 Datenintegration und Deduplizierung

Um die Grundlagen der Deduplizierung zu verstehen, wird im vorliegenden Kapitel die allgemeine Vorgehensweise (Prozess) einer Datenintegration beschrieben. Ausserdem werden wichtige Begriffe rund um das Thema Dubletten und Deduplizierung erklärt.

Die Datenintegration ist der Prozess des Zusammenführens von Daten aus mehreren Quellen sowie der einheitlichen Darstellung der Daten *(Bleiholder & Schmid, 2015, S. 121)*. In der Bibliothekswelt sind Datenintegrationen allgegenwärtig: Einspielen von Verlagsmetadaten von E-Medienpaketen, Integrieren von Benutzerdaten aus einem Hochschulinformationssystem oder die Fusion von mehreren Bibliothekskatalogen, wie in *Kapitel 1.1* beschrieben. Im Folgenden wird bei einer Datenintegration von einer Fusion mehrerer Bibliothekskataloge ausgegangen.

Die Datenintegration als Prozess kann in folgenden Schritten zusammengefasst werden *(Bleiholder & Schmid, 2015, S. 123)*:

(1) Datenbereinigung (*Kapitel 2.2*)

(2) Schema Mapping (*Kapitel 2.3*)

(3) Dublettenerkennung (*Kapitel 2.4*)

(4) Datenfusion (*Kapitel 2.5*)

Natürlich entstehen bei einer Datenintegration auch Probleme. Diese werde insbesondere dadurch verursacht, dass die Quellen autonom und heterogen sind. Die Autonomie der Quellen bedingt dabei die Heterogenität der Quellen. *(Bleiholder & Schmid, 2015, S. 122)*. Unter Autonomie versteht man die unabhängige Erstellung und Pflege von Daten verschiedener Herkunft. Die Heterogenität wird im *Kapitel 2.1* erläutert.

Die **Deduplizierung** ist ein Teilprozess der Datenintegration und umfasst allgemein die automatisierte Eliminierung von doppelt oder mehrfach vorhandener Information und im technischen Sinne auch Optimierung des Speicherbedarfs durch das Entfernen von repetitiven und redundanten Informationen *(Stevenson, 2015)*. In dieser Masterarbeit wird für diesen Teilprozess der Begriff Deduplizie-

rungsverfahren (synonym: Deduplizierungs-Algorithmus), kurz: Verfahren (Algorithmus), verwendet. Die technische Umsetzung eines Deduplizierungsverfahrens wird als Programm bezeichnet.

2.1 Heterogene Daten

Unter Heterogenität versteht man allgemein Verschiedenartigkeit, Ungleichartigkeit, Uneinheitlichkeit im Aufbau oder in der Zusammensetzung *(Munzinger Archiv/Duden, 2012).*

Bei der Deduplizierung bezieht sich diese Verschiedenartigkeit auf Daten oder Datenquellen. Die Daten können sich in Format, aber auch in der Modellierung (Schema- oder Datenebene) unterscheiden *(Bleiholder & Schmid, 2015, S. 124).*

Autor/Hrsg.	**Kuiper Willem G.**
Titel	(East-West) Joint ventures: A special phenomenon in international tax law?
Zusatz	
Art des Schriftstücks	**Druckerzeugnis**
Schlüsselwort	**Ausländisches Steuerrecht** Staatenverbände, Staatengruppen
Erscheinungsort	**Amsterdam**
Verlag	**IBFD**
Erscheinungsjahr	**1988**
ISBN/ISSN	**90 70125 38 2**
Anzahl Seiten	**246**

Abbildung 1: Datensatz aus Datenquelle 1

Ein konkretes Beispiel für heterogene Daten wird in *Abbildung 1* (Datensatz aus Datenquelle 1) und *Abbildung 2* (Datensatz aus Datenquelle 2) dargestellt. Beide Datensätze beschreiben dasselbe Dokument. Die Datenquellen, aus denen die Beispieldatensätze stammen, werden in *Kapitel 4.1* und *5.1* genauer beschrieben. Anhand dieses Beispiels können die unterschiedlichen Heterogenitätsformen (*Kapitel 2.1.1 bis 2.1.3*) illustriert werden.

```
<?xml version="1.0" encoding="UTF-8"?>
<collection xmlns="http://www.loc.gov/MARC21/slim">
 <record>
  <leader>     cam a22        4500</leader>
  <controlfield tag="001">280991436</controlfield>
  <controlfield tag="003">CHVBK</controlfield>
  <controlfield tag="005">20180630063558.0</controlfield>
  <controlfield tag="008">130818s1988    ne ||||||m||| 00    |eng d</controlfield>
  <datafield tag="020" ind1=" " ind2=" ">
   <subfield code="a">90-70125-38-2</subfield>
  </datafield>
  <datafield tag="100" ind1="1" ind2=" ">
   <subfield code="a">Kuiper</subfield>
   <subfield code="D">Willem Gustaaf</subfield>
  </datafield>
  <datafield tag="245" ind1="1" ind2="1">
   <subfield code="a">(East-West) joint ventures</subfield>
   <subfield code="b">a special phenomenon in international tax law?</subfield>
   <subfield code="c">Willem Gustaaf Kuiper</subfield>
  </datafield>
  <datafield tag="246" ind1="1" ind2=" ">
   <subfield code="a">East-West joint ventures</subfield>
  </datafield>
  <datafield tag="260" ind1=" " ind2=" ">
   <subfield code="a">[S.l.]</subfield>
   <subfield code="c">1988</subfield>
  </datafield>
  <datafield tag="300" ind1=" " ind2=" ">
   <subfield code="a">XV, 261 p</subfield>
   <subfield code="c">21 cm</subfield>
  </datafield>
  <datafield tag="650" ind1=" " ind2="7">
   <subfield code="a">Gemeinschaftsunternehmen</subfield>
   <subfield code="0">(DE-588)4071698-3</subfield>
   <subfield code="2">gnd</subfield>
  </datafield>
  <datafield tag="650" ind1=" " ind2="7">
   <subfield code="a">Internationales Steuerrecht</subfield>
   <subfield code="0">(DE-588)4027451-2</subfield>
   <subfield code="2">gnd</subfield>
  </datafield>
 </record>
</collection>
```

Abbildung 2: Datensatz aus Datenquelle 2

2.1.1 Technische Heterogenität

Technische Heterogenität bedeutet unterschiedliche Hard- oder Software der Daten (bzw. Datenquellen) und ist heutzutage kein grosses Problem mehr. In Datenquelle 1 (*Abbildung 1*) können die Daten in Form einer Datei im Textformat «Comma-separated values» (CSV) aus der Datenbank extrahiert werden. In Datenquelle 2 (*Abbildung 2)* befinden sich die Daten in einer Datenbank, aus welcher sie über eine Schnittstelle im Extensible Markup Language (XML) Format extrahiert werden können.

2.1.2 Strukturelle Heterogenität

Strukturelle Heterogenität entsteht auf Schemaebene: gleiche Sachverhalte der Welt (Schema) werden unterschiedlich modelliert.

Im Beispiel beschreiben beide Datensätze eine Informationsressource mit einem Autor, Titel, Internationaler Standardbuchnummer (ISBN) und weiteren Elementen. Der Autor wird in Datenquelle 1 (*Abbildung 1*) in einem Feld «Autor» angegeben. Datenquelle 2 (*Abbildung 2*) hält den Autor in einem Feld (datafield) «100» mit Unterfeldern (subfields) «a» und «D» und differenziert somit nach Vor- und Nachnamen. Ausserdem wiederholt sich der Autor in anderer Reihenfolge im Feld «245», Unterfeld «c».

2.1.3 Semantische Heterogenität

Semantische Heterogenität entsteht auf Datenebene: modellierte Sachverhalte überlappen teilweise. Im Beispiel beschreiben die Datensätze aus beiden Datenquellen dieselbe Informationsressource. Es handelt sich demnach um eine bibliografische Dublette (*Kapitel 2.4.1*). Beispielsweise überlappen die Autoren, sie werden jedoch unterschiedlich geschrieben: in Datenquelle 1 (*Abbildung 1*) ist «Kuiper Willem G.» erfasst, in Datenquelle 2 (*Abbildung 2*) «Kuiper Willem Gustaaf». Dasselbe kann für den Titel oder die ISBN gesagt werden. Andererseits überlappen das Feld «Seiten» und Feld «300», Unterfeld «a» nicht, wobei hier von einem Eingabefehler in einer Datenquelle ausgegangen werden kann.

2.2 Datenbereinigung

Die Datenbereinigung befasst sich mit der Eliminierung oder Reduktion der Datenfehler, die sich aus einer Datenanalyse im Hinblick der in *Kapitel 2.1* aufgelisteten Heterogenitäts-Fälle ergeben. Dazu gehören auch verschiedene Verfahren zur Korrektur von Datenfehlern.

Beispiele von Datenbereinigungen sind einheitliche Formatierungen von Datumswerten, Separieren von Vor- und Nachnamen oder Festlegen von Wertebereichstabellen für akademische Titel *(Zwirner, 2015, S. 102).*

Zur Datenbereinigung gehört auch die Normalisierung und Standardisierung von Daten. Als **Normalisierung** wird im Zusammenhang der Deduplizierung das Normalisieren von Text verstanden, d.h. das Angleichen oder Überführen von Strings in eine bestimmte Form. Beispiele für Normalisierung sind das Entfernen von Sonderzeichen oder diakritischen Zeichen oder die Umwandlung in Kleinbuchstaben. Unter **Standardisierung** versteht man z.B. die einheitliche Schreibweise von Ortschaften oder Namen. Im bibliothekarischen Umfeld ist die Standardisierung von Ansetzungsformen mittels Normdateien, wie z.B. der Gemeinsamen Normdatei (GND)[4], üblich.

Konkrete Beispiele zur Datenbereinigung sowie Normalisierung und Standardisierung werden im *Kapitel 4.2* anhand der IFF-Daten gezeigt.

2.3 Schema Mapping

Nach der Datenbereinigung folgt das Schema Mapping (teilweise in der Literatur auch Schema Matching genannt).

Ziel des Schema Mapping ist die Überwindung der Schemaheterogenität (*Kapitel 2.1.2*). Die unterschiedlichen Repräsentationen der Daten werden angeglichen. Es wird eine Abbildung erstellt, welche die jeweils gleichen Attribute zweier Quellen einander zuordnet (Mapping). Das Ergebnis ist eine vereinheitlichte Darstellung der beschriebenen Objekte *(Bleiholder & Schmid, 2015, S. 124)*.

Erschwerende Faktoren beim Schema Mapping sind (Auswahl):

- kryptische oder zu kurze Attributnamen
- grosse Schemata mit mehreren hundert Tabellen und Attributen
- das Auftreten von Synonymen (verschiedene Worte für dasselbe Konzept) und Homonymen (gleiche Worte für verschiedene Konzepte) bei Attributbezeichnungen

Ein Beispiel für das Schema Mapping mit den vorliegenden Beispieldaten (siehe *Abbildung 1* und *Abbildung 2*) ist das Mapping von Feld «Autor» zum Feld «100»,

[4] GND: dnb.de/DE/Professionell/Standardisierung/GND/gnd_node.html

Unterfeld «a». Ein vollständiges Schema Mapping der beiden Datenquellen ist im *Kapitel 4.1.2* dargestellt.

2.4 Dublettenerkennung

Bei der Überwindung der Datenheterogenität (*Kapitel 2.1.3*) helfen Techniken der Dublettenerkennung. Die Dublettenerkennung geschieht auf Datenebene. Dabei sollen unterschiedliche Repräsentationen ein und desselben Objektes erkannt werden. Zunächst soll definiert werden, was eine Dublette ist.

2.4.1 Definition von Dubletten

Dubletten sind Datensätze, die trotz Unterschieden in den Daten dasselbe Realweltobjekt beschreiben. Die Menge aller Datensätze, die dasselbe Realweltobjekt beschreiben, wird als Dublettengruppe bezeichnet. Dubletten entstehen unter anderem beim Zusammenführen von Datenbeständen sowie bei der Neuanlage und Änderung von Datensätzen *(Bleiholder & Schmid, 2015, S. 127)*.

Unter bibliografischen Dubletten in bibliothekarischen Metadaten versteht man zwei oder mehr Datensätze, welche dieselbe Informationsressource beschreiben. Sie entstehen insbesondere beim Katalogisieren in Verbünden bzw. Zusammenführen von Katalogen in Verbünden *(Sitas & Kapidakis, 2008, S. 287)*.

Bibliothekarische Metadaten umfassen sowohl die bibliografischen Beschreibungen einer Informationsressource auf Werkebene wie auch die Metadaten einer Expression, Manifestation oder eines Exemplars gemäss Functional Requirements for Bibliographic Records (FRBR) (*Kapitel 3.1.1*). Bibliografische Dubletten können demnach auf Werkebene, aber auch auf Manifestations- oder Expressionsebene definiert werden (in seltenen Fällen sogar auf Exemplar-Ebene, z.B. bei Handschriften oder anderen Unikaten).

Im weiteren Text werden mit Dubletten immer bibliografische Dubletten gemeint.

2.4.2 Entstehung und Konsequenzen von Dubletten

Gründe für die Existenz von Dubletten sind *(Sitas & Kapidakis, 2008, S. 288)*:

- unterschiedliche Katalogisierungspraktiken und Regelwerke
- menschliche Fehler (z.B. keine/unzulängliche Dublettenkontrolle vor dem Erfassen eines neuen Katalogisats, Perfektionismus des Bibliothekars)
- Syntax-Fehler im Datenformat

Dubletten verursachen diverse Probleme:

- Informationsüberflutung: Der Endnutzer bekommt beim Retrieval zu viele Dokumente angezeigt.
- Fernleihe: bei einer Bestellung eines nicht deduplizierten Titeldatensatzes werden nicht alle verfügbaren Dokumente berücksichtigt.
- Effizienz: Eine grössere Anzahl Datensätze als notwendig in der Datenbank bedeutet höhere Komplexität beim Indexieren.
- Katalogisierung: Zeitverschwendung durch das vorherige Identifizieren von Dubletten.
- Datenbankmanagement: Höhere Kosten für Speichern, Retrieval und Indexierung.
- Inventur: Probleme bei der eindeutigen Identifikation von Titeln.

Diese Probleme sind in Bibliotheken unerwünscht und konsequenterweise sollen Dubletten vermieden oder bereinigt werden.

2.4.3 Ähnlichkeitsmass von Attributwerten

Die grösste Herausforderung der Dublettenerkennung ist, dass Dubletten sich in den Werten, die das Objekt eindeutig beschreiben nur ähnlich, aber nicht gleich sind. Daher kommt es bei der Entdeckung von Dubletten darauf an, ähnliche Werte zu erkennen. Dabei spielt das Ähnlichkeitsmass eine wichtige Rolle.

In den meisten Fällen werden Objekte durch Zeichenketten (Strings) eindeutig benannt. Zur Ermittlung der Ähnlichkeit von Strings (Ähnlichkeitsmass) gibt es eine ganze Reihe von Algorithmen, teilweise spezialisiert für einzelne Anwendungsbereiche *(Bleiholder & Schmid, 2015, 130):*

- **Phonetische Ähnlichkeit:** Solche Algorithmen gibt es seit über 90 Jahren. Sie eignen sich zum Erkennen von Strings mit ähnlicher Aussprache, sind jedoch meist sprachspezifisch. Beispiel: Mayr, Meier, Mayer.
- **Editierdistanz oder Levenshtein-Distanz:** Geeignet für vertauschte oder hinzugefügte Buchstaben; ermittelt die minimale Anzahl des Hinzufügens, Löschens oder Vertauschens von Buchstaben, um einen String in einen anderen zu überführen. Beispiel: Edit-Distanz (Göthestr, Gothenstr) = 2 (1 Buchstaben vertauschen, 1 Buchstaben hinzufügen).
- **n-Gramm-Verfahren:** Zu einem String werden alle Teilstrings der Länge n betrachtet. Zum Prüfen der Ähnlichkeit wird bei n-Gramm-Verfahren die Anzahl gemeinsamer Teilstrings ermittelt. Beispiel: Bigramme für 'komm': _k, ko, om, mm, m_

Für den Vergleich von Zahlenwerten (Bsp. Seitenzahlen, Jahr) werden oft **Zahlenbereiche** verglichen, um kleinere Abweichungen zu finden. Beispiel: 2010-2012.

Diese Algorithmen müssen immer um anwendungsspezifisches Wissen ergänzt werden. Für alle Datenfelder, die bei der Dublettenerkennung verwendet werden, muss ein passendes Ähnlichkeitsmass gewählt und auf entsprechende Felder angewandt werden. Einige konkrete Anwendungen solcher Ähnlichkeitsmasse werden im *Kapitel 4.1* gezeigt.

2.4.4 Gewichtung von Attributen

Nicht alle Felder eines Datensatzes sind von gleicher Wichtigkeit bei der Erkennung von Dubletten. Um zu einem Gesamtwert für die Ähnlichkeit der beiden Datensätze zu kommen, werden die einzelnen Werte gewichtet: Jedem Datenfeld (Attribut) wird ein Gewicht zugewiesen, mit dem das Ähnlichkeitsmass multipliziert wird. Die Produkte werden aufsummiert. In manchen Anwendungsbereichen lassen sich mehrere Definitionen für Dubletten finden, die sich in den relevanten Datenfeldern oder den Gewichten zur Berechnung des Mittels unterscheiden. Die Möglichkeit, gleichzeitig nach Dubletten unterschiedlicher Definitionen zu suchen und ggf. die Suche zu beenden, wenn mit irgendeiner der Definitionen eine Dublette gefunden wurde, hat entscheidende Auswirkungen auf

die Performanz eines Systems zur Dublettenerkennung *(Bleiholder & Schmid, 2015, S. 132)*.

Ein Beispiel aus dem bibliothekarischen Umfeld ist im Matching-Verfahren des KOBV (*Kapitel 3.3.5*) beschrieben, welches mit Gewichtung von Attributen arbeitet.

2.4.5 Sorted Neighbourhood

Theoretisch müssen alle Datensätze untereinander verglichen werden, um alle Dubletten zu finden. Bei grossen Datenbeständen hat dies starke Auswirkungen auf die Laufzeit: Diese wächst quadratisch mit der Anzahl der Datensätze.

Ein Beispiel für einen solchen Brute-Force-Ansatz (ein Verfahren, welches alle Datensätze untereinander vergleicht) eines Deduplizierungs-Algorithmus zeigt, wie die Laufzeit eines Verfahrens mit dem Datenvolumen so stark wachsen kann, dass dessen Ausführung nicht mehr realistisch ist *(Lohrum, Kirchhoff, Reh & Winkler, 2019)*:

Komplexität für den Vergleich jedes Datensatzes mit jedem Datensatz:

```
N x (N-1) / 2
Laufzeit: O(n2)
```

Mit der Annahme, dass 10 Millionen Vergleiche pro Sekunde durchgeführt werden können, ergeben sich folgende Laufzeiten:

- 1000 Datensätze = 0,05 Sekunden
- 100‘000 Datensätze = 1,5 Stunden
- 10 Millionen Datensätze = 15 Monate
- 100 Millionen Datensätze = 16 Jahre

Dieser Brute-Force-Ansatz ist somit für grosse Datenbestände nicht praktikabel. Ein Lösungsansatz ist der Sorted-Neighbourhood-Algorithmus. Der Datenbestand wird so sortiert, dass potenzielle Dubletten nahe beieinanderstehen, so dass nur noch die Daten in der Nähe (Neighbourhood) nach Dubletten durchsucht werden müssen. Dies kann z.B. mit Sortierschlüsseln (*Kapitel 3.1.4)* erreicht werden. Der Sorted-Neighbourhood-Algorithmus beginnt mit der Erzeugung des Sortierschlüssels für jeden Datensatz. Anschliessend wird der Datenbestand

anhand des Sortierschlüssels sortiert. Bei geschickter Wahl des Schlüssels sind danach Dubletten dicht beieinander angeordnet. Zur Dublettenerkennung wird daher jeder einzelne Datensatz nur mit einer kleinen Gruppe von Datensätzen verglichen, die in der Sortierreihenfolge vor oder nach ihm stehen. Diese Gruppe wird auch Fenster genannt. Dadurch ist beim Sorted-Neighbourhood-Algorithmus eine wesentlich geringere Anzahl von Vergleichen notwendig. Statt einer quadratischen Laufzeitkomplexität ergibt sich mit dem Sorted-Neighbourhood-Verfahren eine konstante Laufzeitkomplexität beim Suchen von Dubletten *(Bleiholder & Schmid, 2015, S. 132)*.

2.5 Datenfusion

Der letzte Schritt der Datenintegration, die Datenfusion, «erzeugt eine bereinigte Tabelle, die keine Widersprüche, und idealerweise genau eine Repräsentation pro Realweltobjekt enthält» *(Bleiholder & Schmid, 2015, S. 134)*. Es soll folglich eine sinnvolle Anzeige eines deduplizierten Datensatzes für den Endnutzer erstellt werden. Es gibt viele Möglichkeiten, wie Datensätze zusammengeführt werden können. Da das Thema Datenfusion nicht Teil der technischen Umsetzung ist und den Rahmen dieser Arbeit sprengen würde, wird hier nicht näher darauf eingegangen (siehe auch *Kapitel 1.3*).

3 Deduplizierungsverfahren

Die Grundlagen zum Thema Dubletten und der Vorgehensweise zu deren Erkennung wurden im *Kapitel 2* aufgelistet. Doch wie kommt man nun von der Theorie zu einer Anwendung? Wie wird ein Verfahren zur Deduplizierung von bibliothekarischen Metadaten aufgebaut? Welche spezifischen Anforderungen müssen für das Deduplizierungsverfahren der IFF-Daten beachtet werden? In diesem Kapitel wird zunächst ein allgemein empfohlenes Vorgehen zur Entwicklung eines Deduplizierungsverfahrens präsentiert. Zudem wird ein Kriterienkatalog für die Integration des IFF-Bestands aufgestellt. Danach werden einzelne Deduplizierungsverfahren untersucht und in Bezug auf ihre Tauglichkeit für die IFF-Datenintegration kritisch beurteilt.

3.1 Entwicklung eines Deduplizierungsverfahrens

Deduplizierungsverfahren sind Verfahren, welche die Integrität von bibliografischen Datenbanken sichern. Da es einfach ist, ein Verfahren zu definieren, welches gleiche bibliografische Beschreibungen erkennt, jedoch schwieriger, ein Verfahren zu definieren, welches ähnliche bibliografische Beschreibungen erkennt, müssen bei der Entwicklung eines Deduplizierungsverfahrens bestimmte Punkte beachtet werden *(Sitas & Kapidakis, 2008, 288-290)*:

- Designziel
- Anwendungsschritte
- Auswahl der Felder
- Sortierschlüssel
- Evaluation
- Merging/Datenfusion
- Qualitätsmessung (Effektivität/Effizienz)

Zudem müssen die spezifischen Anforderungen der vorliegenden Datenintegration beachtet werden (Kriterienkatalog IFF-Bestand, *Kapitel 3.2*). Die konkrete Umsetzung des Prozesses anhand des Verfahrens für die IFF-Daten findet sich im *Kapitel 5.3*.

3.1.1 Designziel

Als Erstes soll der Begriff «Dublette» für das Deduplizierungsverfahren definiert werden. Wann ist ein Dokument eine Dublette, auf welcher Ebene soll dedupliziert werden? Dabei kommt das FRBR-Entitäten-Beziehungsmodell[5] zum Zug, welches in *Abbildung 3* am Beispiel vom ersten Band von «Harry Potter» dargestellt ist.

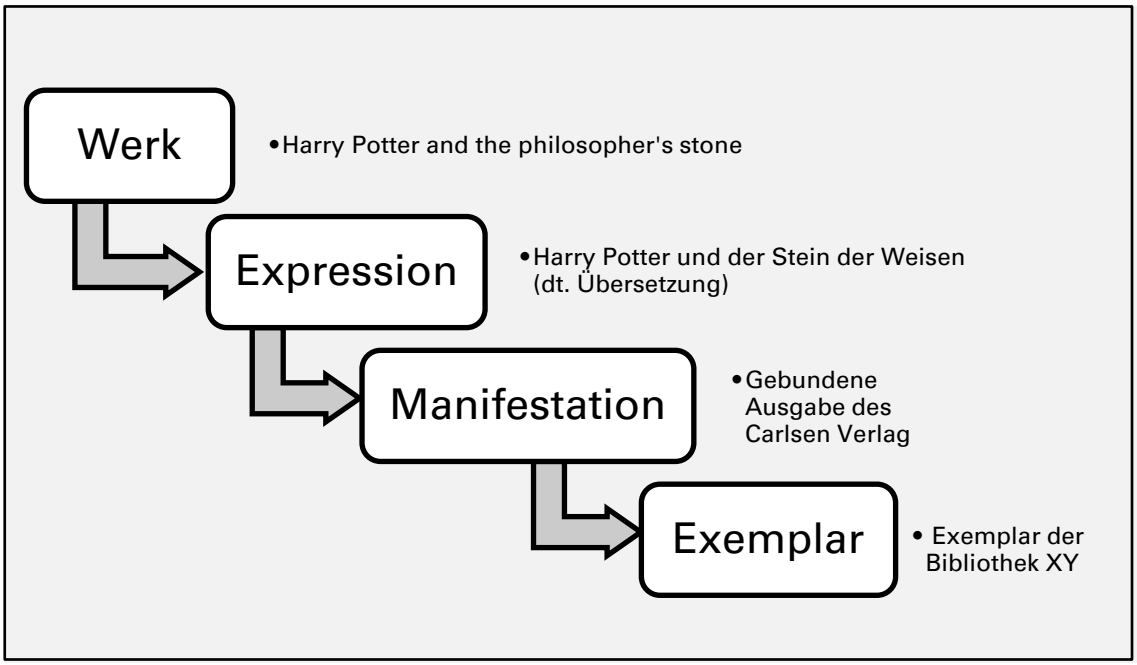


Abbildung 3: FRBR-Gruppe 1 (eigene Darstellung)

Viele Deduplizierungsverfahren erkennen Dubletten nur auf Werk-Ebene, andere Verfahren beachten Dubletten nur auf Manifestationsebene. Im Allgemeinen wird von Deduplizierung auf Werk-Ebene oder Titel-Ebene (=Manifestation) gesprochen.

Ausserdem muss geklärt werden, welche Dokumenttypen verarbeitet werden sollen. Oft eignet sich ein Verfahren nur für bestimmte Materialtypen (z.B. Monografien, Zeitschriftenartikel). Des Weiteren muss definiert werden, ob das Verfahren auch Hierarchien beachtet (z.B. mehrstufige Dokumente mit Oberaufnahmen, mehrbändige Werke mit eigenständigen Bandtiteln, etc.). Letztere beiden

[5] Auf die Functional Requirements for Bibliographic Records (FRBR) wird hier nicht weiter eingegangen, da dies den Rahmen der Arbeit sprengen würde. Mehr Informationen zu FRBR z.B. hier: ifla.org/VII/s13/frbr/frbr

Punkte sind stark abhängig vom bibliothekarischen Regelwerk, z.B. Resource Description and Access (RDA)[6], und Datenformat, z.B. Machine-Readable Cataloging Format (MARC)[7], in welchem die Dokumente abgespeichert sind.

3.1.2 Anwendungsschritte

Ein Deduplizierungsverfahren besteht aus einer Anzahl von Anwendungsschritten. Es gibt grundsätzlich zwei Varianten *(Sitas & Kapidakis, 2008, 288-290)*:

- **1-Schritt-Verfahren**: Dies ist meist ein Kompromiss, um eine schnelle und kostengünstige Deduplizierung zu erreichen. Ein 1-Schritt-Verfahren hat allgemeinere und weniger streng definierte Matching-Kriterien, es verbleiben viele Dubletten zur weiteren Verarbeitung. Normalerweise werden nur wenige Felder verglichen.
- **2-Schritt-Verfahren:** Hauptaspekt des ersten Schrittes ist es, die Anzahl der Vergleiche für den zweiten Schritt zu verringern und Fehlzuordnungen zu vermeiden. Beim 2-Schritt-Verfahren werden im zweiten Schritt nur noch mögliche Dubletten weiterverarbeitet. Die Zuordnungen des ersten Schrittes werden verifiziert und ein detaillierterer und genauerer Vergleich vorgenommen. Oft handelt es sich hier um ein Sorted-Neighbourhood-Verfahren.

3.1.3 Auswahl der Felder

Bei der Auswahl der Felder für die Vergleiche gibt es zwei Ansätze *(Toney, 1992, 21-22)*:

- Wenige Felder (oder nur Teile davon) nutzen, die intellektuell von Experten auf ihre Tauglichkeit überprüft wurden.
- Viele Felder nutzen, um den Review-Prozess durch einen Experten zu vermeiden und allfällige Datenfehler trotzdem auszugleichen.

Toney empfiehlt eine Evaluation der vorliegenden Daten, um die geeigneten Felder bzw. Feld-Teile zu finden *(Toney, 1992, 21-22)*.

[6] RDA: rda-rsc.org

[7] MARC-Format: loc.gov/marc/bibliographic

Die beste Methode zur Deduplizierung sind Kontrollnummern jeglicher Art, z.B. ISBN, sie sind jedoch nicht immer zuverlässig. Geeignete Felder sind Autor, Titel, Verlag, Seitenzählung, Ort und Jahr der Publikation *(Sitas & Kapidakis, 2008, 288-290)*.

Die meisten Verfahren verwenden folgende Felder in nachfolgender Häufigkeit (*Tabelle 1*):

Anteil	Feldbezeichnungen
90%	Autor, Titel, Publikationsjahr
70%	Seitenzahlen
60%	ISBN
50%	Verlag, Library of Congress Control Number (LCCN)
40%	Auflage
30%	Publikationsort, Reihe
20%	Publikationsland, Internationale Standardnummer für fortlaufende Sammelwerke (ISSN), Art der Vorlage
10%	Materialart, Sprache, Datenquelle, Band/Teil, weitere Kontrollnr.

Tabelle 1: Felder für den Sortierschlüssel (Sitas & Kapidakis, 2008, S. 293)

3.1.4 Sortierschlüssel (Matchkey)

Deduplizierungsverfahren verwenden Sortierschlüssel, um ähnliche Datensätze zu finden *(Kapitel 2.4.5)*. Bei Sortierschlüsseln handelt es sich um Strings, welche aus einem oder einer Kombination mehrerer Felder gebildet werden. Ein Feld kann als Teil oder Ganzes für den Schlüssel verwendet werden. Es können auch mehrere Felder oder Teilfelder als Schlüssel kombiniert werden. Vor der Erstellung dieser Schlüssel werden die Daten normalisiert, um Unterschiede wie Schreibfehler, fehlende Daten oder Schreibvarianten auszugleichen (*Kapitel 2.2*).

Beispiel für die Generierung eines Sortierschlüssels (hier nach dem Pfeffer-Algorithmus *(Pfeffer, 2014, S. 441)*, siehe *Kapitel 3.3.3.3*):

- Teil 1: Normierter Name des Autors und weiterer Beteiligter

- Teil 2: Einheitssachtitel oder konkatenierter Titel + Zusatz
- Sortierschlüssel werden für alle Kombinationen erstellt

Der folgende Datensatz:

```
Scheven, E., Kunz, M., & Bellgardt, S. (Eds.) (2012). Regeln für den
Schlagwortkatalog : RSWK. Frankfurt: Dt. Nationalbibliothek.
```

ergibt somit drei Sortierschlüssel:

```
scheven esther|regeln fuer den schlagwortkatalog rswk
kunz martin|regeln fuer den schlagwortkatalog rswk
bellgard sigrid|regeln fuer den schlagwortkatalog rswk
```

3.1.5 Evaluation

Es gibt zwei Methoden, wie Sortierschlüssel verglichen werden können *(Sitas & Kapidakis, 2008, 288-290)*:

- **Binärer Vergleich** (Field comparison): Es finden rein binäre Vergleiche von ausgewählten Feldern mit YES/NO-Angaben statt. Diese Methode ist sehr strikt und erschwert die Detektion von Datensätzen, welche sich z.B. in der Schreibweise unterscheiden.
- **Gewichtung**: Den String-Vergleichen werden je nach Feld unterschiedliche Gewichtungen und Werte zugewiesen. Nur wenn die Gewichtungswerte einen vorgeschriebenen Schwellenwert[8] erreichen, werden die Datensätze als gleichwertig erachtet. Diese Methode ist offener gegenüber Fehlern, Vollständigkeit oder Fehlen gewisser Felder. Beispiele für die gewichtete Evaluation sind der OCLC-Algorithmus (*Kapitel 3.3.2*) oder das Matching-Verfahren des KOBV (*Kapitel 3.3.5*).

3.1.6 Merging / Datenfusion

Nach dem Erkennen von Dubletten wird entschieden, wie mit den Dubletten verfahren wird *(Sitas & Kapidakis, 2008, 288-290)*:

[8] In dieser Masterarbeit wird der Begriff «Schwellenwert» synonym mit «Schwellwert» verwendet, welcher oft in Physik und Elektrotechnik gebraucht wird. Ein Schwellenwert ist der kleinste Wert einer Grösse, der als Ursache einer erkennbaren Veränderung ausreicht *(Munzinger Archiv/Duden, 2012).*

- Ein Datensatz wird als Master-Datensatz gewählt und alle andern gelöscht.
- Ein Datensatz wird als Master-Datensatz gewählt und alle nicht übereinstimmenden Felder der anderen Datensätze werden dem Master hinzugefügt (Merging, z.B. *Kapitel 3.3.6*).
- Alle Datensätze werden behalten, aber um einen Master-Datensatz geclustert (siehe z.B. im Fernleihportal BOSS, *Abbildung 4).*

Es gibt auch Variationen zu den genannten Praktiken, z.B., dass immer der älteste oder jüngste Datensatz behalten wird.

Ein anderer Ansatz ist, Dubletten nur im Retrieval-Prozess zusammenzuführen (Merging on the fly). Das Zusammenführen kann somit auch nur virtuell passieren (z.B. nur in der Resultate-Anzeige für den Endnutzer). Man nennt dies auch Online-Verfahren. Dagegen werden Verfahren, welche vor dem Retrieval durchgeführt werden, Offline-Verfahren genannt.

3.1.7 Qualitätsmessung: Effektivität und Effizienz

Die Qualität des Verfahrens kann u.a. an der Effektivität und Effizienz gemessen werden. Insbesondere zur Messung der Effektivität sind Testdaten notwendig, deren Inhalt und Anzahl vorhandener Dubletten bekannt ist. Nur mit dieser intellektuellen Vorarbeit durch eine Fachperson kann die Effektivität des Verfahrens getestet und gemessen werden.

Oft wird bei der Qualitätsmessung auch von Performanz gesprochen. Damit ist sowohl die Ausführung als auch Leistungsfähigkeit eines Programms gemeint, die mit standardisierten oder willkürlichen Massstäben beurteilt wird *(Fischer & Hofer, 2011, S. 665).*

3.1.7.1 Effektivität

Unter Effektivität versteht man Genauigkeit (Precision) und Vollständigkeit (Recall). Der Erfolg eines Algorithmus kann anhand dieser Werte gemessen werden.

Der Deduplizierungs-Algorithmus kann folgende Resultate ergeben *(Sitas & Kapidakis, 2008, 288-290):*

- **True positive**: Echte Dubletten. Manchmal wird noch zwischen exact match und partial match (vollständiges bzw. teilweises Matching) unterschieden.
- **False positive**: Datensätze werden als Dubletten angezeigt, obwohl es sich nicht um dasselbe Dokument handelt (false match).
- **False negative**: Dubletten, welche durch den Algorithmus nicht erkannt wurden (undetected match).
- **True negative**: Der Datensatz ist keine Dublette und wird auch nicht als solche erkannt.

Tabelle 2 zeigt die möglichen Resultate in übersichtlicher Form:

	Realität: Dublette	**Realität: keine Dublette**
Methode: Dublette	True positive	False positive
Methode: keine Dublette	False negative	True negative

Tabelle 2: Ergebnisse der Dublettenerkennung (Naumann, 2007, S. 2)

Das grössere Problem sind **false positives** (fälschlicherweise als Dubletten identifizierte Datensätze), da beim Zusammenführen ein Informationsverlust eintritt. Um dies zu vermeiden, sollte der Algorithmus so aufgebaut sein, dass möglichst keine false positives darunter sind. Andererseits sollte die Methode streng genug sein, damit nicht zu viele **false negatives** (unentdeckte Dubletten) verbleiben.

Precision entspricht dem Verhältnis der korrekt identifizierten Dubletten zu allen gefundenen Dubletten. Recall entspricht dem Verhältnis der korrekt identifizierten Dubletten zu den tatsächlichen Dubletten. Beide Metriken sind wie folgt definiert *(Naumann & Herschel, 2010, S. 61)*:

$$precision = \frac{|true\ positives|}{|true\ positives| + |false\ positives|} = \frac{|true\ positives|}{|gefundene\ Dubletten|}$$

$$recall = \frac{|true\ positives|}{|true\ positives| + |false\ negatives|} = \frac{|true\ positives|}{|tatsächliche\ Dubletten|}$$

Um die Effektivität zu messen, muss demnach die Anzahl der tatsächlichen Dubletten bekannt sein. Das ist in vielen Datenbeständen nicht möglich und basiert oft auf Schätzungen aus Stichproben.

3.1.7.2 Effizienz

Unter **Effizienz** versteht man z.B. Laufzeit oder Speicherbedarf eines Programms, d.h. der technischen Umsetzung eines Verfahrens. Unter Laufzeit versteht man die Zeit, in der ein Programm als Prozess geladen wird und läuft *(Fischer & Hofer, 2011, S. 513)*. Mit Speicherbedarf ist der benötigte Speicherplatz des Programms gemeint. Auch die Skalierbarkeit fällt unter den Bereich Effizienz. Unter Skalierbarkeit versteht man die Ausbaufähigkeit eines Programms (z.B. durch Zuschalten zusätzlicher Prozessoren) *(Fischer & Hofer, 2011, S. 825)*.

Die Effizienz eines Verfahrens ist v.a. bei Online-Verfahren von grosser Wichtigkeit, da das Retrieval schnell gehen muss. Bei Offline-Verfahren steht die Effizienz nicht an erster Stelle. Ausserdem ist in der heutigen Zeit der Speicherbedarf nicht mehr von so grosser Relevanz wie bei Programmen, welche aus der Ära der 70er Jahre stammen. Jedoch spielt auch heute noch die Grösse des Problems (Datenmenge) eine Rolle, insbesondere bei Laufzeiten, die linear oder höher anwachsen. Eine konstant (= linear) oder logarithmisch wachsende Laufzeit ist daher generell zu empfehlen.

3.2 Kriterienkatalog für die Integration des IFF-Bestands

Basierend auf *Kapitel 3.1* sollen für das IFF-Verfahren folgende Anforderungen erfüllt sein. Die einzelnen Kriterien entsprechen den *Kapiteln 3.2.1ff.* und werden im Verlauf der Arbeit entsprechend referenziert.

3.2.1 Designziel für das IFF-Verfahren

3.2.1.1 Ebene: Titelebene

Das IFF-Verfahren wird ein Deduplizierungsverfahren auf Titelebene. Die IFF-Daten sollen nach Möglichkeit auf Manifestations-Ebene, im Zweifelsfall (bei Ungenauigkeit der Daten) auf Expressions-Ebene dedupliziert werden. Dokumente mit gleichem Titel/Autor/Jahr oder gleicher ISBN sollen als mögliche Dubletten in Betracht gezogen werden. Unterschiede in der Expression oder Manifestation

werden als leichte Ungenauigkeit akzeptiert (z.B. Unterschiede in Auflage, Ausgabe, Materialart), da die Qualität der Originaldaten oft gar keine exakte Identifikation zulässt. Es wird davon ausgegangen, dass ein Zusammenführen – auch bei leichter Ungenauigkeit – datentechnisch gesehen immer eine Qualitätsverbesserung darstellt. Schwerwiegende Fehl-Zuweisungen (z.B. das Matching mit einem ähnlich lautenden Titel oder einem anderen Werk desselben Autors) sollen jedoch vermieden werden, da sie zu Datenverlust führen. Beispiele für leichte und schwerwiegende Ungenauigkeiten finden sich im *Kapitel 5.4.2.*

3.2.1.2 Dokumenttyp: Monografien und analytische Aufnahmen

Das IFF-Verfahren soll vornehmlich Monografien und analytische Aufnahmen unterstützen.

Ein grosser Anteil der vorkommenden Datentypen sind Monografien und Zeitschriftenartikel oder andere unselbständige Publikationen. In den Originaldaten sind keine Hierarchien vorhanden, in der Zieldatenbank jedoch schon. Die Deduplizierung soll nach Möglichkeit die Hierarchien berücksichtigen. Wo die Berücksichtigung des Materialtyps oder der Hierarchien nicht möglich ist, sollen die Originaldaten von der Deduplizierung ausgenommen und manuell bereinigt werden, um false positives zu vermeiden.

3.2.2 Anwendungsschritte: 2-Schritt-Verfahren

Das Verfahren soll in 2 Schritten erfolgen.

In einem ersten Schritt sollen mögliche Dubletten mit einer Suche in einem grossen Datentopf gefunden werden. Im zweiten Schritt soll nur noch das Suchfenster (Resultate-Set) berücksichtigt werden, um allfällige Dubletten zu identifizieren. Die Grösse des Suchfensters kann variieren.

3.2.3 Felder: Attribute mit sinnvollen Informationen

Folgende Felder sind in den IFF-Daten verfügbar und enthalten sinnvolle Informationen: Autor, Titel, Jahr der Publikation, ISBN, Verlag, Ort, Materialart. Da die Angaben der Seitenzahlen in den Originaldaten ungenau sind, können sie

zur Deduplizierung nicht verwendet werden. Bei mehrbändigen Werken sollen zudem Bandangaben berücksichtigt werden, bei analytischen Aufnahmen die Quelle. Es darf nicht zu viel Gewicht auf die Identifier (ISBN, etc.) gesetzt werden, da diese nicht immer stimmen.

3.2.4 Sortierschlüssel: keiner

Das Verfahren muss ohne Sortierschlüssel funktionieren.

Da kein Zugriff auf die Indexe der Zieldatenbank besteht, können keine Sortierschlüssel benutzt werden. Es muss ein Verfahren gefunden werden, welches auch ohne Sortierschlüssel ein geeignetes Fenster an zu vergleichenden Datensätzen findet.

3.2.5 Evaluation: gewichtetes Verfahren

Es wird ein gewichtetes Verfahren gesucht.

Es soll möglich sein, die Gewichtungen anzupassen. Für die Deduplizierung der IFF-Daten ist die Herkunft der Daten in der Zieldatenbank von grosser Wichtigkeit. Daten aus dem eigenen Verbund sollen immer bevorzugt werden, danach gibt es eine Kaskade von bevorzugten Verbünden, basierend auf Ähnlichkeit der Regelwerke und Fortschritt der «RDAfizierung[9]» der Daten.

3.2.6 Merging/Datenfusion: Offline-Verfahren

Es handelt sich um ein Offline-Verfahren. Am Ende soll ein Master-Datensatz feststehen.

Das Verfahren kommt nur einmal zum Einsatz, es gibt keine Neuzugänge und keine Online-Deduplizierung. Da für die Datenfusion bereits Instrumente im Verbund vorliegen, ist das Löschen bzw. Merging des Verlierersatzes nicht Teil des Verfahrens. Am Ende sollen die deduplizierten Datensätze im MARCXML-

[9] Mit «RDAfizierung» ist das automatisierte, nachträgliche Anpassen von Altdaten ans Regelwerk RDA gemeint (z.B. das automatische Einfügen der MARC-Felder 336, 337 und 338).

Format[10], sowie ein Mapping der auszutauschenden Dokumentnummern, vorliegen.

3.2.7 Qualitätsmessung für das IFF-Verfahren

Der IDSSG hat dieses Kriterium folgendermassen definiert: «Das Verfahren soll mit vertretbarem Aufwand ein gutes Resultat liefern.» Was damit gemeint ist, wird nachfolgend beschrieben.

3.2.7.1 Effektivität: Gutes Resultat

Unter einem guten Resultat ist eine Lösung zu verstehen, welche einerseits so genau wie möglich ist, andererseits so viele Dubletten wie möglich findet. Dies soll ohne bibliothekarische Autopsie (d.h. ohne physische Analyse der Dokumente) und ohne grössere manuelle Datenbereinigungen erreicht werden. Geringe Fehlerquellen werden akzeptiert, da die Qualität der Originaldaten oft gar keine exakte Identifikation eines Titels zulässt. Schwerwiegende Identifizierungsfehler sollen jedoch vermieden werden (*Kapitel 3.2.1*).

3.2.7.2 Effizienz: vertretbarer Aufwand

Unter vertretbarem Aufwand ist hierbei sowohl Rechenleistung als auch Programmieraufwand zu verstehen. Es sollen 15'000 Datensätze dedupliziert werden.

Da das Verfahren nur einmalig durchgeführt werden muss, ist die Effizienz des Programms (Laufzeit, Speicherbedarf) zwar relevant, hat aber nicht die höchste Priorität. Die technische Umsetzung (Programm) soll mit möglichst einfachen Mitteln realisiert werden, ohne eigene Datenbankumgebung oder Indexierungsvorgängen, welche grössere Frameworks benötigen würden. Der Aufwand für Programmierung/Entwicklungsumgebung sollte verhältnismässig zum kleinen Datenbestand sein und möglichst mit vorhandenen Mitteln im IDSSG realisiert werden. Das Programm sollte auf einem beliebigen Büro-Rechner laufen.

---

[10] MARCXML ist eine XML-Repräsentation von MARC-Daten. MARCXML-Standard: loc.gov/standards/marcxml

3.3 Existierende Deduplizierungsverfahren

Die klassischen Deduplizierungsverfahren haben ihren Ursprung in den Siebzigerjahren des letzten Jahrhunderts. Mit dem Aufkommen grosser Verbundkataloge und dem Zusammenführen von Daten aus unterschiedlichen Quellen und Systemen war der Bedarf nach einem automatisierten Verfahren für die Erkennung von Dubletten sowie deren Deduplizierung gross. Die in diesem Kapitel beschriebenen Verfahren basieren alle auf Techniken aus dieser Zeit, auch wenn sie kontinuierlich weiterentwickelt wurden. Dies zeigt sich auch in der Literaturrecherche: Viele Artikel zu diesem Thema stammen aus den Jahren 1970-2000. Danach flacht das Interesse am Thema ab, es entstehen neue Technologien, die Datenmengen nehmen mit der raschen Verbreitung des Internets enorm zu (Big Data).

Mit dem Aufkommen von Linked Data[11] und dem Entstehen von Discovery Portalen wie swissbib[12] oder WorldCat[13], die eine Vielzahl von Quellen und Dokumenttypen aggregieren, ändert sich auch der grundlegende Aspekt der Deduplizierung. Recherchiert man die Fachliteratur, so erkennt man deutlich, dass im letzten Jahrzehnt die Clustering-Verfahren auf Werkebene grosse Beachtung finden. Clustering-Verfahren fassen Objekte mit ähnlichen Eigenschaften zusammen. Im bibliothekarischen Umfeld bedeutet dies i.d.R. das Clustern eines Werks mit seinen unterschiedlichen Expressionen und Manifestationen *(Pfeifer & Polak-Bennemann, 2016, S. 150)*. Ziel der Clustering-Verfahren ist es, möglichst viele Informationen miteinander zu verknüpfen, sei dies über Linked Data, Normdaten oder einfach ein benutzerfreundliches Clustern in Suchportalen (z.B. Discovery, Fernleihe), um den Endnutzer nicht mit der explodierenden Informationsflut zu erschlagen. Die meisten Clustering-Verfahren setzen auf bestehenden, klassischen Deduplizierungsverfahren auf, bzw. gehen davon aus, dass eine Deduplizierung zwecks besserem Datenbankmanagement bereits stattgefunden hat. Ihr Hauptziel ist daher nicht die Reduktion von redundanten Datensätzen, sondern das Zusammenführen von gleichen Inhalten.

---

[11] Mehr Informationen, s.a. linkeddata.org

[12] swissbib: swissbib.ch

[13] Worldcat: worldcat.org

Die neuesten Verfahren gehen noch weiter beim Clustering und basieren auf künstlicher Intelligenz und maschinellem Lernen. Sie sind auf sehr grosse Datenmengen (Big Data) angewiesen und müssen trainiert werden. Subasic, Gvozdenovic und Jack schildern ein solches Verfahren für Mendeley[14] und verweisen im Abschnitt «Related work» auf andere, ähnliche Ansätze *(Subasic, Gvozdenovic & Jack, 2016, S. 141).* Diese Art von Verfahren wird für das IFF-Verfahren nicht berücksichtigt, da dafür grosse Datenmengen und entsprechende MapReduce-Frameworks[15] benötigt werden. Dieser Verfahrenstyp wird daher im folgenden Kapitel nicht aufgeführt.

Einige Verfahren sind offen dokumentiert (z.B. FRBR-Work-Set Algorithmus), jedoch ohne Implementierung. Wieder andere haben ihren Code veröffentlicht oder nutzen Open-Source-Frameworks.

Viele dieser Verfahren arbeiten mit Big-Data-Frameworks, deren Skalierbarkeit sich besonders für grosse Datenmengen eignet.

Die Auswahl der in *Kapitel 3.3.2ff*. beschriebenen Algorithmen hat keinen Anspruch auf Vollständigkeit. Auswahlkriterien waren:

- **Verfügbare Literatur und Dokumentation.** Über die meisten der nachfolgend beschriebenen Algorithmen wurde in der Fachliteratur berichtet. Teilweise wurde auch deren Effizienz und Effektivität beschrieben, was eine bessere Bewertung (*Kapitel 3.4*) erlaubt.
- **Umfeld und Zeitabschnitt**, in denen die Algorithmen angesiedelt sind: Es wurden Algorithmen aus der Anfangszeit (70er-Jahre), der Epoche der grossen Verbundkataloge (90er-Jahre) sowie aus dem Einflussbereich des IDSSG (lokal und national im Einsatz stehende Verfahren) ausgewählt.
- **Eignung für das IFF-Verfahren** (Kriterienkatalog, *Kapitel 3.2*): Es wurden überwiegend klassische Verfahren ausgewählt, welche auf Titelebene funktionieren. Werkbasierte sowie Clustering-Verfahren werden zwar kurz angesprochen, in die Auswertung jedoch nicht aufgenommen, da ein titelbasiertes Verfahren gesucht wird.

[14] Mendeley: mendeley.com

[15] z.B. Hadoop: hadoop.apache.org

Die Bezeichnung und Datierung der gewählten Verfahren wurden teilweise von der Autorin festgelegt, da sie in der Literatur nicht immer einheitlich benannt oder datiert sind.

3.3.1 Gemeinsame Eigenschaften

Die klassischen Deduplizierungs-Algorithmen haben grob zusammengefasst folgende Eigenschaften:

Matching-Ebene: Manifestation

Die Definition von Dubletten basiert in der Regel auf Manifestations-Ebene. Das Konzept des Werks ist zwar schon verbreitet, der Schwerpunkt der Deduplizierung basiert aber auf anderen Kriterien als das Clustering auf Werk-Ebene zur besseren Darstellung im Retrieval. Grund dafür sind namentlich die Datenhaltung (möglichst keine doppelten Einträge im Index) sowie die Effizienz der Datenbanksysteme, sowohl beim Retrieval als auch bei der Dateneingabe.

Dokumenttypen

Die meisten dieser klassischen Verfahren sind nicht für alle Dokumenttypen geeignet oder haben unterschiedliche Abläufe je nach Dokumenttyp. Viele der früheren Verfahren sind auf Monografien ausgerichtet und behandeln nur Print-Materialien (*Tabelle 3*).

	Hickey/Rypka *(3.3.2)*	**KOBV** *(3.3.5)*	**Swissbib** *(3.3.6)*	**IDSSG** *(3.3.8)*
Dokumenttypen	M	M / Z	M / Z	M / Z
Anzahl Schritte	2	2	2	2
Gewichtungen	x	x	x	x
Sortierschlüssel	x		x	x

Tabelle 3: Charakteristik einiger Algorithmen (eigene Darstellung)

Erläuterung zur Tabelle: M = Monografien / Z = Zeitschriften

Ablauf

Es werden in den meisten Fällen Sortierschlüssel erstellt und auf Grund von wenigen Kriterien eine Kandidatenliste gebildet. Innerhalb dieser Kandidaten werden oft sehr ausführliche Vergleiche basierend auf vielen Feldern gemacht. Dabei werden Felder mit klaren Werten (z.B. Ziffern) bevorzugt, da die Vergleiche einfacher sind als bei unscharfen Werten wie Buchstabenfolgen. Bei diesen Feldern ist der Normalisierungsaufwand höher, insbesondere bei der Bereinigung von Sonderzeichen. Dies führt dazu, dass Felder wie Paginierung bevorzugt werden, obwohl hier bekannterweise viele Ungenauigkeiten bestehen. Aufwändigere Algorithmen verwenden auch unscharfe Suchen (Fuzzy-String-Suche) mittels Ähnlichkeitsmassen. Die meisten Algorithmen arbeiten mit Gewichtungen der Felder (*Tabelle 4*).

Felder (Auswahl)	**Hickey/ Rypka**	**KOBV**	**swissbib**	**IDSSG**
Kontrollnr. (z.B. ISBN)	x	x	x	x
Format	x			x
Autor	x	x	x	x
Titel	x	x	x	x
Ausgabe	x	x	x	x
Publikationsort	x	x		
Verlag	x	x	x	x
Publikationsjahr	x	x	x	x
Paginierung	x	x	x	x
Band/Teil		x	x	
Reihe/Serie	x			x

Tabelle 4: Übersicht d. Felder einiger Verfahren (eigene Darstellung)

Technische Umsetzung

Die meisten der klassischen Verfahren basieren auf Skriptsprachen (z.B. Perl, Python). Insbesondere bei älteren Verfahren sind die Laufzeiten sowie der Speicherplatz der Programme für die Indexierung ein wichtiges Kriterium, oft auch zu Lasten eines besseren Matching-Resultats. Einige Verfahren sind nicht skalierbar auf grosse Datenbestände. Bis auf wenige Ausnahmen werden manuelle (Nach)-Kontrollen vermieden. Einige Verfahren sind jedoch so konzipiert, dass sie nur mit der Expertise einer Fachperson gut funktionieren.

Standards

Viele der nachfolgenden Algorithmen sind Eigenentwicklungen von grossen Verbünden. Sie nutzen gemeinhin Standards bei der Datenbereinigung und der Datenhaltung, sind jedoch meist nicht Open source im klassischen Sinne. Die wenigsten Implementierungen sind frei zugänglich, die meisten Urheber legen jedoch den Ablauf ihres Algorithmus offen, einige auch die Performanz (jedoch meist nach eigenen Messkriterien). Insbesondere jüngere Verfahren nutzen Standard-Bibliotheken und dokumentieren ihr Vorgehen oft auf Code-Sharing-Plattformen wie GitHub, z.B. swissbib[16].

Merging-Prozess

In den meisten Fällen werden Dubletten gelöscht oder zusammengeführt, um die Datenbank nicht unnötig aufzublasen. Damit keine Informationen unbeabsichtigt verloren gehen, müssen dabei oft Kompromisse in der Genauigkeit gemacht werden, um false positives zu verhindern. Neuere Verfahren führen Dubletten oft nur virtuell zusammen, ohne die Originaldaten zu löschen. Dabei geht es v.a. um ein besseres Retrieval, weniger um die Bereinigung der Datenbank.

3.3.2 Hickey und Rypka / OCLC (1979)

Ziel der Studie von Hickey und Rypka war die Entwicklung eines verbesserten Algorithmus zur automatischen Erkennung von Dubletten bei monografischen Datensätzen im Union Catalog des Online Computer Library Center (OCLC).

[16] swissbib auf GitHub: github.com/swissbib

Der daraus resultierende Algorithmus verwendet einen längenvariablen Schlüssel (durchschnittlich 53 Bytes pro Datensatz). Tests im OCLC Union Catalog hatten mehr als 5% des Bestandes als Dubletten erkannt *(Hickey & Rypka, 1979, S. 125)*.

Für die Entwicklung des OCLC-Algorithmus wurden folgende Kriterien festgelegt *(Hickey & Rypka, 1979, S. 127)*:

- vernünftige Laufzeit für die über 5 Millionen Datensätze
- Anwendbarkeit auch auf Neuzugänge zur Datenbank
- Matching nur aufgrund von vorab erstellten Sortierschlüsseln (ein Sortierschlüssel pro Datensatz)
- bessere Performanz als das bestehende Verfahren, insbesondere sollten Katalogisierungsunterschiede weniger streng betrachtet werden
- keine Notwendigkeit einer manuellen Dublettenkontrolle

Vorhergehende Studien, welche für den neuen Algorithmus berücksichtigt wurden, waren sowohl eigene Verfahren (z.B. OCLC Master Database Update (MDBUPD), 1976) als auch Verfahren anderer Universitäten (z.B. University of Illinois Study, 1979 sowie Oak Ridge National Laboratory Study, 1976) *(Hickey & Rypka, 1979, S. 128–130)*.

Methodologie

Der neue OCLC-Algorithmus sollte auf MDBUPD aufbauen. Diverse Studien und die Literatur hatten erwiesen, dass die benutzten Felder gut geeignet sind und zuverlässig funktionieren. Der MDBUPD-Algorithmus funktionierte sehr genau und machte wenig Fehler, jedoch übersah er zahlreiche Dubletten, da er sehr strenge Matchingkriterien hatte. Folgendes waren die grössten Probleme:

- Im Titel gab es zahlreiche Varianten, insbesondere im Untertitel.
- Verlag und Erscheinungsort wurden unterschiedlich erfasst (Abkürzungen, etc.). Diese Felder mussten bereinigt werden.
- Unterschiedliche Seitenzahlen machten einen grossen Teil der nicht erkannten Dubletten aus.

In diversen Test-Samples wurde der neue Algorithmus getestet und kontinuierlich verbessert *(Hickey & Rypka, 1979, S. 132)*.

Beschreibung des Algorithmus

Für jeden Datensatz wird genau ein Sortierschlüssel erstellt, der aus zwei Teilen besteht *(Hickey & Rypka, 1979, 133-135)*.

- Der **erste Teil** besteht aus Exact-Match-Feldern. Man entschied sich für vier Felder, welche häufig vorkommen und wenig Fehler beinhalten: Jahr, Art der Vorlage, Art des Datensatzes, Titelanfang (8 Zeichen). Bei Tests mit der ISBN stellte man fest, dass sich dieses Feld nicht gut eignet und viele Fehlresultate bringt. Der erste Teil des Schlüssels wird dazu verwendet, die Datensätze so zu sortieren, dass nur dieselben Schlüssel miteinander verglichen werden.
- Der **zweite Teil** des Schlüssels besteht aus den restlichen Vergleichsfeldern und muss nicht zwingend identisch sein, um als Dublette zu gelten. Der Algorithmus wird hierbei von einer Entscheidungstabelle gesteuert. Er beinhaltet 16 Möglichkeiten (siehe Zeile 1 in *Tabelle 5*) wie zwei Schlüssel zueinander passen können. Jeder Vergleich für jedes Feld ergibt einen von drei Werten: Exact Match (E), Partial Match (P), Mismatch (-), wobei die Werte in folgender Reihenfolge bewertet werden: E > P > -

Sobald alle Werte des Schlüssels bekannt sind, wird die Entscheidungstabelle (*Tabelle 5*) ausgelesen. Wenn alle Werte des Schlüssels grösser gleich alle Werte irgendeiner Spalte der Tabelle sind, handelt es sich um eine Dublette. Die detaillierten Angaben zu den einzelnen Schlüsselteilen (wie sie erstellt wurden und was die möglichen Resultate waren) befinden sich in *Tabelle 6*.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Art der Vorlage	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E
Art d. Datensatzes	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E
Titelanfang	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E
Jahr	E	E	E	E	E	E	E	E	E	E	E	-	E	E	E	E
Ort	-	-	-	E	-	-	-	E	-	-	E	E	-	-	-	-
LCCN	E	E	E	-	P	P	P	P	P	P	E	P	P	P	P	P
ISBN	P	P	P	P	E	E	E	-	P	P	P	E	P	P	P	P
Auflage	P	P	-	P	-	-	-	P	P	P	P	P	P	P	P	E
Reihe	-	-	-	P	-	E	-	P	P	P	P	P	P	P	E	P
Name	-	-	-	P	-	-	-	-	-	-	P	P	E	-	-	-
Seitenzahl	P	-	-	P	P	-	-	E	E	P	E	E	P	E	P	P
Verlag	-	P	-	P	-	-	-	P	-	-	-	P	P	P	P	P
Titel-Hash	-	-	P	E	-	-	P	E	-	P	-	E	P	P	P	E

Tabelle 5: Auszug Entscheidungstabelle (Hickey & Rypka, 1979, S. 134)

Resultate des Algorithmus

Die Performanz sowie Nützlichkeit des Algorithmus wurde mit mehreren Messungen überprüft *(Hickey & Rypka, 1979, 136-140)*. Über 80% der Datensätze aus einer Testprobe (ca. 200'000 Datensätze) passten zu keinem andern Datensatz (d.h. kein exact match). Grösstenteils (99%) waren die Dublettengruppen unter 6 Datensätze gross, mit extremen Ausreissern bis zu 112 Datensätzen (z.B. Titel wie «annual reports, proceedings»).

Erkannte Dubletten (true positives):

Es wurde ein Abgleich gegen eine Stichprobe von 234 bzw. 184 (ohne Reprints) manuell identifizierten Dublettenpaaren gemacht.

- Der Algorithmus identifizierte in beiden Varianten 127 Dubletten, was einer Erfolgsrate von 54% bzw. 69% entspricht (true positives).
- Auf den gesamten Union Catalog hätte der Algorithmus 7 – 8.9% Dubletten identifiziert (je nach Definition von Reprints als Dubletten).

Gründe für die Nicht-Detektion (false negatives):

- Abweichende Daten bei Reprints (25%)
- Unterschiedliche Titelanfänge (12%)
- Unterschiedlicher Publikationsort (18%)
- Abweichende Seitenzahlen (12%)

Im Vergleich mit MDBUPD fand der neue Algorithmus 50-90% mehr Dubletten als der alte mit oben erwähnter Dubletten-Stichprobe.

Präzisionsfehler (false positives):

Aus einer weiteren Stichprobe von 1000 manuell identifizierten Dublettenpaaren führte der Algorithmus 13 davon falsch zusammen (1.3%). Bei 5 davon handelte es sich um Dateneingabefehler. Mit einer Abwandlung der Wertetabelle (strengere Kriterien) konnte die Fehlerrate auf 0.1-0.2% reduziert werden, jedoch litt die Erfolgsrate enorm darunter, es wurden 24% weniger Dubletten gefunden.

Feld	MARC	Inhalt	Resultat	Erläuterung Partial Match
Art d. Vorlage	$008	Pos. 23 um Fotokopien und Originale zu unterscheiden	N / E	
Art d. Datensatzes	LDR	Pos. 06: z.B. Sprachmaterialien, kartografische Mat., Musikaufnahme	N / E	
Titelanfang	$245$$a, $245$$b	Vorherige Standardisierung / Normalisierung von Buchstaben, Abkürzungen, Lowercasing. Artikel werden entfernt. Danach werden die nichtleeren Zeichen 1,2,3,5,8,13,21 und 34 verwendet.	N / E	
Jahr	$260$$c, LDR	Das grössere der beiden Daten (um Reprints auszuschliessen).	N / E	
Ort	$260$$a	Die ersten 6 nichtleeren Zeichen. Vorherige Normalisierung von Schreibweisen und Abkürzungen	N / E	
Name	$100, $110, $700, $710	3-teilig: 1) Feldbezeichnung 2) Länge des Feldes (Nachname bei Autoren, ganze Länge bei Körperschaften etc.) / 3) 3-byte-Schlüssel: 2,1,0. Bsp. "David J. Rypka" = RYD.	N / P / E	Wenn 1) unterschiedlich oder wenn einer / beide Felder leer.
Verlag	$260$$b	Nach Bereinigung wird ein 61-bit-Schlüssel generiert mit Bigrammen (ohne Leerzeichen). Kein exakter Match benötigt.	N / P	Wenn Set aus Key1 ein Subset aus Key2
Titel-Hash	$245$$a, $245$$b	Ganzer Titel wird in ein 109-bit-Schlüssel gehasht mit Nicht-Wörterübergreifenden Trigrammen.	N / P / E	Wenn Set aus Key1 ein Subset aus Key2
ISBN (wenn vorh.)	$020		N / P / E	Wenn nicht in beiden Keys vorh.
Aufl. (wenn vorh.)	$250	2-teilig: 1) Zielpublikum (kein P möglich), 2) Auflage (nur Zahl, "fourth" wird erkannt als 4)	N / P / E	Wenn in 2) keine Zahl erkannt = 0; matched als P zu einer gefundenen Zahl

Tabelle 6: Erstellung der Schlüsselteile (Hickey & Rypka, 1979, 133-135)

3.3.2.1 DDR (1990)

1990 kreierte OCLC einen neuen Deduplizierungs-Algorithmus für Monografien und Zeitschriften namens Duplicate Detection and Resolution (DDR). Bei diesem zweistufigen, gewichteten Verfahren werden im ersten Schritt mögliche Dubletten anhand eines Sortierschlüssels geclustert. Der Sortierschlüssel besteht aus den Feldern LCCN, ISBN, Erscheinungsjahr, Seitenzahl, Autor, Verlag, gesamter Titel. Nur Datensätze mit demselben Schlüssel und identischer LCCN oder ISBN oder Datensätze mit demselben Schlüssel und mindestens zwei übereinstimmenden Elementen kommen als mögliche Dubletten in Frage. Im zweiten Schritt vergibt der Algorithmus Ähnlichkeitswerte von 0.0 (nicht identisch) bis 1.0 (absolut identisch). Es gibt ein Teil-Matching, wenn die Feldinhalte zu 85% übereinstimmen. Wenn keine automatische Entscheidung möglich ist, werden Fälle manuell untersucht.

DDR erbringt folgende Resultate:

- Clustering Recall: 96%
- Dublettenerkennung: 56%

OCLC hat daraus die DDR-Software entwickelt *(Sitas & Kapidakis, 2008, S. 298).* Das DDR-Verfahren wurde für das IFF-Verfahren nicht berücksichtigt, da die Dokumentation nicht offen zugänglich ist.

3.3.3 OCLC-Verfahren auf Werkebene (sowie Spin-offs)

Folgende Verfahren seien hier der Vollständigkeit halber erwähnt, können aber für IFF-Datenintegration nicht verwendet werden, da sie nur auf Werk-Ebene deduplizieren.

3.3.3.1 FRBR Work-Set (2009)

Der FRBR-Work-Set Algorithmus wurde von Hickey und Toves für OCLC (Worldcat) entwickelt, ist jedoch offen dokumentiert. Ziel des FRBR-Work-Set-

Algorithmus ist es, einen Sortierschlüssel (FRBR-Key) für jeden Datensatz zu erstellen, der eindeutig und zuverlässig ein Werk (*Kapitel 3.1.1*) identifizieren kann und dazu dient, Werke zusammenzuführen. Das Format wird ignoriert. Autor und Titel mit unterschiedlichen Ansetzungsformen werden in ihre bevorzugte Form überführt. Teile der Sortierschlüssel wie Namen/Titel werden mit Mapping Files (u.a. aus LCNAF[17]) verglichen und normalisiert *(Hickey & Toves, 2009, S. 3)*.

Es gibt vier mögliche Schlüsselmuster (*Tabelle 7*).

Schlüsselmuster	Vorkommen	Beispiel
<author>/<title>	97,961,220 (70.65%)	bjorling, jussi\1911 1960/opera arias and duets
<uniform title>	1,569,352 (1.13%)	10 commandments
/<title>/[one or more <name>]	26,559,404 (19.16%)	/bergler/bergler, friedrich
/<title>/<oclc number>	12,559,537 (9.06%)	/britain and antarctica/289903387

Tabelle 7: Schlüsselmuster FRBR Work-Set (Hickey & Toves, 2009, S. 4)

Autor/Titel-Kombinationen können anderen Autor/Titel-Kombinationen oder einem Einheitssachtitel zugeordnet werden. Wenn das erste Nachschlagen nichts zurückgibt, wird der Schlüssel in mehreren Schritten editiert, bis ein Match gefunden wird oder die maximale Anzahl Editierschritte erreicht ist *(Hickey & Toves, 2009, S. 7)*.

3.3.3.2 GLIMIR (2012)

Dem Global Library Manifestation Identifier (GLIMIR)-Verfahren liegen zwei bereits bestehende Algorithmen zugrunde: der FRBR-Work-Set Algorithmus (*Kapitel 3.3.3.1*) sowie DDR (*Kapitel 3.3.2.1*). Ziel des GLIMIR-Verfahrens war ein benutzerfreundliches Werk-Clustering von Datensätzen in Worldcat, da man annahm, dass Endnutzer hauptsächlich an gleichem Inhalt interessiert waren, unabhängig von der Manifestation oder Expression *(Gatenby, 2012).*

[17] Library of Congress Name Authority File (LCNAF): id.loc.gov/authorities/names

GLIMIR clustert insbesondere auch Datensätze, welche in unterschiedlichen Katalogisierungssprachen erstellt wurden. Durch das Clustern nach Inhalt innerhalb eines Work-Sets werden die Unterschiede (Übersetzungen, Auflagen, etc.) sichtbarer. GLIMIR hat nicht nur die Deduplizierung, sondern auch den FRBR-Algorithmus erheblich verbessert. Der FRBR-Algorithmus berücksichtigt nur Autor, Titel und Einheitssachtitel, GLIMIR zieht zusätzliche Elemente eines Datensatzes heran und kann Sonderfälle besser abbilden. GLIMIR punktet vor allem bei Werken ohne Autor, insbesondere Filme und Zeitschriften sowie Übersetzungen und Datensätze, welche in nicht-englischsprachigem Umfeld katalogisiert wurden. *(Gatenby, 2012).*

3.3.3.3 Magnus Pfeffer (2012)

Ursprung des Verfahrens von Magnus Pfeffer war nicht die Deduplizierung in grossen Verbundkatalogen, sondern das uneinheitliche (oder fehlende) Beschlagworten und Klassifizieren von Werken innerhalb von mehreren Auflagen/Ausgaben sowie in unterschiedlichen Verbünden.

Der von Pfeffer vorgeschlagene Algorithmus basiert auf der Annahme, dass die meisten Bibliotheken in deutschsprachigen Ländern die GND für die Beschlagwortung und die Regensburger Verbundklassifikation (RVK[18]) für die Klassifikation verwenden. Die Idee des Verfahrens ist, alle Ausgaben eines Werkes zu finden und ihre Schlagwörter und Klassifikationen zu aggregieren bzw. bei fehlenden Angaben zu ergänzen. Das Verfahren überprüft dieselben Felder wie der FRBR Work-Set Algorithmus *(Pfeffer, 2014, 440-441)*.

Der Algorithmus wurde auf mehrere deutsche Verbundkataloge angewendet (nur Monografien). Nach zwei Durchläufen wurde eine signifikante Zunahme von neu erschlossenen Dokumenten festgestellt, wie in *Tabelle 8* ersichtlich.

[18] Regensburger Verbundklassifikation (RVK): rvk.uni-regensburg.de

Verbund	Datensätze	Mit RVK	Mit SWD	RVK ergänzt	SWD ergänzt
SWB	13’330’743	4’217’226	4’083’113	581’780	957’275
Hebis	8’844’188	1’933’081	2’237’659	1’097’992	1’308’581
HBZ	13’271’840	1’018’298	3’322’100	2’272’558	1’080’162
BVB	22'685’738	5'750’295	6'055’164	2’969’381	2’765’967

Tabelle 8: Resultate des Pfeffer-Algorithmus (Pfeffer, 2014, S. 443)

Weitere Anwendungsmöglichkeiten sind das Ergänzen von Personennormsätzen beziehungsweise das automatisierte Zuordnen von nicht mit GND-Sätzen verknüpften Namenssätzen. Es genügt, wenn ein einziger Titeldatensatz aus einem Werk-Cluster mit dem richtigen individualisierten Personennormsatz verknüpft ist, um die Zuordnung auch für alle anderen Datensätze des Clusters zu übernehmen. Auch für das automatisierte Erstellen von Werk-Normsätzen (z.B. für die Verknüpfung von Print- und Online-Ausgaben) könnte der Algorithmus ausgebaut werden *(Wiesenmüller & Pfeffer, 2013, 626-628).*

Einschränkend muss dazu gesagt werden, dass für dieses Projekt nur mit Perl-Skripten gearbeitet wurde, welche in einem kleinen Umfeld gut funktionieren, jedoch nicht skalieren auf grosse Datensets. Dazu sind Big Data-Frameworks notwendig, wie z.B. Metafacture (*Kapitel 6.2.3*), die besser skalierbar sind für verteilte Anwendungen *(Pfeffer, 2014, S. 444)*.

3.3.4 Culturegraph (2013)

Culturegraph ist ein Projekt der Deutschen Nationalbibliothek (DNB)[19]. Es enthält die Metadaten der DNB, aller grossen deutschen Verbünde sowie des Österreichischen Bibliotheksverbunds und weiterer Partner. Im Jahr 2018 lag der Datenbestand bei über 160 Mio. Datensätzen. Das Ziel des Projekts ist die Vernetzung der Datenbestände. Culturegraph bietet unterschiedliche Anwendungen,

[19] Culturegraph: hub.culturegraph.org

insbesondere im Linked Data-Bereich *(Vorndran, 2018, S. 166)*. Für die Deduplizierung ist in diesen Anwendungen eine Werkbündelung (auch: Werkclustering) von Interesse. Bei der Werkbündelung werden die benötigten MARC-Felder extrahiert und teilweise bereinigt oder kombiniert. Daraus werden dann die Sortierschlüssel erzeugt. Dies geschieht mit Metamorph, einer Beschreibungssprache der Softwareplattform Metafacture[20], welche für Culturegraph entwickelt wurde. Metafacture bündelt die modular aufgebauten Programme, stützt sich auf bereits existierende freie Software, wie Apache Hadoop u.a., und ist gut dokumentiert. Durch den Einsatz von Hadoop, einem Framework zur Entwicklung von skalierbarer Software, lassen sich auch Datenmengen in der Grösse von Culturegraph verarbeiten *(Wiesenmüller & Pfeffer, 2013, S. 629)*.

Der Schlüssel wird auf mehrere Arten aus folgenden Feldern gebildet *(Vorndran, 2018, S. 171)*:

1) ISBN + Bandangabe + Titel/Einheitstitel + Publikationstyp

2) Titel + Titelzusatz + Ersteller + Bandangabe + Publikationstyp

Ein Werkschlüssel kann folgendermassen aussehen *(Vorndran, 2018, S. 173)*:

```
<record>
<isbnVolumeTitle>9781138026131-X-therootsoffootballhooliganism-book
</isbnVolumeTitle>
<titleCreator>rootsoffootballhooliganismhistoricalusociologicalstudydu
nningeric-X-book</titleCreator>
<titleCreatorAddedEntry>therootsoffootballhooliganismhistoricalusocio
logicalstudy-murphypatrick-X-book</titleCreatorAddedEntry>
</record>
```

Datensätze mit demselben Schlüssel werden zu Werkbündeln zusammengefasst. Da ein Datensatz teilweise mehrere Schlüssel besitzt, können die Bündel mittels Breitensuche[21] miteinander verknüpft werden. Diese Bündel können weiterverwendet werden, z.B. für Sacherschliessung (*Pfeffer-Verfahren, 3.3.3.3*), Normdatenanreicherung oder Darstellung in (Fernleihe)-Portalen. Diese Werk-Cluster können auf der Weboberfläche von Culturegraph angezeigt werden.

---

[20] Metafacture: github.com/metafacture/metafacture-core/wiki

[21] Die Breitensuche ist eine Baumsuche, bei der zunächst alle direkt unter dem Knoten liegenden nächsten Knoten durchsucht werden, statt einem Knoten nach dem andern in die Tiefe zu folgen (*A dictionary of computer science,* 2016).

Das Culturegraph-Verfahren ist für die Datenintegration des IFF nicht geeignet, da die Werkebene herangezogen wird.

3.3.5 KOBV (1999)

Im Unterschied zu anderen Verbünden in Deutschland gibt es im Kooperativen Bibliotheksverbund Berlin Brandenburg (KOBV) keine zentrale Nachweisdatenbank, sondern einen gemeinsamen Index für die KOBV-Suchmaschine. Das KOBV-Match-Verfahren wurde speziell für diesen gemeinsamen Index entwickelt. Es sollte vor allem zur optimierten Darstellung der Rechercheergebnisse dienen. Da die Pflege und Indexierung bei unterschiedlichen Institutionen liegen, wurde zunächst ein Verfahren gesucht, welches auf die Erstellung von Sortierschlüsseln, welche indexiert werden müssten, verzichtet *(Rusch, 1999, S. 2)*.

Beim ersten Einsatz zeigte sich, dass das Verfahren aufwändiger war als angenommen und in einem zweiten Index-Aufbau bereinigt werden sollte (insbesondere Parameteränderungen bei der Gewichtung). Ausserdem sollten später Zeitschriften miteinbezogen werden *(Kuberek, 1999, S. 3)*.

Der Algorithmus führt Dokumente auf Titel-Ebene zusammen und ist ein zweistufiges Verfahren (Search & Match). Um die Anzahl Vergleiche beim Matching zu verringern, wird ein Set von potenziellen Dubletten erstellt mittels einer **Search-Funktion**, welche folgendermassen aufgebaut ist *(Lohrum, 1999, 3-4)*:

```
get one word from title (no stopwords)
do a register scan for this word
if result set is <= 100 records, get records
if result set is >= 100, take another word from title
do a find for this word
repeat until there is an appropriate result set
```

Im zweiten Schritt, der Dublettenerkennung, werden Dokumente mit einer **Match-Funktion** verglichen, die sich wie folgt aufbaut *(Rusch, 1999, S. 3)*:

- Feldspezifische Normierung (Löschen, Trunkierung, Zeichenumsetzung)
- Feldspezifischer Vergleich (auf Gleichheit bzw. auf Ähnlichkeit)
- Feldspezifische Vergabe von Gewichten
- Berechnung des Gesamtgewichts
- Gesamtgewicht über Schwellenwert: es handelt sich um eine Dublette

Für die Normierung wurden zahlreiche Feldauswertungen und Versuche gemacht, bis die Normierungsfunktionen (abhängig vom Feld) optimal funktionierten.

Für die Match-Funktion wurde die Ähnlichkeit der Strings mit n-grams gemessen. Die n-gram-Funktion gibt die Ähnlichkeit der Strings als Nummer zwischen 0 und 1 zurück. Die Länge der n-grams (2-grams, 3-grams etc.) hat einen wichtigen Einfluss auf die Resultate. Je kürzer die Strings, desto besser funktionieren kurze n-grams (2-gram). Für gewisse Felder wie ISBN wurden spezielle Match-Funktionen geschrieben *(Lohrum, 1999, 5-7)*.

Nach mehreren Tests wurde das Verfahren mit zwei Gewichtungen umgesetzt (*Tabelle 9*).

MAB-Feld[22]	Feldinhalt	Match-Wert (+)	Nonmatch-Wert (-)
100, 200	Autor	30	50
331, 310, 335	Titel	70	70
403	Ausgabe	30	60
410	Erscheinungsort	20	30
412	Verlag	20	20
425	Erscheinungsjahr	30	60
433	Seitenzahl	20	40
455	Bandangabe	--	70
540-580	Standardnummern	70	60

Tabelle 9: KOBV-Verfahren mit 2 Gewichtungen (Kuberek, 1999, S. 19)

Die Match-Funktion addiert bei einem Match (d.h. Ähnlichkeit liegt über einem gewissem Schwellenwert) einen positiven Ähnlichkeitswert, bei einem Nonmatch einen negativen. Auch die Qualität der einzelnen Felder wird über Gewichtungen ausgewertet. Am Ende des Verfahrens steht der qualitativ beste Datensatz fest.

[22] Maschinelles Austauschformat für Bibliotheken (MAB): format.gbv.de/mab

Alle Funktionen sowie die Gewichtungen sind parametrisierbar, ohne dass der Algorithmus geändert werden muss *(Lohrum, 1999, S. 16)*.

3.3.5.1 GVI (2019)

Der Gemeinsame Verbünde-Index (GVI) enthält 170 Mio. Titel in einem zentralen Index. Er umfasst die bibliografischen Daten aller deutschen Verbünde, der ZDB[23] und der DNB. Die wichtigste Anwendung des GVI ist die Fernleihe, daher ist eine gute Deduplizierung wichtig. Dazu wird der GVI in die Fernleihportale der jeweiligen Verbünde eingebunden *(Deutsche Nationalbibliothek, 2019)*. Anforderung an das Deduplizierungsverfahren im GVI ist an erster Stelle die Qualität (möglichst alle Dubletten finden, aber keine false positives produzieren), aber auch eine gute Laufzeit der Deduplizierung *(Lohrum et al., 2019).*

Der KOBV nutzt den GVI für die Leitwegsteuerung (Fernleihe) im KOBV-Portal. Die Deduplizierung des KOBV im GVI basiert in groben Zügen auf dem gleichen Prinzip wie in *Kapitel 3.3.5* beschrieben. Sie geschieht in drei Schritten: Datenvorbereitung (Normalisierung), Kandidatensuche und Datensatz-Vergleich. Es gibt keine gemeinsamen Schlüssel, daher gestaltet sich der erste Schritt, die Kandidatensuche, etwas schwieriger. Um die Kandidatenmenge zu reduzieren, werden die Daten im KOBV zunächst partitioniert (z.B. aufgrund der Materialart), dann wird eine Suche mit Titelwörtern gestartet, bis das Suchfeld unter 100 Treffer fällt. Im Datensatzvergleich werden die Daten auf Feldebene mit feldabhängigen Operatoren verglichen (z.B. Trigramme für Strings, Identität für Zahlen). Den Feldern werden entsprechende Gewichtungswerte zugeteilt, wenn sie identisch sind, ein Feld fehlt oder die Felder nicht übereinstimmen (*Tabelle 10*).

[23] Zeitschriftendatenbank (ZDB): zeitschriftendatenbank.de

Feld	Typ	Pro1	Pro2	Con
Autor	String	40	10	30
Titel	String	70	0	30
ISBN/ISSN	Zahl	80	10	20
Jahr	Zahl	20	0	40
Ort	String	20	5	30
Herausgeber	String	20	5	20
Auflage	Zahl	10	5	5
Seitenzahl	Bereich	30	0	40

Tabelle 10: Gewichtungen im GVI (KOBV) (Lohrum et al., 2019)

Damit ein Datensatz als dublett gewertet wird, darf die Summe der Con-Werte nicht grösser als 39 sein und die Summe der Pro-Werte muss über 75 liegen.

Erfahrungen in der KOBV-Fernleihe zeigen, dass die Qualität des Verfahrens gut ist (weniger als 0.5% false positives). Eine vollständige Deduplizierung für die 30 Mio. Datensätze des KOBV kann in 48 Stunden realisiert werden. Für den gesamten GVI ist dieses Verfahren jedoch zu langsam. Hier kommen eigene Sortierschlüssel zum Einsatz für die Kandidatensuche *(Lohrum et al., 2019):*

```
1. Material:ISBN:Pubdate
2. Material:Author:Title:Pubdate:Publisher
3. Material:Author:Title:Pubdate
```

Gut sichtbar wird das Resultat der Deduplizierung im Fernleihe-Portal des BSZ (BOSS)[24]. Im Beispiel wurde nach «Perl Kochbuch» gesucht (*Abbildung 4*). Benutzer können die Dubletten gruppieren bzw. ein/ausklappen.

[24] BOSS: fernleihe.boss.bsz-bw.de

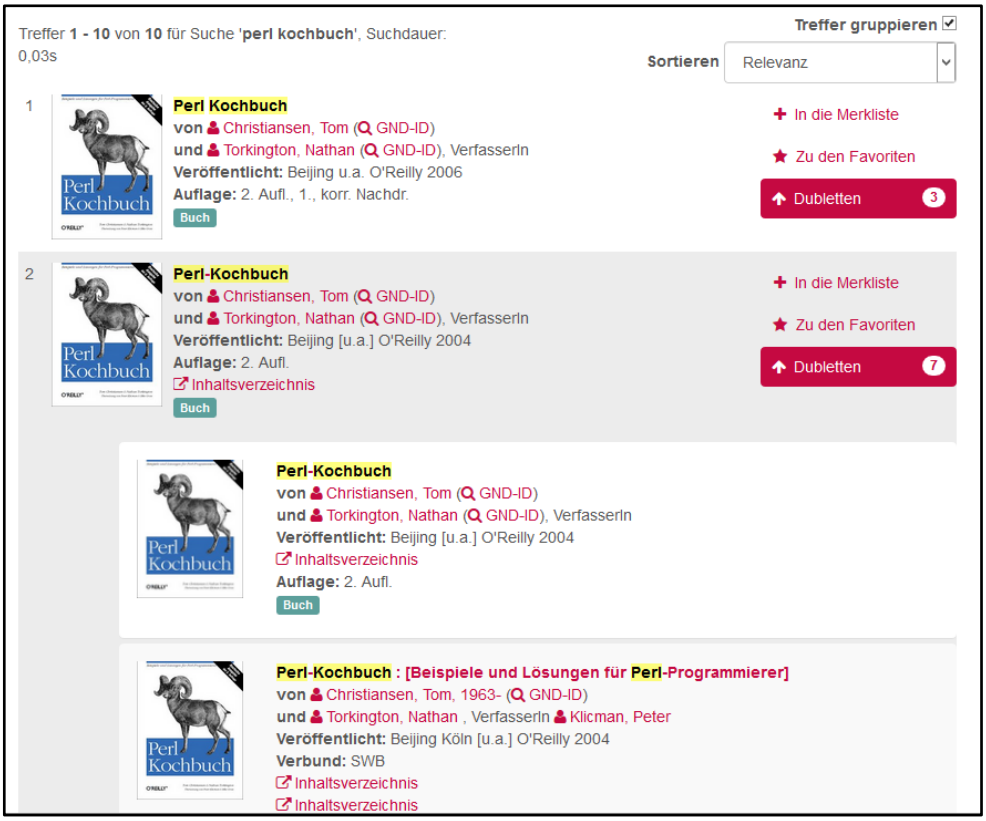


Abbildung 4: Deduplizierung im Fernleihe-Portal BOSS

3.3.6 swissbib (2008)

swissbib[25] ist der Metakatalog aller Schweizer Hochschulbibliotheken, der Schweizerischen Nationalbibliothek, zahlreicher Kantonsbibliotheken und weiterer Institutionen. Er bietet einen raschen, einfachen und umfassenden Zugang zu (wissenschaftlicher) Information in der Schweiz. Die Daten sind im Format MARC gespeichert, es werden jedoch unterschiedliche Regelwerke und Autoritätsdaten verwendet. Aktuell beinhaltet swissbib 30 Mio. Aufnahmen aus 17 Verbünden (970 Bibliotheken) sowie 6.5 Mio. Artikeln aus Nationallizenzen *(Witzig & Hipler, 2019)*.

swissbib ist ein Partner der Swiss Library Service Platform (SLSP)[26] und übernimmt eine wichtige Rolle bei der Zusammenführung, Bereinigung und Deduplizierung der Metadaten für SLSP (*Kapitel 3.3.6.1)*.

[25] swissbib: swissbib.ch

[26] Swiss Library Service Platform (SLS): slsp.ch

Die Plattform swissbib wird seit 2008 betrieben. Die Deduplizierung in swissbib geschieht nicht in den Datenbeständen der einzelnen Verbünde, sondern nur virtuell auf übergeordneter Ebene. Um einen Überblick über die zu erwartenden Schwierigkeiten zu erhalten, wurde im Jahr 2007 eine Machbarkeitsstudie erstellt, welche zum Schluss kam, dass man mit dem entworfenen Algorithmus weiterarbeiten konnte. Der Algorithmus wurde seither kontinuierlich verbessert und ist immer noch im Einsatz.

In der Machbarkeitsstudie wurden folgende Resultate publiziert *(Swissbib, 2007):*

- 5.3% «false duplicates» (Annahme: false positives)
- 2.2% «false non-duplicates» (Annahme: false negatives)
- 9.2% «probable duplicates» (Annahme: true positives)

Es wird davon ausgegangen, dass die Bezeichnungen in der Studie den Begriffen in Klammern (siehe *Kapitel 3.1.7.1*) entsprechen. In der Studie werden die Bezeichnungen nicht näher erläutert.

Als Werkzeuge zur Deduplizierung kommen bei swissbib Big-Data-Frameworks wie Hadoop und Metafacture (*Kapitel 3.3.4*) zum Einsatz.

swissbib spricht bei ihrer Deduplizierung von Clustering. Das swissbib-Verfahren beinhaltet mehrere Schritte:

- Vorarbeiten: Indexierung und Bereinigung der Kriterien
- Schritt «divide»: mögliche Dubletten finden
- Schritt «evaluate»: vergleichen und Merge-Entscheid fällen
- Einen Master bestimmen und die Informationen des Slaves (oder mehrerer Slaves) hinzufügen.

Der Schritt «divide» unterteilt die Datensätze in überschaubare Dublettengruppen mittels eines der folgenden Sortierschlüssel:

- Identifier (ISBN, ISSN) in Kombination mit Format- oder Datenträgerinformationen
- Sortierschlüssel aus gewissen Titelbuchstaben, in Kombination mit Format- oder Datenträgerinformationen

Der Schritt «evaluate» vergleicht nun innerhalb der Dublettengruppen bestimmte Felder der Datensätze miteinander *(Witzig & Hipler, 2019)*.

Die Vergleiche basieren auf folgenden Feldern *(Swissbib, 2018):*

- ID (ISSN, ISBN, etc.: Felder 020, 022, 024, 035)
- Publikationsjahr, Dekade, Jahrhundert
- Titel (Felder 245$a,b,n,p sowie 246 $a)
- Auflage (Feld 250$a)
- Autor/Körperschaften
- Verlag (Feld 260$b / 264$b)
- Umfang (Feld 300$a, +/- 1 Seite), Bandnummer
- Koordinaten / Massstab

Für jeden Vergleich gibt es drei Fälle:

- Nur einer der Datensätze hat einen Indexeintrag für das Kriterium.
- Beide Datensätze haben Indexeinträge, wovon mindestens einer übereinstimmt.
- Kein Indexeintrag stimmt überein.

In den ersten beiden Fällen ist ein Zusammenführen möglich, in letzterem Fall nicht. Die Ähnlichkeit wird basierend auf dem resultierenden Fall berechnet *(Witzig & Hipler, 2019).*

Es werden Schwellenwerte zwischen 0.0 und 1.0 definiert, wobei alles unter 0.77 keine Dublette ist, alles über 0.9 sicher eine Dublette ist und dazwischen vielleicht *(Viegener, 2010).* Aus den geclusterten Dubletten wird ein Datensatz basierend auf dem reichhaltigsten Datensatz gebaut (Master Record). Zusätzliche Informationen der andern Datensätze (Slave Records) werden hinzugefügt *(Swissbib, 2018).*

3.3.6.1 SLSP (2019)

SLSP wurde 2018 gegründet. Ziel von SLSP ist die Bereitstellung einer zentralen Dienstleistungsplattform für die wissenschaftlichen Bibliotheken der Schweiz, basierend auf einem zentral betriebenen Bibliotheksmanagementsystem sowie

einem gemeinsamen Katalog. Als Bibliotheksmanagement-Software wurde Alma[27], als Discovery-Software Primo VE[28] der Firma Ex Libris gewählt.

Beteiligt an SLSP sind wissenschaftliche Bibliotheken sowie Vertreter von Verbünden und Hochschulen aus der deutsch-, französisch- und italienischsprachigen Schweiz, inklusive des IDSSG. Die Bibliothekslandschaft Schweiz stellt sich ab 2021 folgendermassen dar (*Abbildung 5*):

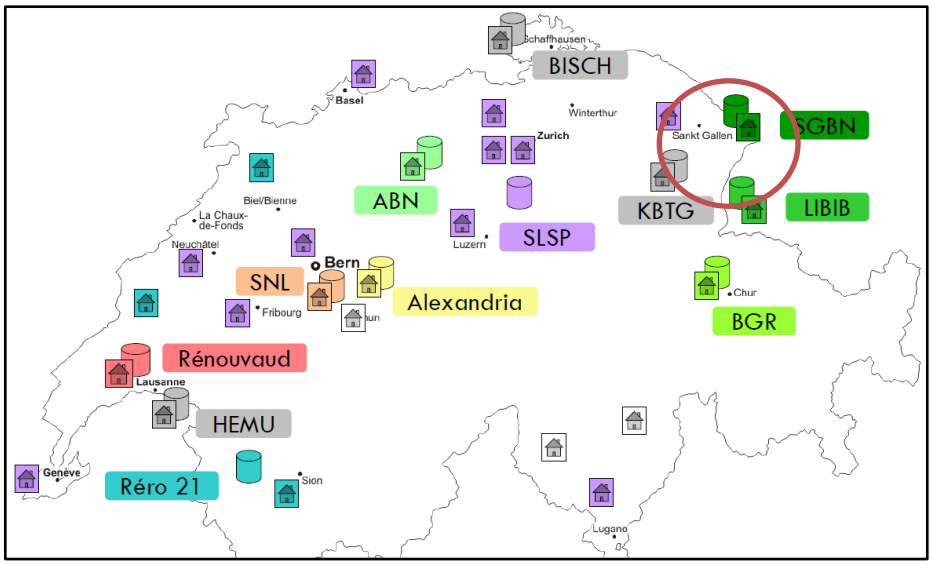


Abbildung 5: Bibliothekslandschaft Schweiz ab 2021 (Mattmann, 2018)

Die bibliografischen Daten der teilnehmenden Bibliotheken werden gemäss der für Alma üblichen Struktur in eine zentrale SLSP Network Zone (NZ) geladen. Die Verwaltungsdaten der Bibliotheken werden in einzelnen Institutional Zones (IZ) gehalten. Der IDSSG (siehe *Abbildung 5*, rote Markierung) wird ab 2021 als Verbund aufgelöst und seine Verwaltungsdaten (Ausleihen, Erwerbungs- und Exemplardaten) werden in verschiedene IZs aufgeteilt.

Aktuell befindet sich das SLSP-System in der Realisierungsphase (2018-2020). Zum Zeitpunkt der Einreichung dieser Masterarbeit (September 2019) findet die zweite von drei Testmigrationen statt.

Für die Zusammenführung aller Metadaten der beteiligen Institutionen zu SLSP wird der Deduplizierungs-Algorithmus von swissbib verwendet, nicht der Ex

[27] Alma (Ex Libris): knowledge.exlibrisgroup.com/Alma

[28] Primo VE (Ex Libris): knowledge.exlibrisgroup.com/Primo/Product_Documentation/020Primo_VE

Libris-Algorithmus *(Mattmann, 2018)*. Die deduplizierten Daten werden in die SLSP NZ geladen und über das Marc-Feld 035 (Nummern aus dem Ursprungs-System) mit den Lokaldaten der einzelnen Bibliotheken in den IZs verknüpft (*Basil Marti, persönliche Kommunikation, 6.3.2019*)

Das Verfahren ist aktuell, bis auf wenige Details, identisch mit dem swissbib-Verfahren (*Kapitel 3.3.6*). SLSP und swissbib gehen davon aus, dass sich dies noch ändern wird. Die Voraussetzungen bei SLSP unterscheiden sich vorrangig durch die kleinere Menge und etwas homogenere Bestände als die gesamten swissbib-Daten, was einen positiven Einfluss auf die Deduplizierung haben sollte (*Silvia Witzig, persönliche Kommunikation, 29.8.2019*).

Das Deduplizierungsverfahren geschieht, wie bei swissbib, in zwei Schritten. Im ersten Schritt werden potenzielle Dubletten identifiziert. Im zweiten Schritt findet eine Normalisierung aller Felder und dann ein detaillierter Vergleich statt. Ist festgelegt, welche Datensätze zusammengeführt werden, kommt der swissbib-Algorithmus zum Zug, welcher den Master-Datensatz sowie ausgewählte Felder der Slave-Datensätze bestimmt.

Da die Anpassungen für das SLSP-Deduplizierungsverfahren noch nicht abgeschlossen sind, liegt noch keine öffentlich zugängliche Dokumentation des Verfahrens vor. Für die zweite Testmigration im September 2019 (TL2) liegen folgende Ergebnisse vor (*Tabelle 11, Zahlen zusammengestellt aus: Silvia Witzig, persönliche Kommunikation, 27.8.19*). Das Merging wurde mit CBS[29] erstellt, welches in der neuesten Software-Version ein parallelisiertes Merging erlaubt.

[29] Software zur Datenaufbereitung: CBS - Metadata management solution von OCLC

Clustering (Stunden)	0.83
Initiales Merging (Stunden)	22
Anzahl verarbeitete Aufnahmen	23'011'993
Anzahl generierter Masterdatensätze	3'467'291
Anzahl nicht geclusterter Datensätze	14'307'626

Tabelle 11: Resultate SLSP-Deduplizierung TL2 (eigene Darstellung)

3.3.7 Alma und Primo / Ex Libris (2015)

Das Deduplizierungs-Verfahren der Firma Ex Libris für die Bibliotheksmanagement-Software Alma sowie das Discovery-Tool Primo wird in dieser Auflistung erwähnt, da es in vielen Bibliotheken zur Anwendung kommt. Es wird jedoch für das IFF-Verfahren nicht in Betracht gezogen, da die Software Alma sowie Primo im IDSSG noch nicht zur Verfügung stehen.

Ex Libris verwendet einen zweistufigen Ansatz für die Deduplizierung, Alma (Extended) Fuzzy Matching. Zuerst wird nach Dubletten gesucht, dann nach FRBR-Gruppen. Im ersten Schritt wird eine Dedup-ID erstellt, welche als Sortierschlüssel für den FRBR-Prozess genutzt wird *(Baksik & Koerber, 2019)*. Alma (Extended) Fuzzy Matching verwendet zwei unterschiedliche Matchroutinen für das Einspielen von neuen Datensätzen: Es gibt den Serial Match und den Non-Serial Match. Diese sind im Ex Libris Knowledge Center dokumentiert und können in den Importprofilen konfiguriert werden *(Ex Libris, 2015b)*. Das Verfahren ist in *Abbildung 6* zusammengefasst.

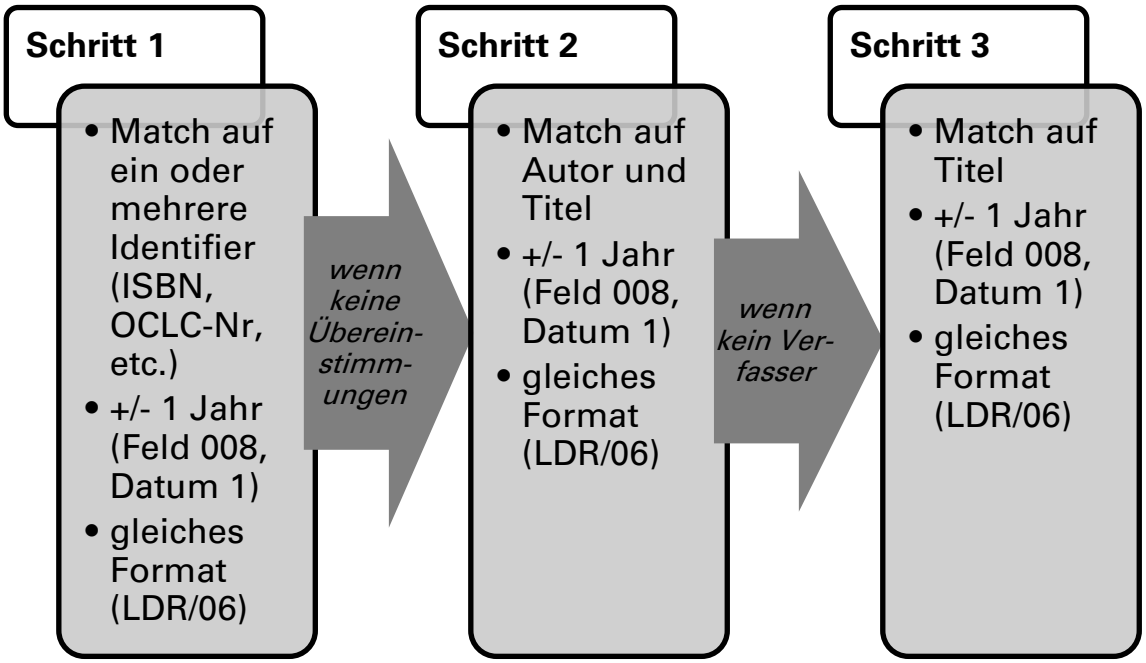


Abbildung 6: Alma Extended Fuzzy Matching (eigene Darstellung)

Deduplizierung in Primo und Primo VE

Der Deduplizierungs-Prozess in Primo basiert auf dem Erstellen eines Duplicate-Detection-Vektors für jeden Alma-Datensatz. Der Vektor enthält alle notwendigen Daten für den Deduplizierungs-Algorithmus. Die Vektoren enthalten ein oder mehrere Sortierschlüssel, welche den Datensatz identifizieren. Der Duplicate Detection Vector ist im Knowledge Center von Ex Libris ausführlich dokumentiert. Er wurde zusammen mit der California Digital Library (CDL) (*Kapitel 3.3.7.1)* entwickelt *(Ex Libris, 2015a)*.

In Primo VE (dem Nachfolgeprodukt von Primo) werden Datensätze nach FRBR-Prinzip auf Werk-Ebene gruppiert. Datensätze mit gleichen Sortierschlüsseln werden einer FRBR-Gruppe hinzugefügt und erhalten eine FRBR-ID. Jeder Datensatz kann nur zu einer FRBR-Gruppe gehören, daher werden Datensätze nach der ersten Zuordnung zu einer Gruppe nicht weiter mit andern Datensätzen verglichen. Die FRBR-Gruppen werden indexiert und dazu genutzt, Suchresultate mit derselben ID zu clustern *(Ex Libris, 2017).*

3.3.7.1 Melvyl / CDL (1992/2006)

Das Matching-Verfahren der California Digital Library (CDL) ist die Grundlage des Primo Duplicate Detection Vector (*Kapitel 3.3.7)*. CDL betreibt den Verbundkatalog Melvyl[30], der bis 2011 auf einem Bibliotheksmanagementsystem der Firma Ex Libris lief.

Der Melvyl-Algorithmus ist ein 2-Schritt-Verfahren und kann sehr unterschiedliche Datentypen, z.B. Musik *(Coyle, 1992, S. 5),* verarbeiten. Im ersten Schritt werden LCCN/ISBN, Publikationsjahr und die ersten 25 Buchstaben des Titels verglichen und es wird eine Gewichtung basierend auf der Ähnlichkeit vergeben. Erreicht ein Dokument den vorgegebenen Schwellenwert nicht, so werden weitere Felder verglichen (Titel, Hauptansetzung, Erscheinungsland, Seitenzahl, Verlag). Das Verfahren wird offline angewendet, wenn neue Dokumente in Melvyl hinzugefügt werden, mögliche Dubletten werden zu diesem Zeitpunkt identifiziert und in einem Merging Pool gespeichert. Beim Retrieval kommt das Verfahren dann nur noch «on the fly» zur Anwendung, indem für jede Suche dieser Merging Pool konsultiert wird und jeweils das passende Dokument angezeigt wird (Sitas & Kapidakis, 2008, S. 300).

Ziel von Melvyl ist es, für jede unterschiedliche Ausgabe eines Werks genau ein Union Record zu speichern (Coyle, 1992, S. 5). In den «Rules for Merging MELVYL records» wird eine ältere Version des Verfahrens im Detail beschrieben (Coyle, 1992).

3.3.8 IDSSG (2009)

Bei der Verschmelzung von bibliografischen Datenbanken innerhalb des IDSSG sowie Integration von externen Institutsbeständen wurde ein eigenes Vorgehensmodell für die Deduplizierung von Datensätzen entwickelt und bereits mehrfach angewendet. Das halbautomatische Deduplizierungsverfahren umfasst die folgenden drei Schritte *(Leu, 2009)*[31]:

- Ermittlung von potenziellen Dubletten (Kandidaten)

[30] CDL - Melvyl: cdlib.org/services/d2d/melvyl

[31] Quelle aus dem Intranet des IDSSG (geschützter Bereich, Zugangsdaten: ids / ids).

- Gruppierung der Kandidaten auf Ebene von ausgewählten Feldern
- Gruppen werden auf Grund von Stichproben gesamthaft als Dubletten / keine Dubletten markiert

Vor der Migration wird bestimmt, welche Kandidaten-Gruppen als Dubletten gewertet werden. Während der Migration werden für beide Datenbanken die Kurztitellisten erstellt. Durch Vergleich der Kurztitellisten wird eine Tabelle mit möglichen Kandidaten, gruppiert nach übereinstimmenden Attributen, erstellt. Alle Paare, die zu den vorgängig als Dubletten markierten Gruppen gehören, werden markiert. Ein Merge-Programm führt Dubletten zusammen.

Gruppierung der Datensätze

Für alle Titel-Datensätze der zusammenzuführenden Datenbanken wird ein komprimierter Satz erstellt, der folgende Elemente enthält:

- ISBN oder ISSN
- Titel (245 a/b), normalisiert
- Autor (100a/700a/710ab), normalisiert
- Zeitschriften: Verlag (260b) statt Autor
- Auflage (250a, nur Ziffern)
- Ausgabejahr (260c, 4 Ziffern)
- Nummer innerhalb Serie (490v)
- Seitenzahl (300a, nur Ziffern)
- Material (245h)

Die komprimierten Sätze werden dann verglichen:

- Wenn die ISBN übereinstimmt, wandert das Paar direkt in die Kandidatenliste, wobei bei abweichenden 100/245/260/250/490/300-Feldern weniger signifikante Gruppen-Nummern vergeben werden.
- Stimmt die ISBN nicht überein oder fehlt diese, wird aus dem ersten Wort des Autorenfeldes, den ersten 30 Zeichen des Titels und dem Erscheinungsjahr ein Schlüssel gebildet. Stimmen die Schlüssel überein, wandert das Paar in die Kandidatenliste, wobei die Gruppennummer gemäss gleichen Feldern in den beiden komprimierten Sätzen ergänzt wird.

Die Kandidatenliste und die Datenbank-Tabelle wird von einem Perl-Skript angelegt und befüllt *(Leu, 2009)*[32].

Gruppe	ISBN	Titel	Autor	Material	Jahr	Auflage	Band	Seiten	Körper-schaft	Neben-autor	noch offen	dublett	versch.	Aktion
1023	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	0	9577	0	zeigen setzen
1022	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	0	105	0	zeigen setzen
1021	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	0	32	0	zeigen setzen
1020	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	0	1	0	zeigen setzen
1019	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	821	0	0	zeigen setzen
1018	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	20	0	0	zeigen setzen
1015	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	0	204	0	zeigen setzen
1014	✓	✓	✓	✓	✓	✓	✗	✓	✓	✗	0	1	0	zeigen setzen
1013	✓	✓	✓	✓	✓	✓	✗	✓	✗	✓	0	1	0	zeigen setzen
1011	✓	✓	✓	✓	✓	✓	✗	✗	✓	✓	22	0	0	zeigen setzen
1007	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	92	0	0	zeigen setzen
1006	✓	✓	✓	✓	✓	✗	✓	✓	✓	✗	0	2	0	zeigen setzen
1003	✓	✓	✓	✓	✓	✗	✓	✗	✓	✓	20	0	0	zeigen setzen
1002	✓	✓	✓	✓	✓	✗	✓	✗	✓	✗	1	0	0	zeigen setzen
991	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	480	0	0	zeigen setzen

Abbildung 7: Kandidatenliste IDSSG: Gruppenübersicht

[32] Quelle aus dem Intranet des IDSSG (geschützter Bereich, Zugangsdaten: ids / ids).

Kriterien zur Bestimmung potenzieller Dubletten

Kernstück der Dublettenkontrolle ist eine Weboberfläche (*Abbildung 8*), welche mögliche Dubletten nebeneinander anzeigt, wobei übereinstimmende und unterschiedliche Hauptelemente farblich hervorgehoben sind. So lässt sich recht schnell entscheiden, ob ein Paar in den Dubletten-Topf gehört oder nicht.

Dieses manuelle Verfahren ist für grosse Datenbestände keine Option. Bei der Verschmelzung von grösseren Datenbeständen wird deshalb nur mit der Gruppenübersicht *(Abbildung 7)* gearbeitet.

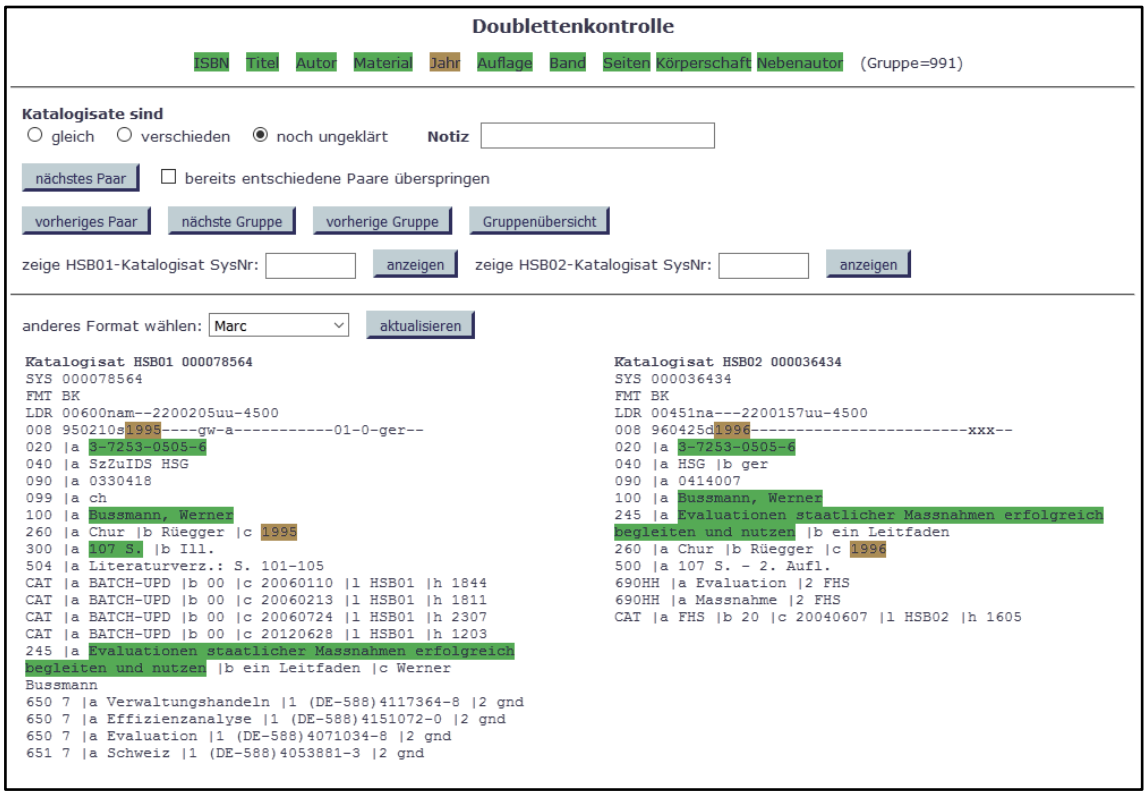


Abbildung 8: Weboberfläche Dublettenkontrolle IDSSG

Ob eine Gruppe als Ganzes eine Dublette ist, wird durch Stichproben bestimmt. Bei zwei grossen Merge-Projekten im IDSSG hat sich dieser Ansatz bewährt. Es hat sich bei den meisten Gruppen empirisch gezeigt, dass die abweichenden Elemente innerhalb einer Gruppe für eine deutliche Mehrheit der Gruppenmitglieder die gleiche Ursache haben. Diese Ursache kann durch eine Fachperson überprüft werden.

Der Ablauf des Deduplizierungsverfahrens im IDSSG unterscheidet sich somit grundsätzlich von allen anderen hier beschriebenen Verfahren, da bei diesem

Verfahren das manuelle Eingreifen einer Fachperson (Daten- bzw. Katalogisierungsspezialist) notwendig ist.

3.4 Bewertung der Verfahren

In diesem Kapitel werden die einzelnen Verfahren allgemein sowie nach ihrer Eignung für die IFF-Datenintegration bewertet und allfällige Defizite und Weiterentwicklungspotenziale aufgezeigt.

Am Ende des Kapitels wird aufgrund dieser Evaluation ein geeignetes Verfahren als Grundlage für die IFF-Datenintegration gewählt.

3.4.1 Allgemeine Bewertung der Verfahren

Allgemein kann gesagt werden, dass aufgrund der Literatur das Verfahren von Hickey sehr empirisch und eng mit bibliothekarischem Fachpersonal entwickelt wurde, was auch auf verlässliche Resultate schliessen lässt. Zwar ist das Verfahren nicht mehr zeitgemäss (beispielsweise gelten die Einschränkungen in der Effizienz heute kaum mehr), dennoch bietet diese Dokumentation eine Fundgrube für die Normalisierung und Erstellung von Sortierschlüsseln, aber auch für die Bestimmung der Fenstergrösse für Dublettengruppen. Insbesondere der Vergleich der Werte mittels einer ausgeklügelten Entscheidungstabelle kann auch heute noch als interessanter Ansatz gewertet werden. Das Nachfolgeverfahren DDR dient auch als Vorlage für den FRBR-Algorithmus, GLIMIR und Pfeffer-Algorithmus und somit als Grundlage vieler moderner Clustering-Verfahren.

Beim KOBV-Matching-Verfahren werden ebenfalls Konzepte eingesetzt, die schon älter sind. Dennoch ist die Deduplizierung in ähnlicher Form heute für den GVI immer noch im Einsatz. Auch hier bietet die Literatur einen guten Einblick, wie ein Match-Verfahren entwickelt und verbessert wird. Die gute Dokumentation der Feldnormalisierung und Gewichtungen ist für das Entwickeln eines eigenen Verfahrens von grossem Vorteil. Die Besonderheit des KOBV-Verfahrens, keinen Sortierschlüssel zu indexieren, sondern mittels einer Search-Funktion die Dublettengruppen zu finden, ist für die Verwendung im eigenen Verfahren (*Kapitel 4*) ausschlaggebend.

Die Verfahren von swissbib, KOBV und des IDSSG sind immer noch – teilweise in weiterentwickelter Form - im Einsatz (wobei das Verfahren des IDSSG nur für einmalige Merging-Aktionen zum Zuge kommt). Der Deduplizierungs-Algorithmus von swissbib wurde kontinuierlich verbessert und wird vom Projekt SLSP so gut eingeschätzt, dass man ihn für die Zusammenlegung und Migration aller Schweizer Hochschulbibliotheken auf Alma verwendet (und somit dem Ex Libris-Verfahren vorzieht). Das muss nicht als Kritik am Ex Libris-Verfahren aufgefasst werden, jedoch ist der swissbib-Algorithmus für den zu migrierenden Datenbestand entwickelt worden und kennt Spezialitäten und Anwenderwissen, welche ein kommerzieller Anbieter nicht wissen oder nur mit grossem Aufwand dazu programmieren kann.

3.4.2 Effektivität der Algorithmen (Recall / Precision)

Die Evaluation der Effektivität eines fremden Algorithmus ist grundsätzlich schwierig, da vorausgesetzt wird, dass die Anzahl aller Dubletten (true positives) bekannt ist. Dies gelingt i.d.R. nur mit manuell (durch Fachexperten) evaluierten Stichproben, bei denen die Anzahl der echten Dubletten bekannt ist. Nur wenige Verfahren berichten darüber, ob der Algorithmus gegen ein solches Sample getestet wurde.

Für die evaluierten Algorithmen sind die Zahlen für Precision und Recall leider nicht vorhanden, unvollständig oder unklar dokumentiert. Die folgende Übersicht wurde aufgrund der Literatur sowie der eigenen Nachforschungen und Berechnungen nach bestmöglicher Interpretation der Zahlen erstellt.

Es zeigt sich, dass ohne aufwändiges (und in diesem Rahmen nicht realistisches) Nachforschen oder Nachprogrammieren der Algorithmen und Testen gegen ein bekanntes Sample ein Vergleich aufgrund von Precision und Recall nicht möglich ist.

Auf jeden Fall kann die Eigenentwicklung (*Kapitel 4*) nach diesen Kriterien gemessen werden, indem entsprechende Test-Samples mit bekannten Dubletten erstellt werden.

	Hickey/ Rypka *(3.3.2)*	**Swissbib**[33] *(3.3.6)*	**IDSSG** *(3.3.8)*
Datensätze gesamt	5 Mio.	21 Mio.[34]	700'000[35]
Dubletten-Sample (manuell überprüft)	234 / 184	700	
True positive (%)	7 – 8.9	9.2	
False positive (%)	0.1-0.2	2.2	
False negative (%)		5.3	
True negative (%)	80		
Recall gemäss Literatur (%)	54-69		
Recall berechnet (%)		63	
Precision gemäss Literatur (%)			
Precision berechnet (%)	98	80	
Manuelle Kontrolle notwendig?			x

Tabelle 12: Übersicht Recall & Precision (eigene Darstellung)

Wie aus *Tabelle 12* ersichtlich ist, sind die Zahlen nur bei OCLC (Hickey & Rypka) und swissbib publiziert. Hinzu kommt, dass die genannten Zahlen unklar deklariert sind und oft nur auf kleinen Samples beruhen. Daher lassen diesen Zahlen keine Bewertung zu.

3.4.3 Effizienz der Algorithmen

Wenige Ersteller von Deduplizierungs-Algorithmen publizieren ihre Effizienz. Ausserdem erschweren unterschiedliche Dokumenttypen, Definitionen von

[33] Wo nicht anders vermerkt: Zahlen aus Machbarkeitsstudie Swissbib (2007).

[34] Zahlen aus Swissbib, swissbib.ch/HelpPage/about_swissbib

[35] Zahlen 2017, unisg.ch/de/universitaet/bibliothek/ueberuns/zahlenfakten

Dubletten, Datenkonsistenzen und Ziele der einzelnen Algorithmen einen Vergleich. Hickey und Rypka haben ihren OCLC-Algorithmus ausführlich auf Effizienz und Effektivität getestet (*Kapitel 3.3.2*). Grundsätzlich ist die Effizienz für das IFF-Verfahren nicht von grosser Bedeutung, daher wird diesem Punkt bei der Auswahl weniger Beachtung geschenkt.

3.4.4 Bewertung in Bezug auf Kriterienkatalog

Für das IFF-Verfahren (siehe *Kriterienkatalog, 3.2*) wird ein Deduplizierungsverfahren auf Titelebene gesucht, da es um das möglichst exakte Matching der IFF-Datensätze geht (*Designziel, 3.2.1*). Aus Gründen der Datenqualität ist erwünscht, für möglichst viele IFF-Datensätze einen Match zu finden, in der Annahme, dass im Endsystem vorhandene Datensätze von besserer Qualität sind. Bei der Umsetzung wird folglich die Abwägung zwischen Genauigkeit (Precision) und hoher Deduplizierung eine wichtige Rolle spielen. Hierbei sind die Erfahrungen aus dem OCLC-Verfahren und KOBV-Verfahren, welche gut dokumentiert sind, von enormer Wichtigkeit.

Da die IFF-Deduplizierung mittels Suche in einem Datenpool durchgeführt wird und kein Zugriff auf die Suchschlüssel des Datenpools besteht, ist die Search-Funktion des KOBV-Verfahrens eine gute Vorlage.

Aus Sicht des IDSSG, welcher im Jahr 2021 zu SLSP migriert, ist es naheliegend, die Deduplizierung von swissbib in Betracht zu ziehen. Die Instrumente, mit welchen swissbib arbeitet (Big-Data-Techniken), sind jedoch für den kleinen IFF-Bestand nicht angemessen (Aufwand und Ertrag, siehe *Kriterienkatalog, 3.1.7*).

Das lokale, bisherige Verfahren des IDSSG ist gut dokumentiert, auch sitzt der Autor des Verfahrens im Haus und somit können Erfahrungen und Verbesserungsvorschläge ausgetauscht werden. Das Verfahren erfüllt jedoch die Anforderungskriterien des IFF-Verfahrens nicht, da es auf manueller Kontrolle durch Experten basiert (*Kriterienkatalog, 3.1.7*).

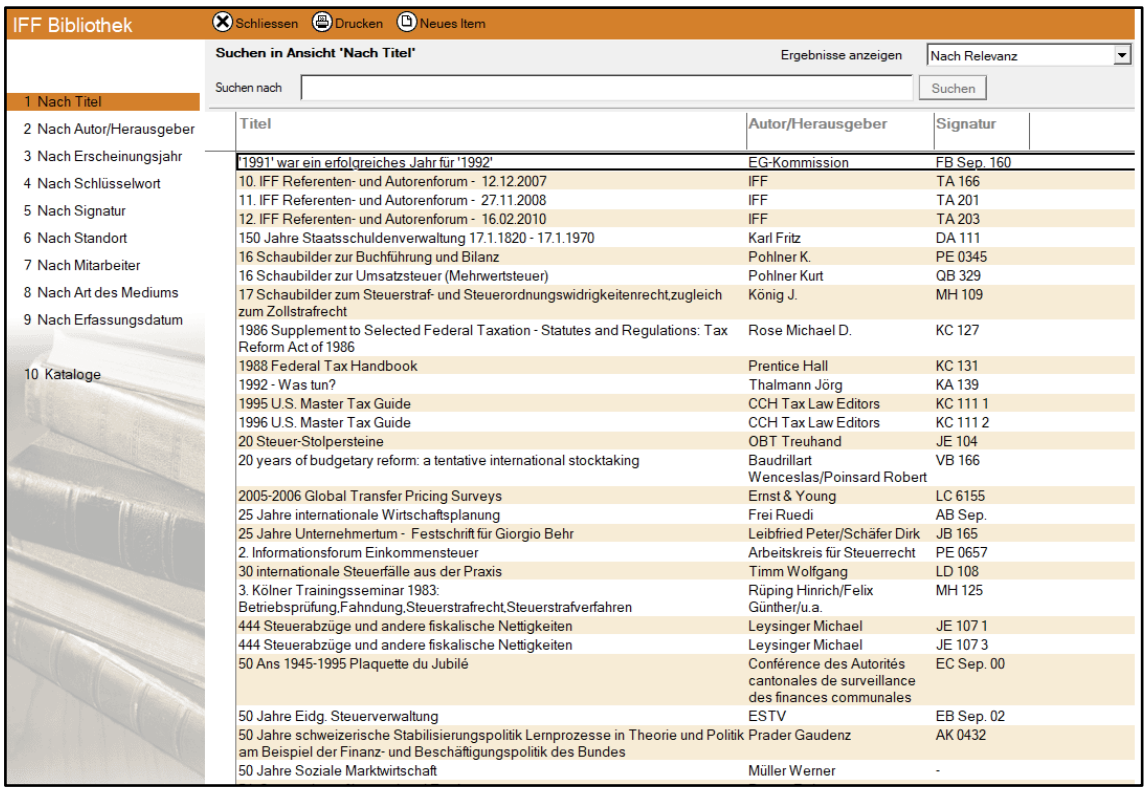
Grundsätzlich kann keines der bestehenden Verfahren 1:1 übernommen werden. Aus den meisten Verfahren konnten jedoch wichtige Erkenntnisse gewonnen werden. Die Haupt-Inspiration für das eigene Verfahren kommt aus den Verfahren von OCLC und KOBV.

4 Datenanalyse und Datenbereinigung der IFF-Daten

Dieses Kapitel beschreibt die detaillierte Analyse der Katalogdaten des IFF (nachfolgend IFF-Daten genannt) inklusive des Schema Mappings *(Kapitel 4.1.2)*.

Die IFF-Daten müssen vorbereitet werden, damit sie für ein Deduplizierungsverfahren verwendet werden können. Gruppen von Sonderfällen müssen identifiziert und speziell behandelt werden. Dies wird im Abschnitt Datenbereinigung (*Kapitel 4.2*) erläutert.

4.1 Datenanalyse



Titel	Autor/Herausgeber	Signatur
'1991' war ein erfolgreiches Jahr für '1992'	EG-Kommission	FB Sep. 160
10. IFF Referenten- und Autorenforum - 12.12.2007	IFF	TA 166
11. IFF Referenten- und Autorenforum - 27.11.2008	IFF	TA 201
12. IFF Referenten- und Autorenforum - 16.02.2010	IFF	TA 203
150 Jahre Staatsschuldenverwaltung 17.1.1820 - 17.1.1970	Karl Fritz	DA 111
16 Schaubilder zur Buchführung und Bilanz	Pohlner K.	PE 0345
16 Schaubilder zur Umsatzsteuer (Mehrwertsteuer)	Pohlner Kurt	QB 329
17 Schaubilder zum Steuerstraf- und Steuerordnungswidrigkeitenrecht,zugleich zum Zollstrafrecht	König J.	MH 109
1986 Supplement to Selected Federal Taxation - Statutes and Regulations: Tax Reform Act of 1986	Rose Michael D.	KC 127
1988 Federal Tax Handbook	Prentice Hall	KC 131
1992 - Was tun?	Thalmann Jörg	KA 139
1995 U.S. Master Tax Guide	CCH Tax Law Editors	KC 111 1
1996 U.S. Master Tax Guide	CCH Tax Law Editors	KC 111 2
20 Steuer-Stolpersteine	OBT Treuhand	JE 104
20 years of budgetary reform: a tentative international stocktaking	Baudrillart Wenceslas/Poinsard Robert	VB 166
2005-2006 Global Transfer Pricing Surveys	Ernst & Young	LC 6155
25 Jahre internationale Wirtschaftsplanung	Frei Ruedi	AB Sep.
25 Jahre Unternehmertum - Festschrift für Giorgio Behr	Leibfried Peter/Schäfer Dirk	JB 165
2. Informationsforum Einkommensteuer	Arbeitskreis für Steuerrecht	PE 0657
30 internationale Steuerfälle aus der Praxis	Timm Wolfgang	LD 108
3. Kölner Trainingsseminar 1983: Betriebsprüfung,Fahndung,Steuerstrafrecht,Steuerstrafverfahren	Rüping Hinrich/Felix Günther/u.a.	MH 125
444 Steuerabzüge und andere fiskalische Nettigkeiten	Leysinger Michael	JE 107 1
444 Steuerabzüge und andere fiskalische Nettigkeiten	Leysinger Michael	JE 107 3
50 Ans 1945-1995 Plaquette du Jubilé	Conférence des Autorités cantonales de surveillance des finances communales	EC Sep. 00
50 Jahre Eidg. Steuerverwaltung	ESTV	EB Sep. 02
50 Jahre schweizerische Stabilisierungspolitik Lernprozesse in Theorie und Politik am Beispiel der Finanz- und Beschäftigungspolitik des Bundes	Prader Gaudenz	AK 0432
50 Jahre Soziale Marktwirtschaft	Müller Werner	-

Abbildung 9: IFF-Katalog vor der Datenintegration

Das IFF hat seine bibliothekarischen Daten bis Anfang 2018 mit einer einfachen Katalog-Anwendung basierend auf Lotus Notes gepflegt. *Abbildung 9* zeigt eine Ansicht der Katalog-Anwendung. Der Datenexport aus diesem Katalog liegt im CSV-Format vor.

Nach eingehender Analyse der IFF-Daten ergeben die folgenden Merkmale der IFF-Daten besondere Herausforderungen:

- Für jedes einzelne Dokument in der IFF-Bibliothek wurde ein eigener Datensatz erfasst (ein Dokument entspricht einer Titelaufnahme mit Signatur), für einige Dokumente gibt es sogar doppelte Einträge in der Datenbank (z.B. Stücktitel in monografischer Reihe). Es gibt demnach viele IFF-Datensätze, die nach bibliothekarischer Katalogisierungs-Logik zu einem Titeldatensatz mit mehreren Exemplaren zusammengefasst werden müssen (z.B. Zeitschriften, mehrbändige Werke).
- Viele wichtige Informationen wurden in einem Textfeld «Zusatz» erfasst (z.B. übergeordneter Titel, Titel des Bandes, Quelle bei Zeitschriftenartikeln, aber auch Verwaltungsvermerke).
- Im Autorenfeld wurden alle Namen in einem Feld erfasst (bei mehreren Autoren getrennt durch Schrägstrich), Format: Nachname Vorname (getrennt durch Leerschlag. Es wurde nicht unterschieden zwischen Autor, Herausgeber oder Körperschaften.
- Wichtige Identifikationsdaten wie Auflage oder ISBN wurden nicht konsequent erfasst.
- Die Schlüsselwörter kommen aus einem eigenen Thesaurus des IFF und sollen migriert werden. Der Datentyp Zeitschriften wird zudem über die Schlüsselwörter definiert, nicht über die Materialart, wie bei den anderen Datentypen.
- Es gibt nur Titel- und Exemplardaten, keine Ausleih- oder Erwerbungsdaten.
- Die IFF-Daten enthalten zahlreiche Eingabefehler. Beispiele dafür sind die inkonsistente Reihenfolge der Autorennamen (Vorname Nachname statt Nachname Vorname), Tipp- oder Flüchtigkeitsfehler im Titel (z.B. «Ursache» statt «Ursachen»), sowie Fehler in fremdsprachigen Titeln, insbesondere diakritische Zeichen.

Ein einzelner Beispieldatensatz sieht so aus (*Abbildung 10*):

Abbildung 10: Einzelner Datensatz im IFF-Katalog

4.1.1 Datentypen

Folgende Datentypen sind in den Rohdaten vorhanden:

- Monografien und mehrteilige Monografien
- Zeitschriften/Jahrbücher
- Schriftenreihen (monografische Reihen)
- Analytische Aufnahmen (Beitrag in Monografie, Jahrbuch oder Zeitschrift)
- Loseblattsammlungen
- CD-ROMs, DVDs und Online-Medien

Jeder Datentyp enthält spezifische Merkmale, die nachfolgend erläutert werden. In *Kapitel 4.1.2* wird ein Mapping für die vom Verfahren berücksichtigten Datentypen mit allen benötigten MARC-Feldern erstellt.

4.1.1.1 Monografien

Die identifizierenden Daten für Monografien befinden sich in folgenden Feldern: Autor, Titel, ISBN, Jahr. Ein Beispiel ist in *Tabelle 13* abgebildet. Zusätzlich können Angaben aus Verlag, Verlagsort sowie Materialart hinzugezogen werden. Im Zusatz stehen manchmal Titelzusätze, manchmal Auflagenbezeichnungen. Die Monografien können grösstenteils problemlos dedupliziert werden. Ausnahmen kommen auf die Blacklist, siehe *Kapitel 5.3.1*.

Autor	Bauer Helfried/Paleczny Alfred/Schulmeister Agnes
Titel	Aufgaben der Gemeinden
ISBN	3-7141-7823-6
Zusatz	Erfüllung und Ursachen der Aufgaben in österreichischen Gemeinden
Jahr	1977

Tabelle 13: Monografie (eigene Darstellung)

4.1.1.2 Mehrteilige Monografien

Hier gibt es zwei Varianten:

A) Der Bandtitel befindet sich im Zusatz. Beispiel siehe *Tabelle 14*

B) Es gibt einen Stücktitel im Titel oder Titelzusatz. Beispiel siehe *Tabelle 15*

In den IFF-Daten gibt es keine mehrstufigen Werke und somit keine Verlinkungen zwischen den Bänden und der Oberaufnahme. Der Stücktitel muss je nach Fall im MARC-Feld 550 oder 245 des Endsystems gesucht werden.

Einige schwierige Fälle (z.B. wenig aussagekräftige Stücktitel) kommen auf die Blacklist, siehe *Kapitel 5.3.1*.

Autor	Eppler Rudolf
Titel	Steuerüberwälzung - Bd. 1
ISBN	3 428 04562 9
Zusatz	1. Band: Eine modelltheoretische und empirische Analyse ausgewählter Probleme der Überwälzung der Gewinnsteuern

Tabelle 14: Mehrteilige Monografie, Fall A (eigene Darstellung)

Autor	Parczyk Wolfgang
Titel	Internationales Steuer-Lexikon - Bd. 1: Bundesrepublik Deutschland
ISBN	3 7220 3700 X
Zusatz	

Tabelle 15: Mehrteilige Monografie, Fall B (eigene Darstellung)

4.1.1.3 Stücktitel in monografischer Reihe

Für Stücktitel in monografischen Reihen hat das IFF einen speziellen Ansatz verfolgt und diese Dokumente doppelt erfasst: über den Stücktitel (Beispiel siehe *Tabelle 16*) sowie über die Reihen-Nummer (Beispiel siehe *Tabelle 17*). Dies wurde im IFF von einem Professor so gewünscht, die Einträge können jedoch im Verbundkatalog des IDSSG zusammengeführt werden. Da die Reihen bekannt sind, können die Reihen-Nummer-Aufnahmen leicht über die Blacklist (*Kapitel 5.3.1.*) herausgefiltert werden (z.B. Reihen IFSt-Schrift, IFSt-Brief, IFSt-Heft). Die Stücktitel können wie Monografien behandelt werden.

Autor	Beland Ulrike
Titel	Entwicklung der Realsteuerhebesätze der Gemeinden mit 50000 und mehr Einwohnern im Jahr 2008 gegenüber 2007
ISBN	-
Zusatz	IFSt-Schrift 452
Seiten	86
Jahr	-

Tabelle 16: Stücktitel in monografischer Reihe, Fall A (eigene Darstellung)

Autor	IFSt
Titel	IFSt-Schrift 452
ISBN	3-89737-141-2
Zusatz	Ulrike Beland, Entwicklung der Realsteuerhebesätze der Gemeinden mit 50 000 und mehr Einwohnern im Jahr 2008 gegenüber 2007
Seiten	86
Jahr	2008

Tabelle 17: Stücktitel in monografischer Reihe, Fall B (eigene Darstellung)

4.1.1.4 Analytische Aufnahmen

Analytische Aufnahmen, also unselbständige Beiträge in Zeitschriften, Jahrbüchern oder Monografien, können über das Feld «Zusatz» erkannt werden. Dort ist die Quelle vermerkt, beginnend mit «in: ». Ein weiterer Hinweis auf analytische Aufnahmen ist das Feld «Seiten», da hier ein Seitenbereich (von… bis…) angegeben wird. Ausserdem fehlt die Jahresangabe. Die analytischen Aufnahmen enthalten oft im Feld «ISBN» den Identifier der Quelle, daher muss für die Suche das Feld ISBN geleert werden, um keine falschen Treffer zu erhalten (Beispiel siehe *Tabelle 18*).

Da bei analytischen Aufnahmen oft weniger Daten vorhanden sind als bei Monografien, wurden gewisse Gewichtungen anders gesetzt, damit der vorgegebene Schwellenwert für ein gutes Matching trotz weniger Kriterien erreicht wird (Beispiele im *Anhang B*).

Autor	Heini Anton.
Titel	Wettbewerbsbeschränkungen auf dem EU-Markt vor schweizerischen Schiedsgerichten - zu einem Aufsatz von Roger Zäch
ISBN	3 7255 3887 5
Zusatz	In: Baldi, Marino, Baumann, Max & u.a., (Hrsg.): Der Einfluss des europäischen Rechts auf die Schweiz, Festschrift für Professor Roger Zäch zum 60. Geburtstag
Seiten	S. 317-324
Jahr	-

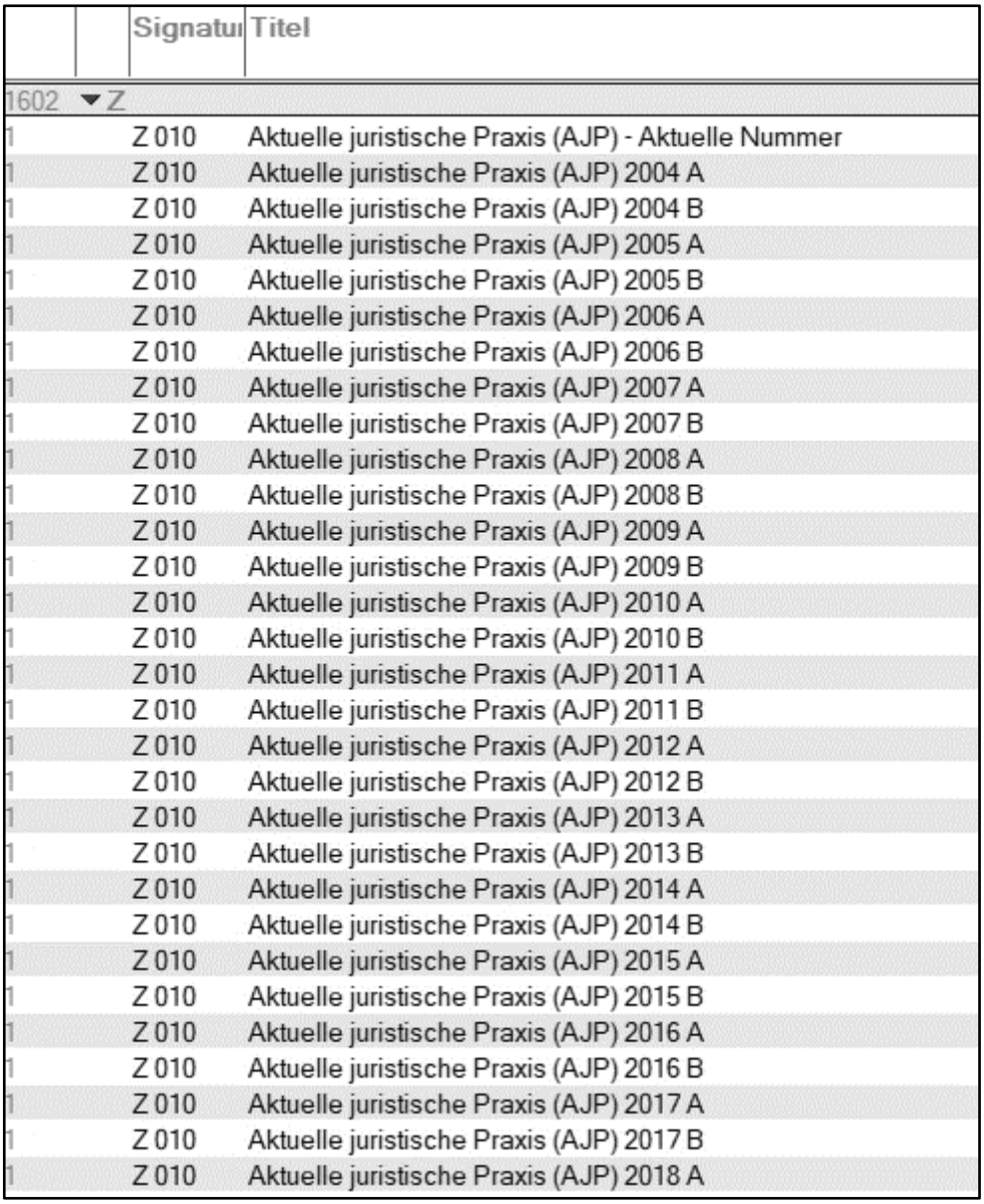
Tabelle 18: Analytische Aufnahme (eigene Darstellung)

4.1.1.5 Zeitschriften / Jahrbücher (Print und Online)

Dieser Datentyp ist über die Spalte «code1 = Z» oder über das Wort «Jahrbuch» bzw. «Yearbook» im Titel erkennbar. Bei den Zeitschriften gibt es einen oder mehrere Datensätze pro Jahr, sowie einen Dummy-Datensatz für die aktuelle Nummer. Ein Beispiel der Zeitschrift «Aktuelle Juristische Praxis» ist in *Abbildung 11* abgebildet.

Da Zeitschriften und Jahrbücher sehr schwierig abzugleichen sind und bei den Online-Zeitschriften zusätzlich noch die Zugriffsmöglichkeiten geprüft werden müssen, wurden diese manuell bereinigt. Die meisten Zeitschriftentitel sind im IDSSG vorhanden. Es handelt sich um ca. 30 laufende Zeitschriftentitel und insgesamt etwa 1600 Datensätze in den IFF-Daten. Die Anzahl der Jahrbücher können nicht genau beziffert werden.

Die Zeitschriften und Jahrbücher werden über eine Blacklist (siehe *Kapitel 5.3.1.*), die Materialart «Online» oder den Code «Z» von der Deduplizierung ausgenommen.



		Signatur	Titel
1602	▼Z		
1		Z 010	Aktuelle juristische Praxis (AJP) - Aktuelle Nummer
1		Z 010	Aktuelle juristische Praxis (AJP) 2004 A
1		Z 010	Aktuelle juristische Praxis (AJP) 2004 B
1		Z 010	Aktuelle juristische Praxis (AJP) 2005 A
1		Z 010	Aktuelle juristische Praxis (AJP) 2005 B
1		Z 010	Aktuelle juristische Praxis (AJP) 2006 A
1		Z 010	Aktuelle juristische Praxis (AJP) 2006 B
1		Z 010	Aktuelle juristische Praxis (AJP) 2007 A
1		Z 010	Aktuelle juristische Praxis (AJP) 2007 B
1		Z 010	Aktuelle juristische Praxis (AJP) 2008 A
1		Z 010	Aktuelle juristische Praxis (AJP) 2008 B
1		Z 010	Aktuelle juristische Praxis (AJP) 2009 A
1		Z 010	Aktuelle juristische Praxis (AJP) 2009 B
1		Z 010	Aktuelle juristische Praxis (AJP) 2010 A
1		Z 010	Aktuelle juristische Praxis (AJP) 2010 B
1		Z 010	Aktuelle juristische Praxis (AJP) 2011 A
1		Z 010	Aktuelle juristische Praxis (AJP) 2011 B
1		Z 010	Aktuelle juristische Praxis (AJP) 2012 A
1		Z 010	Aktuelle juristische Praxis (AJP) 2012 B
1		Z 010	Aktuelle juristische Praxis (AJP) 2013 A
1		Z 010	Aktuelle juristische Praxis (AJP) 2013 B
1		Z 010	Aktuelle juristische Praxis (AJP) 2014 A
1		Z 010	Aktuelle juristische Praxis (AJP) 2014 B
1		Z 010	Aktuelle juristische Praxis (AJP) 2015 A
1		Z 010	Aktuelle juristische Praxis (AJP) 2015 B
1		Z 010	Aktuelle juristische Praxis (AJP) 2016 A
1		Z 010	Aktuelle juristische Praxis (AJP) 2016 B
1		Z 010	Aktuelle juristische Praxis (AJP) 2017 A
1		Z 010	Aktuelle juristische Praxis (AJP) 2017 B
1		Z 010	Aktuelle juristische Praxis (AJP) 2018 A

Abbildung 11: Beispiel einer Zeitschrift im IFF-Katalog

4.1.1.6 Loseblattsammlungen

Das IFF hat i.d.R. bei Loseblattsammlungen das letzte Jahr erfasst, was bei der Suche und beim Matching zu Fehlern führt, da in den MARC-Daten des Endsystems der gesamte Erscheinungszeitraum erfasst ist. Es handelt sich um ca. 100 Datensätze. Ein Beispiel ist in *Tabelle 19* abgebildet.

Erkennbar sind sie in der Materialart Loseblattwerk. Einige schwierige Titel werden über eine Blacklist (siehe *Kapitel 5.3.1.*) von der Deduplizierung ausgenommen.

Autor	ESTV
Titel	Steuerentlastungen auf Grund von Doppelbesteuerungsabkommen - Bd. 1
ISBN	-
Materialart	Loseblattwerk
Jahr	2011

Tabelle 19: Loseblattwerk (eigene Darstellung)

4.1.1.7 CD-ROMs, DVDs (Audiovisuelle Medien)

Es kann aufgrund der Daten nicht unterschieden werden, um welchen Typ von audiovisuellem Medium es sich handelt. Da es nur wenige Dokumente sind (total 40 Datensätze), die Verwechslungsgefahr jedoch aufgrund fehlender Daten gross ist, werden diese anhand der Materialart von der Deduplizierung ausgenommen. *Abbildung 12* zeigt eine Übersicht der vorhandenen CD-ROMs und DVDs.

Duden. Die deutsche Rechtschreibung	Duden-Redaktion	WD 901 1
Duden. Die deutsche Rechtschreibung auf CD-ROM	Duden-Redaktion	WD 901 6
Luzerner Steuerbuch + Steuerentscheide des Kantons Luzern (CD-ROM)	Steuerverwaltung des Kantons Luzern	HB 203 1.1
Luzerner Steuerbuch + Steuerentscheide des Kantons Luzern (Server)	Steuerverwaltung des Kantons Luzern	HB 203 1.2
Schweizer Handbuch der Wirtschaftsprüfung 1998 (CD-ROM)	Treuhand-Kammer	WA 298 1.4.2
Schweizer Handbuch der Wirtschaftsprüfung 1998 (CD-ROM)	Treuhand-Kammer	WA 298 1.4.1
Schweizer Handbuch der Wirtschaftsprüfung 1998 (Server)	Treuhand-Kammer	WA 298 1.5
Schweizer Handbuch der Wirtschaftsprüfung 2009 (CD-ROM)	Treuhand-Kammer	WA 298 2.5.2
Schweizer Handbuch der Wirtschaftsprüfung 2009 (CD-ROM)	Treuhand-Kammer	WA 298 2.5.1
Schweizer Handbuch der Wirtschaftsprüfung 2009 (Server)	Treuhand-Kammer	WA 298 2.6
Statistisches Jahrbuch der Schweiz 1998 - CD-ROM	Bundesamt für Statistik	YG 012 2
Statistisches Jahrbuch der Schweiz 1999 - CD-ROM	Bundesamt für Statistik	YG 012 2
Statistisches Jahrbuch der Schweiz 2000 - CD-ROM	Bundesamt für Statistik	YG 012 2
Statistisches Jahrbuch der Schweiz 2002 - CD-ROM	Bundesamt für Statistik	YG 012 2
Statistisches Jahrbuch der Schweiz 2003 - CD-ROM	Bundesamt für Statistik	YG 012 2
Statistisches Jahrbuch der Schweiz 2004 - CD-ROM	Bundesamt für Statistik	YG 012 2
Statistisches Jahrbuch der Schweiz 2005 - DVD	Bundesamt für Statistik	YG 012 2
Statistisches Jahrbuch der Schweiz 2006 - DVD	Bundesamt für Statistik	YG 012 2
Statistisches Jahrbuch der Schweiz 2007 - DVD	Bundesamt für Statistik	YG 012 2
Statistisches Jahrbuch der Schweiz 2008 - CD-ROM	Bundesamt für Statistik	YG 012 2
Statistisches Jahrbuch der Schweiz 2009 - CD-ROM	Bundesamt für Statistik	YG 012 2
Statistisches Jahrbuch der Schweiz 2010 - CD-ROM	Bundesamt für Statistik	YG 012 2
Statistisches Jahrbuch der Schweiz 2011 - CD-ROM	Bundesamt für Statistik	YG 012 2
Statistisches Jahrbuch der Schweiz 2012 - CD-ROM	Bundesamt für Statistik	YG 012 2
Statistisches Jahrbuch der Schweiz 2013 - CD-ROM	Bundesamt für Statistik	YG 012 2
Sternstunde Philosophie: Arm und Reich. Anthony Atkinson im Gespräch mit Nathalie Wappler	Schweizer Fernsehen	YD 017
Sternstunde Philosophie: Krise und Kritik. Nobelpreisträger Joseph Stiglitz im Gespräch mit Roger de Weck	Schweizer Fernsehen	YD 016

Abbildung 12: CD-ROMs und DVDs im IFF-Katalog

4.1.2 Schema Mapping der IFF-Daten

Aus der Analyse der Datentypen wurde für die Datentypen, welche dedupliziert werden sollen, ein Mapping der IFF-Daten mit den möglichen MARC-Feldern aus einem Datenpool wie swissbib oder GVI erstellt (*Abbildung 13*). Die Felder der IFF-Daten müssen den entsprechenden MARC-Feldern des gewählten Datenpools zugeordnet werden.

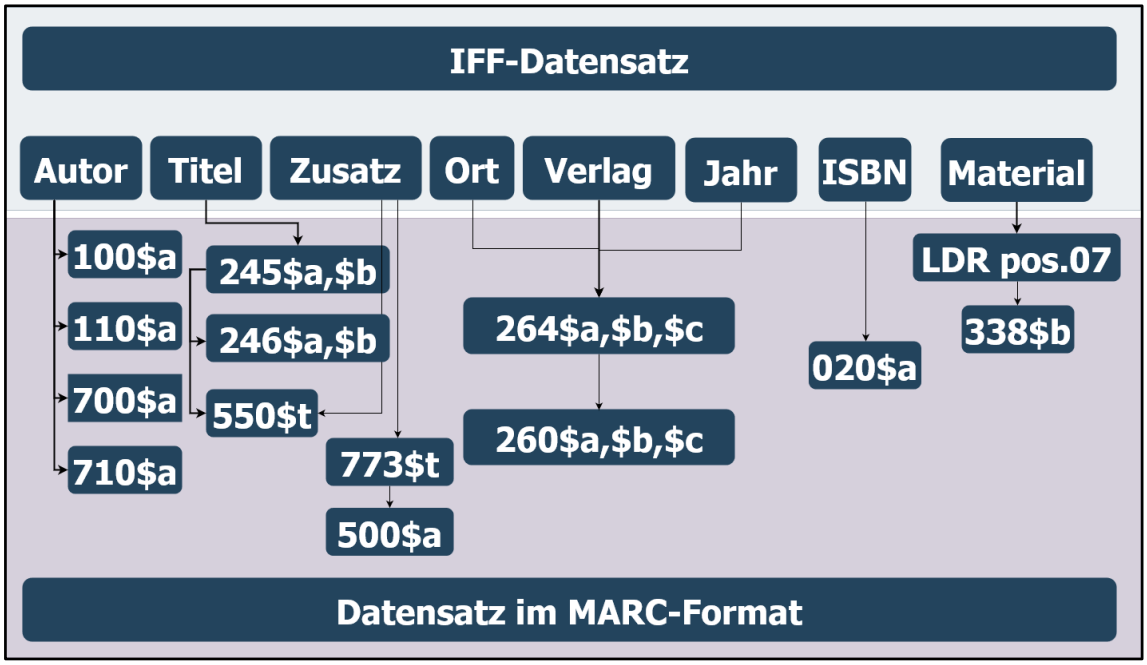


Abbildung 13: Schema Mapping IFF-Daten (eigene Darstellung)

Einige Datentypen werden manuell bereinigt und von der Deduplizierung ausgeschlossen. Der IDSSG-Verbund beschloss, die Zeitschriften manuell durch die Zeitschriftenabteilung zu bereinigen (*Kapitel 4.1.1.5)*.

Zusätzlich zeigte sich bei den Tests, dass aufgrund fehlender Daten (bzw. mangelhafter Datenqualität) das Matching von Jahrbüchern, monografischen Reihen, Online-Dokumenten, audiovisuellen Medien sowie einigen juristischen, mehrstufigen Werken sehr schwierig und fehleranfällig ist. Die manuell zu bereinigenden Datensätze beziffern sich auf etwa 3500 Stück, was ca. 23% aller Datensätze entspricht (genaue Zahlen siehe *Kapitel 5.4.3*).

Diese Datensätze sind im Schema Mapping nicht abgebildet, da sie von der automatischen Deduplizierung ausgenommen werden.

4.2 Datenbereinigung der IFF-Daten

Damit für möglichst viele IFF-Dokumente ein passender Datensatz gefunden werden kann, müssen die Rohdaten bereinigt werden (Normalisierung, Standardisierung, siehe *Kapitel 2.2)*.

Da es sich hier um statische Daten handelt (neue Daten werden nach der Migration vom IDSSG katalogisiert), kann die Datenbereinigung im Hinblick auf ein einfacheres Matching mit MARC-Daten gestaltet werden. Es muss keine Rücksicht auf die Original-Daten genommen werden. Mögliche Bereinigungsarten sind *(Hildebrand, Gebauer, Hinrichs & Mielke, 2015, S. 113)*:

- Korrekturen von Formatierungen (z.B. ISBN)
- Standardisierung oder Normierung von Daten (z.B. Autorennamen)
- Strukturierung von Daten (z.B. Titel aufteilen in mehrere Felder)

Es gibt verschiedene Tools zur Datenbereinigung. Ein Beispiel für eine leicht zu handhabende Software ist OpenRefine[36]. Einige Bereinigungsarbeiten konnten gut mit diesem Tool umgesetzt werden, andere wurden erst im Perl-Skript beim Einlesen der Daten vorgenommen.

Das Bereinigen der Daten lohnt sich, auch wenn dabei ein gewisser intellektueller (sprich: manueller) Einsatz notwendig ist. Der Unterschied des Matching-Verfahrens mit minimaler Normierung (nur reguläre Ausdrücke im Perl-Skript) zu einem Verfahren mit Normierung mittels OpenRefine ist im *Kapitel 5.4* beschrieben.

Die wichtigsten Datenbereinigungen werden nachfolgend beschrieben.

4.2.1 Herausfiltern von gleichlautenden Titeln

Ein eindeutiger Titel ist ein wichtiges Kriterium für ein erfolgreiches Matching, da das Feld in jedem Datensatz vorkommt. Je generischer der Titel, je unsicherer wird das Zusammenführen von Datensätzen. Mit dem Clustern nach Titeln können häufig vorkommende Titel festgestellt und separat behandelt werden. Viele dieser Titel sind Zeitschriften oder zeitschriftenähnliche Dokumentarten (Monografische Reihen, mehrbändige, unselbständige Werke), z.B. die «Cahiers de droit fiscal international». Beim Testen hat sich herausgestellt, dass diese oft falsch zusammengeführt werden oder den Schwellenwert für ein sicheres Matching nicht erreichen. Dasselbe liess sich für juristische Werke mit sehr allgemei-

[36] OpenRefine: openrefine.org

nem Titel, aber komplizierter Teilband-Struktur feststellen (Beispiel «Umsatzsteuergesetz»). Daher werden diese Titel mithilfe einer Blacklist von der Deduplizierung ausgeschlossen. Die Clustering-Funktion von OpenRefine war hilfreich, um solche problematischen Titel zu finden (*Abbildung 14*).

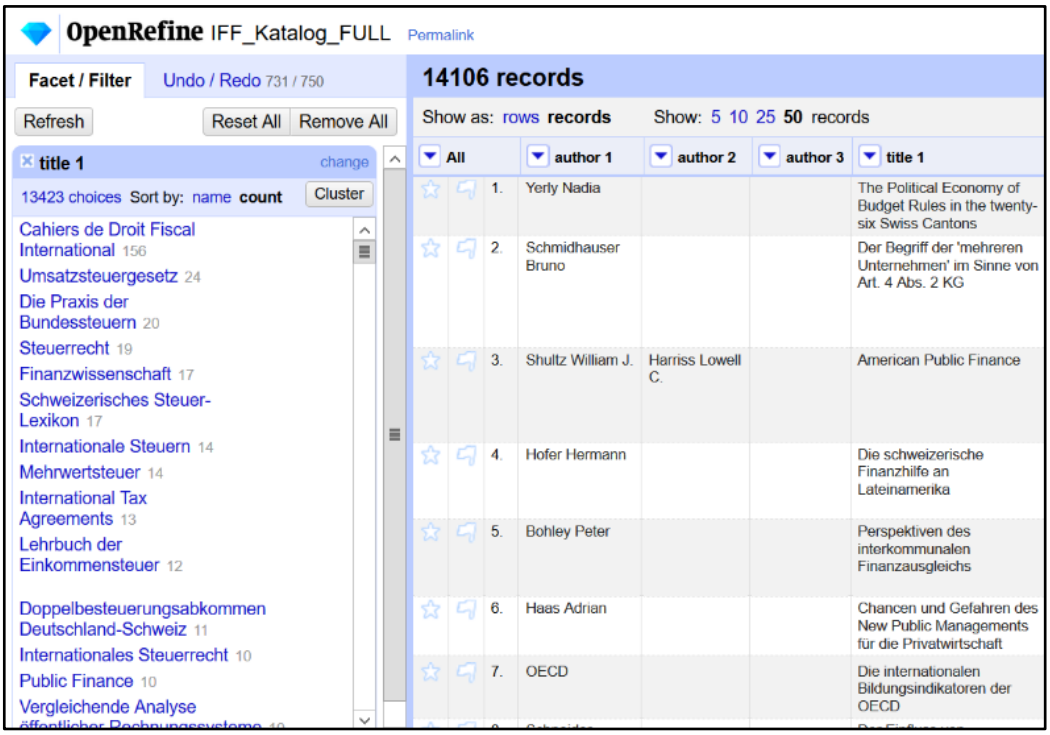
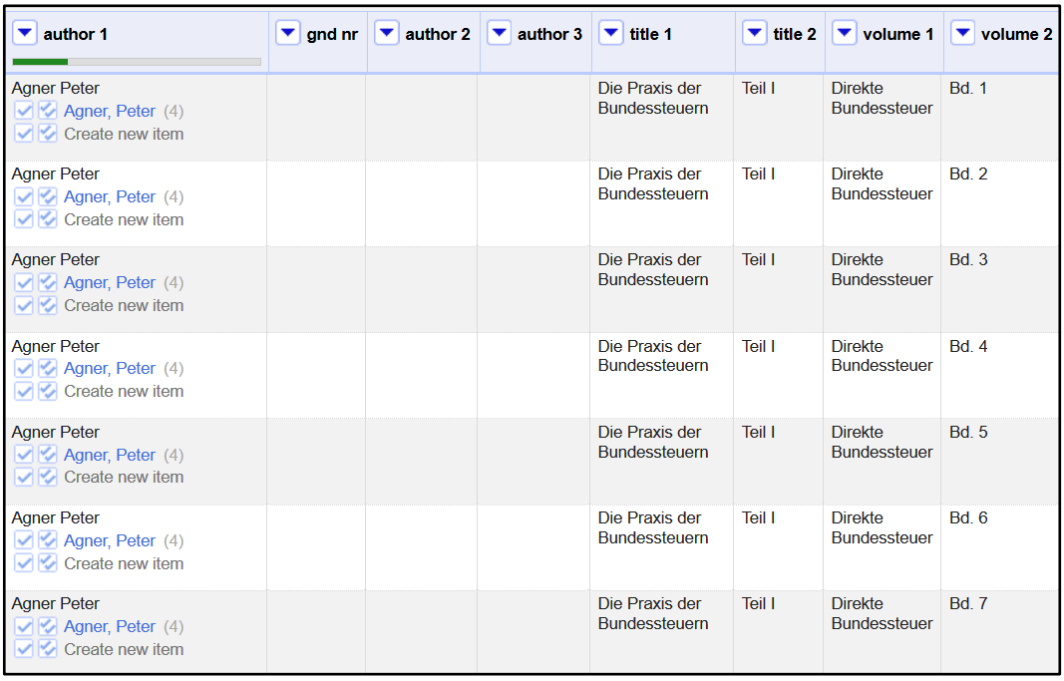


Abbildung 14: Clustern nach Titeln mit OpenRefine

4.2.2 Strukturierung von Titel- und Autorenspalten

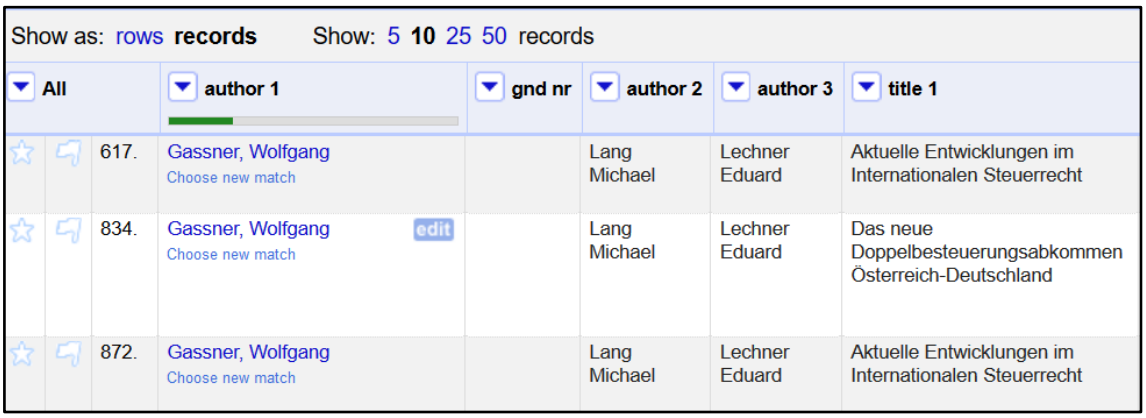
Des Weiteren konnten mit einfachen Befehlen in General Refine Expression Language (GREL)[37] bestimmte Daten in separate Spalten aufgeteilt werden. Dies wurde auf die Autoren sowie die Titel (Titel, Zusatz, Bandangaben) angewendet und ist in *Abbildung 15* sowie *Abbildung 16* ersichtlich. Dies erlaubte eine einfachere Handhabung der einzelnen Felder im Programm.

[37] General Refine Expression Language (GREL): https://github.com/OpenRefine/OpenRefine/wiki/General-Refine-Expression-Language



author 1	gnd nr	author 2	author 3	title 1	title 2	volume 1	volume 2
Agner Peter Agner, Peter (4) Create new item				Die Praxis der Bundessteuern	Teil I	Direkte Bundessteuer	Bd. 1
Agner Peter Agner, Peter (4) Create new item				Die Praxis der Bundessteuern	Teil I	Direkte Bundessteuer	Bd. 2
Agner Peter Agner, Peter (4) Create new item				Die Praxis der Bundessteuern	Teil I	Direkte Bundessteuer	Bd. 3
Agner Peter Agner, Peter (4) Create new item				Die Praxis der Bundessteuern	Teil I	Direkte Bundessteuer	Bd. 4
Agner Peter Agner, Peter (4) Create new item				Die Praxis der Bundessteuern	Teil I	Direkte Bundessteuer	Bd. 5
Agner Peter Agner, Peter (4) Create new item				Die Praxis der Bundessteuern	Teil I	Direkte Bundessteuer	Bd. 6
Agner Peter Agner, Peter (4) Create new item				Die Praxis der Bundessteuern	Teil I	Direkte Bundessteuer	Bd. 7

Abbildung 15: Aufsplitten des Titels mit OpenRefine



Show as: rows **records** Show: 5 **10** 25 50 records

All		author 1	gnd nr	author 2	author 3	title 1
617.		Gassner, Wolfgang Choose new match		Lang Michael	Lechner Eduard	Aktuelle Entwicklungen im Internationalen Steuerrecht
834.		Gassner, Wolfgang edit Choose new match		Lang Michael	Lechner Eduard	Das neue Doppelbesteuerungsabkommen Österreich-Deutschland
872.		Gassner, Wolfgang Choose new match		Lang Michael	Lechner Eduard	Aktuelle Entwicklungen im Internationalen Steuerrecht

Abbildung 16: Aufsplitten der Autoren mit OpenRefine

4.2.3 Standardisierung von Namen sowie Orten

Mithilfe der Clustering-Funktionen von OpenRefine können unterschiedliche Schreibweisen von Autoren oder Körperschaften vereinheitlicht werden. Dabei zeigt sich die unterschiedliche Eignung der Ähnlichkeitsmasse. Als ziemlich praktisch erwies sich die Levenshtein-Distance sowie das ngram-Verfahren (*Kapitel 2.4.3*) für das Clustern von Verlagen, Ortschaften und Autorennamen, wie in *Abbildung 17* sowie *Abbildung 18* zu sehen ist. Typische Eingabefehler wie «Suttgart» oder «Feiburg», aber auch unterschiedliche Schreibweisen von

«St.Gallen» (mit und ohne Leerschlag) lassen sich so mit wenig manuellem Aufwand vereinheitlichen.

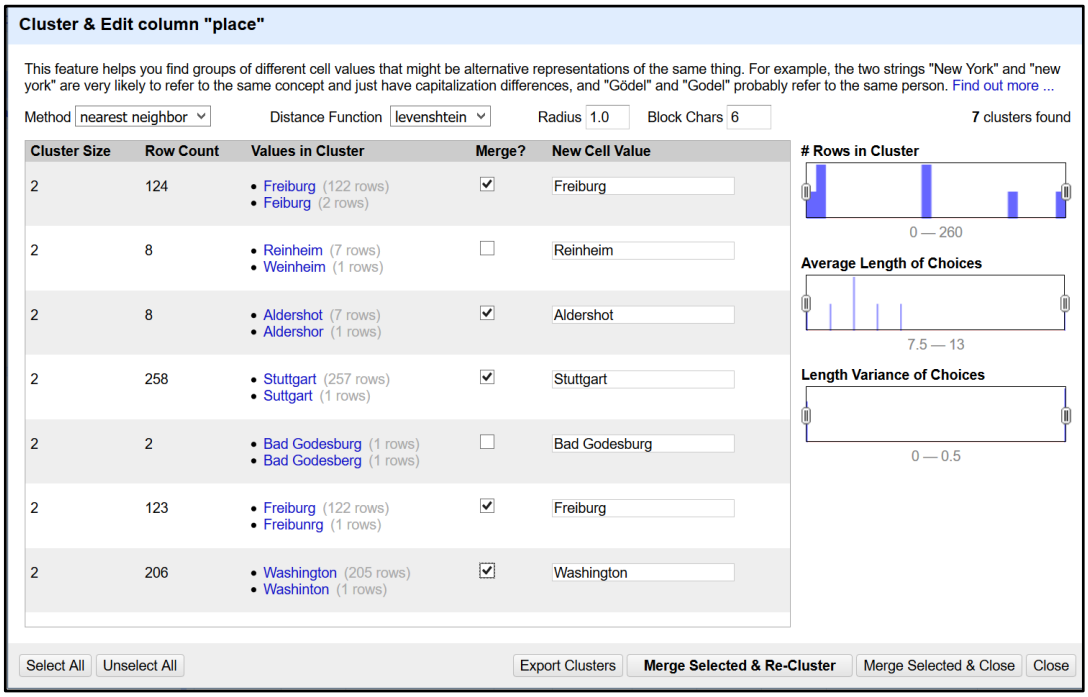


Abbildung 17: Levenshtein-Distance bei Orten mit OpenRefine

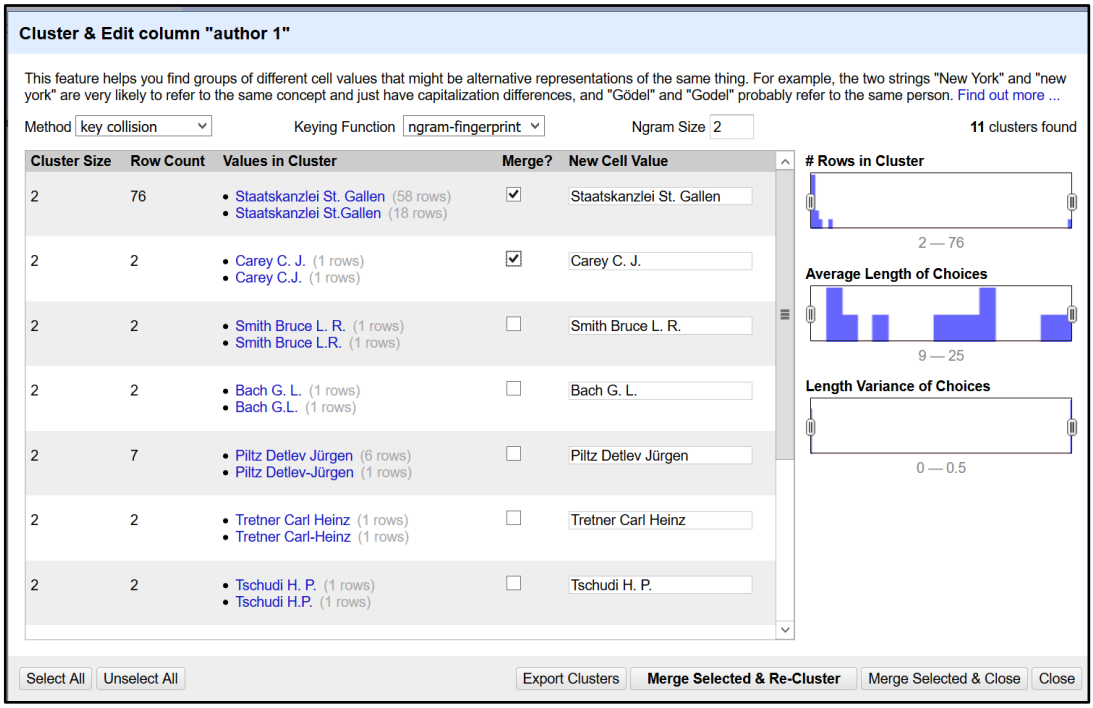


Abbildung 18: ngram-Ähnlichkeitsmass von Autoren mit OpenRefine

Im Gegensatz dazu zeigte sich, dass phonetische Verfahren – zumindest für den vorliegenden Datenbestand – wenig sinnvoll sind, wie in *Abbildung 19* (Beispiel: Autorennamen) gut ersichtlich wird.

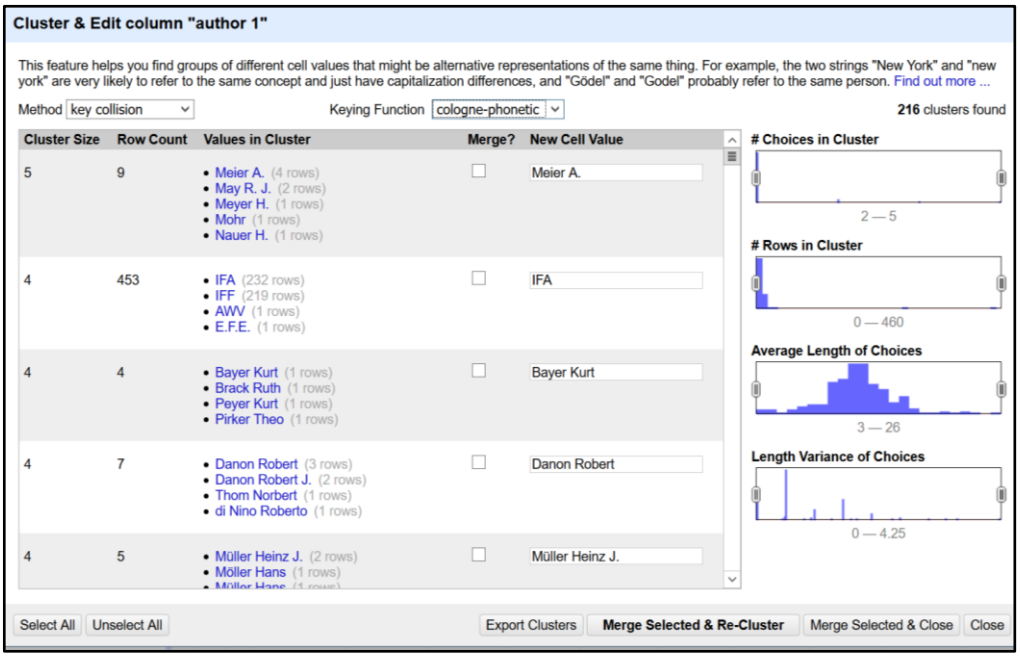


Abbildung 19: Phonetisches Ähnlichkeitsmass mit OpenRefine

4.2.4 Normierung nach GND

OpenRefine bietet die Möglichkeit einer «Reconciliation» mit der GND auf ganze Datenspalten *(Steeg & Pohl, 2018)*. Dies wurde versuchsweise mit der Spalte «Autor 1» getestet. Das Resultat sah sehr vielversprechend aus. Insbesondere für Körperschaften konnte so eine einheitliche Schreibweise erreicht werden (Beispiel «OECD» statt der französischen Form «OCDE»). Jedoch zeigte sich im Verlauf der Tests, dass dadurch mit der Abfrage in swissbib viele Titel nicht mehr gefunden wurden, bei denen nur das IFF-Original verfügbar war (*Kapitel 5.2.5*). Daher wurden einige dieser Normierungen wieder rückgängig gemacht. Beispiele dafür sind:

- IFF statt Institut für Finanzwissenschaft, Finanzrecht und Law and Economics
- IFA statt International Fiscal Association

Diese «Reconciliation» bietet eine gute Möglichkeit für eine Anreicherung der Daten, welche auch für Datensätze möglich wäre, wofür kein Ersatz auf swissbib

oder GVI gefunden wurde. Dadurch könnten zumindest die Personen, welche häufig vorkommen (z.B. Professoren und wissenschaftliche Mitarbeitende des IFF), mit einem GND-Normdatensatz verknüpft und somit besser auffindbar gemacht werden. Nach der manuellen Auswahl des korrekten GND-Datensatzes lässt sich die GND-Nummer mit GREL in eine separate Spalte extrahieren. Dies ist in *Abbildung 20* illustriert.

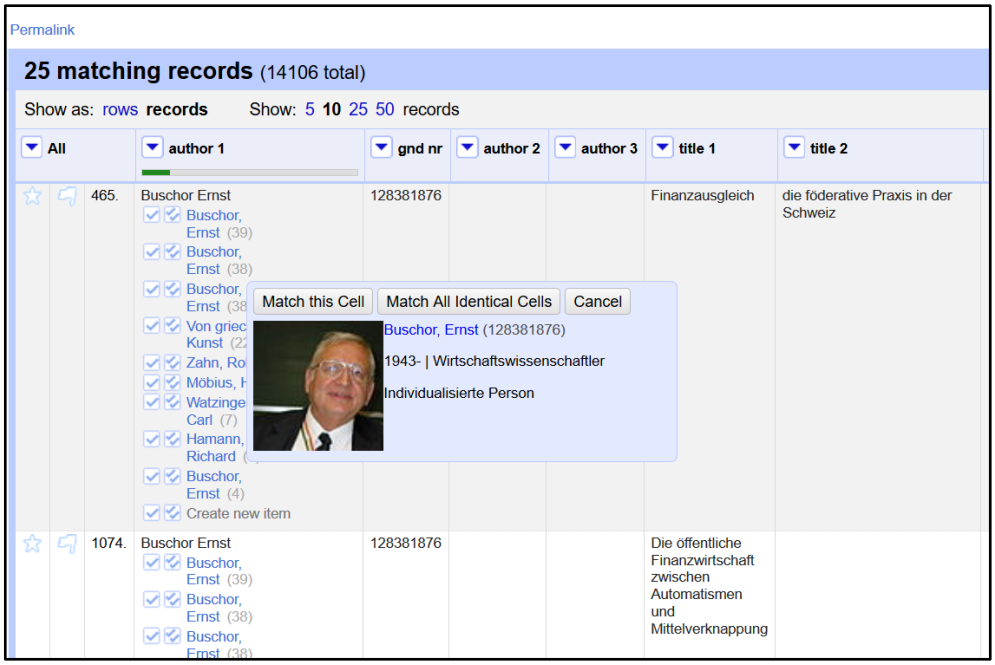


Abbildung 20: GND-Anreicherung eines IFF-Professors

Diese Anreicherung wurde in dieser Masterarbeit nicht umgesetzt und wäre ein Ausbau (*Kapitel 6.2.2*).

4.2.5 Normalisierung: Beispiel ISBN

Die ISBN ist ein Attribut, welches gut normalisiert werden kann. Eine mögliche Normalisierung ist in *Abbildung 21* ersichtlich. Nach dieser Normalisierung ist gewährleistet, dass die ISBN formal die korrekte Länge hat und somit für einen Abgleich oder eine Suche benutzt werden kann. Auf eine Gültigkeitsprüfung der ISBN wurde verzichtet. Dies wäre ein möglicher Ausbau (*Kapitel 6.2.1*).

```
398  sub normalize_isbn {
399      my $originalIsbn = shift;
400      my ( $n_isbn1, $n_isbn2, $flag_isbn1, $flag_isbn2 );
401
402      #   remove all but numbers and X
403      $originalIsbn =~ s/[^0-9xX]//g;
404      my $isbnlength = length($originalIsbn);
405
406      # check for valid numbers and set flags accordingly
407      if ( $isbnlength == 26 ) {
408
409          #there are two ISBN-13
410          $n_isbn2 = substr $originalIsbn, 13;
411          $n_isbn1 = substr $originalIsbn, 0, 13;
412          $flag_isbn1 = 1;
413          $flag_isbn2 = 1;
414      }
415      elsif ( $isbnlength == 20 ) {
416
417          #there are two ISBN-10
418          $n_isbn2 = substr $originalIsbn, 10;
419          $n_isbn1 = substr $originalIsbn, 0, 10;
420          $flag_isbn1 = 1;
421          $flag_isbn2 = 1;
422      }
423      elsif ( $isbnlength == 13 || $isbnlength == 10 ) {
424
425          #one valid ISBN
426          $n_isbn1    = $originalIsbn;
427          $n_isbn2    = undef;
428          $flag_isbn1 = 1;
429          $flag_isbn2 = 0;
430      }
431      else {
432          # not a valid isbn number
433          $flag_isbn1 = 0;
434          $flag_isbn2 = 0;
435          $n_isbn1    = undef;
436          $n_isbn2    = undef;
437      }
438
439      return ( $n_isbn1, $n_isbn2, $flag_isbn1, $flag_isbn2 );
440  }
441
```

Abbildung 21: Normalisierung der ISBN

4.2.6 Erkennung von Körperschaften

Ein weiteres Beispiel für die Datenbereinigung ist das Erkennen einer Körperschaft (*Abbildung 22*). In den Originaldaten des IFF stehen diese in derselben Spalte wie Autoren (es wird nicht unterschieden zwischen Person und Körperschaft). Die Variable @authorityWords enthält ausgewählte Begriffe wie «Amt», «Kanzlei», «Kanton», «Institut», etc.

```
458
459 sub normalize_author {
460     my $originalAuthor = shift;
461     my $authorflag;
462     my $authorsize;
463     my @authority;
464
465     my $authorityWords = $config->{regex}->{authority};
466     my $lastname;
467
468     if ( $originalAuthor =~ /\A\Z/ ) {
469
470         # author row is empty
471         $authorflag = 0;
472         $lastname   = undef;
473     }
474     else {
475         $authorflag = 1;
476
477         # check for authority
478         if ( $originalAuthor =~ /$authorityWords/i ) {
479             $authorsize = 5;    # is an authority
480         }
481         else {
482             @authority  = split( ' ', $originalAuthor );
483             $authorsize = scalar @authority;
484         }
485         if ( $authorsize <= 3 ) {
486
487             # probably a person, trim author's last name:
488             if ( $originalAuthor =~ /\Avon\s|\Ade\s|\Ale\s/i ) {
489                 $lastname = ( split /\s/, $originalAuthor, 3 )[1];
490             }
491             else {
492                 $lastname = ( split /\s/, $originalAuthor, 2 )[0];
493             }
494         }
495         else {
496             # keep the full name (is an authority)
497             $lastname = $originalAuthor;
498         }
499     }
500     $lastname = trim($lastname);    # remove whitespaces
501     return ( $lastname, $authorflag );
502 }
503
504
```

Abbildung 22: Prüfung von Körperschaften

Weitere Bereinigungen werden im Bereich Titel, Zusatz, Verlag, Verlagsort, Jahr, Seitenzahlen sowie Materialart vorgenommen. Es soll hier nicht auf jeden einzelnen Schritt eingegangen werden, diese können dem Programm im *Anhang C* entnommen werden.

5 Technische Umsetzung der IFF-Daten-Deduplizierung

Dieses Kapitel beschreibt die Datenintegration des IFF-Kataloges sowie die technische Umsetzung des Verfahrens, d.h. das Programm.

Im *Kapitel 5.1* und *5.2* werden die konkreten Voraussetzungen (Vorarbeiten sowie Annahmen zur Datenintegration) erläutert. Ein Matching-Verfahren soll der Situation entsprechend programmiert und angewendet werden, mit Nutzung der gewonnenen Erkenntnisse aus der Literaturstudie der bestehenden Verfahren. Dies wird im *Kapitel 5.3* beschrieben. Anschliessend werden die Ergebnisse des Verfahrens präsentiert und erläutert (*Kapitel 5.4*).

5.1 Beschreibung der Ausgangslage

5.1.1 Informationen zum IDSSG

Der IDSSG ist ein Bibliotheksverbund, bestehend aus wissenschaftlichen Bibliotheken, welche mit der Universität St.Gallen (HSG) assoziiert sind. Im IDSSG wird nach RDA katalogisiert, mit Normdaten aus der GND. Die Katalogdaten liegen im MARC21-Format vor, wie bei allen IDS-Verbünden.

5.1.2 Vorarbeiten des IDSSG

Im Frühjahr 2018 erhielt der IDSSG den Auftrag, die Daten des IFF-Katalogs zu migrieren. Wegen zeitlicher Vorgaben wurde dies zum Zeitpunkt (Mai 2018) mit einem einfachen Migrations- und Upload-Programm ausgeführt. Es fand nur eine minimale Deduplizierung basierend auf ISBN statt und in den meisten Fällen wurde aus den bestehenden IFF-Daten ein neuer Datensatz im MARC-Format in der Datenbank des IDSSG erstellt.

5.1.3 Konsequenzen für die Deduplizierung der IFF-Daten

Wie bereits in der Einleitung (*Kapitel 1*) sowie im Kriterienkatalog (*Kapitel 3.2.1*) beschrieben, soll das Verfahren für möglichst viele der IFF-Datensätze einen qualitativ besseren Datensatz finden. Die Datenqualität der Datenmigration von Mai 2018 soll verbessert werden, d.h. die damals eingespielten Dubletten im IDSSG sollen möglichst eliminiert werden. In den anderen Fällen (wo ein neuer Datensatz erstellt wurde) soll ein besserer Datensatz in einem grossen Datenpool gefunden werden.

Die Suche nach diesen Datensätzen erfolgt in einem grossen Datenpool mit standardisierten Daten (nach Regelwerk RDA, im MARC-Format). Die Suche in folgenden beiden Datenpools wurden im Programm implementiert:

- **swissbib**: Dies ist der grösste Datenpool der wissenschaftlichen Bibliotheken der Schweiz und enthält auch die Daten des IDSSG.
- **GVI**: Dieser Datenpool bietet mit seiner grossen Zahl an Dokumenten aus deutschsprachigen Bibliotheken eine noch grössere Datenquelle.

Beide Datenpools haben eine standardisierte Search/Retrieve via URL (SRU)-Schnittstelle[38], welche das automatische Abfragen mit Contextual Query Language (CQL)[39] Retrieval anbieten, und sind dokumentiert.[40]

Der Bestand des IDSSG ist ein Teilbestand von swissbib, nicht jedoch des GVI, was im Programm berücksichtigt werden muss (*Kapitel 5.2*).

Natürlich könnte das Programm ausgebaut werden auf weitere Datenpools mit einer standardisierten SRU-Schnittstelle und CQL Retrieval.

Am Ende des Programms sollen Datensätze im gewünschten Format (*siehe 3.2.6*) zum Einspielen in den IDSSG vorliegen.

Da für das Einspielen bzw. Umhängen der Daten sowie Erzeugen von Exemplar-Datensätzen bereits Instrumente im IDSSG vorliegen, ist dieser Schritt nicht Teil der Umsetzung.

---

[38] SRU protocol: loc.gov/standards/sru/sru-1-1.html

[39] Contextual Query Language (CQL): loc.gov/standards/sru/cql

[40] SRU-Schnittstellen-Dokumentation:
swissbib: swissbib.org/wiki/index.php?title=SRU, GVI: z3950.kobv.de/gvi

5.2 Annahmen zur Datenintegration

Das Verfahren unterscheidet grundsätzlich zwischen einer Suche im GVI oder einer Suche in swissbib. Da der IDSSG nicht im GVI enthalten ist, in swissbib jedoch schon, müssen bei einer Suche in swissbib zusätzliche Punkte berücksichtigt werden.

Vor der Umsetzung wurde von folgenden Annahmen ausgegangen:

- Bei einer Suche in einem Datenpool (egal welchen) gibt es einen, mehrere oder keine Treffer.
- Bei einem Treffer muss überprüft werden, ob der gefundene Treffer eine Dublette ist.
- Ergibt die Suche mehrere Treffer (= mehrere potenzielle Dubletten), so muss sowohl die Richtigkeit der Treffer überprüft als auch der beste Treffer gefunden werden.
- Ergibt die Suche keine oder zu viele Treffer, soll eine zweite, verbesserte Suche durchgeführt werden.

Zusätzlich gelten für swissbib folgende Annahmen:

- Aufgrund der Vorarbeiten sollten theoretisch für jeden IFF-Datensatz mindestens ein Treffer im Datenpool von swissbib gefunden werden, nämlich derjenige der Dateneinspielung der Originaldaten von Mai 2018
- Ausserdem wird für viele Datensätze bereits eine Dublette im IDSSG vorliegen, welche ebenfalls via swissbib identifiziert werden kann, nicht aber via GVI.

Im Folgenden wird kurz auf die einzelnen Fälle eingegangen. Einige Fälle gelten nur für die swissbib-Abfrage.

5.2.1 Nur Original-Dokument des IFF vorhanden

Gilt nur für die Abfrage in swissbib. Da die IFF-Daten bereits 2018 eingespielt wurden, muss sich theoretisch unter den Treffern in swissbib immer ein IFF-Dokument befinden.

Diese Datensätze können über MARC-Feld 035 (beginnend mit IDSSG) und den entsprechenden Nummern-Bereich (Nummern über 990‘000) identifiziert werden. Ist dies der einzige Treffer, kann davon ausgegangen werden, dass niemand sonst das Dokument besitzt, und am Datensatz, welcher im Mai 2018 erstellt wurde, nichts verbessert werden kann.

Im Verlauf der Umsetzung zeigte sich, dass in einigen Fällen in der Zwischenzeit (seit Mai 2018) von einer anderen Bibliothek Daten ergänzt wurden (erkenntlich an mehreren MARC-035-Feldern). In diesem Fall ist davon auszugehen, dass sich die Datenqualität verbessert hat und der Datensatz erneut importiert werden sollte.

5.2.2 Dublette im IDSSG vorhanden

Gilt nur für die Abfrage in swissbib. Alte Datensätze des IDSSGs können über MARC-Feld 035 (beginnend mit IDSSG) und den entsprechenden Nummern-Bereich (tiefer als 990‘000) identifiziert werden.

Kann eine Dublette eindeutig festgestellt werden, müssen die Exemplardaten und Stichworte (aus IFF-Thesaurus) des IFF im lokalen Verbund umgehängt und der IFF-Datensatz gelöscht werden. Es müssen nur die entsprechenden Datensatznummern des IDSSGs geliefert werden, da für das Umhängen von Exemplar-Datensätzen bereits Werkzeuge zur Verfügung stehen.

5.2.3 Dublette in einem anderen Datenpool vorhanden

Kann eine Dublette eindeutig festgestellt werden, wird der Datensatz aus swissbib oder GVI exportiert und der Datensatz des IFF von 2018 mit diesen Daten überschrieben (ausser Exemplardaten). Es muss ein kompletter XML-Datensatz erstellt und mit den Stichworten (aus IFF-Thesaurus) des IFF ergänzt werden. Dazu muss die Nummer des IFF-Datensatzes, der zu überschreiben ist, geliefert werden. Diese Nummer kann nur über swissbib eruiert werden, da die Daten im GVI nicht vorhanden sind.

5.2.4 Mehrere Dubletten: Entscheidungstabelle

Gibt es im Suchfenster mehrere Dubletten, so muss der beste Datensatz (Gewinner) gefunden werden. Dazu kommt einerseits das Matching der einzelnen Datenelemente zum Zug. Andererseits gibt es noch weitere Kriterien, wieso gewisse Dokumente bevorzugt werden, insbesondere die Herkunft der Daten. Dafür kommt eine Gewichtung der möglichen Fälle gemäss einer Entscheidungstabelle zum Zug.

Bei einer Abfrage im **GVI** ist die Entscheidungstabelle einfach: die Treffer werden in folgender Reihenfolge (absteigend) nach ihrer Herkunft bewertet: BSZ, BVB, GBV, DNB, KOBV, HBZ, HEBIS, OBV, andere Verbünde[41]. Diese Reihenfolge basiert auf einer Einschätzung der Qualität der Quellen durch einen Experten (*Stefan Lohrum, persönliche Kommunikation, 1.7.2019*).

Die Entscheidungstabelle ist bei **swissbib** etwas komplexer, da hier die Daten des IDSSG berücksichtigt werden müssen.

Verbundkürzel: (in MARC-Feld 035)

- IDSSG = IDSSG
    - alt: Systemnummern vor IFF-Migration
    - neu: Systemnummern ab IFF-Migration vom Mai 2018
    - HIFF: Bibliothekskennung des IFF im IDSSG
- IDSBB/IDSLU/NEBIS = IDS-Verbünde Basel-Bern, Luzern, Zürich-ETH[42]
- RERO = Westschweizer Bibliotheksverbund (Réseau Romand)[43]
- SGBN = St. Galler Bibliotheksnetz (Kantonsbibliothek u.a.)[44]

[41] Übersicht über die deutschen Verbünde: wikipedia.org/wiki/Bibliotheksverbund

[42] IDS-Verbünde: informationsverbund.ch/21.0.html

[43] RERO: Westschweizer Bibliotheksverbund, rero.ch

[44] SGBN: St. Galler Bibliotheksnetz, sgbn-primo.hosted.exlibrisgroup.com

Verbund / Output	IDSSG alt mit HIFF	IDSSG alt ohne HIFF	IDSSG neu	IDSBB/ IDSLU/ NEBIS	RERO	SGBN	OTHER
bestcase	X	-	-	-	-	-	-
replace	O	X	X	-	-	-	-
replace/reimport	O	O	X	X	-	-	-
replace/reimport	O	O	X	O	X	-	-
replace/reimport	O	O	X	O	O	X	-
replace/reimport	O	O	X	O	O	O	X
iffonly	O	O	X	O	O	O	O

Tabelle 20: swissbib-Entscheidungstabelle bei mehreren Dubletten (eigene Darstellung)

Legende zur swissbib-Tabelle: X = trifft zu / O = trifft nicht zu / - = nicht relevant

Gibt es einen bestehenden (alten) Treffer aus dem IDSSG, so soll dieser unbedingt bevorzugt werden, um in der eigenen Datenbank keine Dubletten zu erzeugen. Daher erhalten diese Dokumente viele Zusatzpunkte im Matching.

Ist das neu eingespielte IFF-Dokument bereits an ein altes IDSSG-Dokument angehängt, so muss dies als Best-Case-Szenario erkannt werden. Dies wird ebenfalls über die Matching-Werte sichergestellt.

Befindet sich unter den Treffern kein IDSSG-Dokument, so sollen unter den anderen Treffern gewisse Bibliotheken bevorzugt werden. Bei Vorkommen werden folgende Bibliotheksverbünde in untenstehender Reihenfolge bevorzugt, indem ihnen zusätzliche Punkte vergeben werden:

- Andere IDS-Bibliotheken: Seit 1998 gemeinsame Katalogisierungsregeln (bis 2015 KIDS[45], ab 2016 RDA) und ab 2021 gemeinsamer Katalog (SLSP).
- RERO-Bibliotheken: Ab 2021 gemeinsamer Katalog (SLSP), teilweise RDA-Daten vorhanden.

[45] KIDS: Katalogisierungsregeln IDS, siehe informationsverbund.ch/27.0.html

- SGBN-Bibliotheken: Regionale Bevorzugung, RDA-Daten vorhanden, gemeinsamer Katalog ab ca. 2024 (SLSP).

5.2.5 Problemfälle

Im Rahmen der Umsetzung sind folgende Problemfälle aufgetreten:

A) Das Original-IFF-Dokument wurde nicht gefunden

B) Die SRU-Abfrage ergab keinen Treffer

C) Die SRU-Abfrage ergab zu viele Treffer

Fall A) betrifft nur die Suche in swissbib. Fall A) und B) dürfen bei der swissbib-Suche theoretisch nicht vorkommen, da immer mindestens 1 Treffer (nämlich der bereits eingespielte) gefunden werden sollte. Praktisch trifft dieser Fall leider bei unter 2% der Abfragen ein. Aufgrund der geringen Menge an Fällen und keines ersichtlichen Musters muss Fall A) manuell bereinigt werden.

Bei Fall B) oder C) wird eine zweite, verbesserte Suche abgeschickt. Wenn das Resultat danach immer noch B) oder C) ist, wird dieser Datensatz in der Ausgabedatei mit Vermerk «notfound» gekennzeichnet und muss manuell bereinigt werden. In *Kapitel 5.4.3* (swissbib) bzw. *Kapitel 5.4.4* (GVI) sind die Zahlen aufgeführt, bei wie vielen Fällen diese Probleme auftreten.

Für Fall C) kann in der Konfigurationsdatei festgelegt werden, wie viele Treffer maximal gesucht werden sollen (Fenstergrösse). Grundsätzlich könnte man diesen Schwellenwert sehr hoch setzen. Darunter leidet höchstens die Performanz, was jedoch bei diesem Verfahren nicht von grosser Relevanz ist, da es nur einmal und «offline» durchgeführt wird (*Kapitel 3.1.6*). Allerdings hat sich gezeigt, dass bei sehr grossen Treffermengen meist kein gutes Resultat gefunden wird oder gar die Gefahr eines falschen Matchings besteht, da die Daten für die Suche in diesen Fällen zu generisch sind. Daher ist eine Fenstergrösse mit mehr als 20 Datensätzen nicht sinnvoll. Solche generischen Titel landen auf der Blacklist.

5.2.6 Reihenfolge der Abfrage in swissbib / GVI

Für die Datenintegration der IFF-Daten wird empfohlen, die Deduplizierung zuerst mit der swissbib-Schnittstelle durchzuführen, um die Datensätze der Originaleinspielung von Mai 2018 zu finden und so weitere Dubletten zu vermeiden. Das Programm wurde unter dieser Voraussetzung konzipiert.

Da eine Bereinigung der entstandenen Dubletten nur mit dem swissbib-Datenpool möglich ist (die Daten des IDSSG sind im GVI nicht integriert), macht die Abfrage im GVI nur Sinn für die verbliebenen Datensätze, welche mit swissbib nicht verbessert werden können. Nach einer ersten Abfrage in swissbib sind die Nummern der Datensätze, von denen nur das IFF-Original vorliegt, bekannt (Fall «iffonly» in Exportdatei). Es handelt sich um beinahe ein Viertel aller Daten (ca. 23%). Diese können nun nach demselben Prinzip im GVI gesucht und dedupliziert werden. Die Resultate dieses Vorgehens werden im *Kapitel 5.4.4* beschrieben.

Für zukünftige Datenintegrationen, welche noch nicht eingespielt wurden, spielt die Reihenfolge keine Rolle.

5.3 Entwicklung des IFF-Verfahrens

Hauptanspruch an das entwickelte Verfahren war es, einerseits in möglichst vielen Fällen auf Basis der Originaldaten einen guten vollständigen MARC-Datensatz zu finden und diesen in den IDSSG zu übernehmen (mit dem Risiko, dass z.B. Auflage oder Ausgabe nicht mit dem physisch vorhandenen Titel übereinstimmen). Andererseits sollte bei keinem oder keinem klaren Match die Originaldaten von Mai 2018 belassen werden. Bei der Umsetzung wurden die Prinzipien der Gestaltung eines solchen Deduplizierungsverfahrens (siehe *Kapitel 2* und *3.1*) beachtet, ein Kriterienkatalog erstellt (*Kapitel 3.2*) sowie die Erkenntnisse aus den untersuchten, bestehenden Verfahren (*Kapitel 3.4*) eingebracht.

5.3.1 Überblick

Bei der Umsetzung konnte in Bezug auf die Anforderungskriterien folgendes festgestellt werden:

Designziel-Ebene *(Kapitel 3.2.1.1*): Da identifizierende Angaben der Manifestation oder Expression wie Auflage oder Ausgabe in den IFF-Daten oft unvollständig oder ungenau erfasst wurden, ist dieser Vergleich nicht möglich, ohne das Dokument physisch zu überprüfen. Somit kann die Anforderung «Matching auf Manifestations-Ebene» nicht durchgehend erfüllt werden. Wo immer möglich, wird auf die Expressions-Ebene ausgewichen (*Kapitel 3.1.1*).

Designziel-Datentypen (*Kapitel 3.2.1.2*): Das Verfahren eignet sich nicht für Zeitschriften, Jahrbücher, audiovisuellen Medien und Online-Dokumente. Sie werden daher vom Verfahren ausgeschlossen. Eine gewisse Einschränkung gilt auch für mehrbändige Werke, insbesondere für Gesetzessammlungen oder juristische Kommentarwerke, oder sonstige, wenig aussagekräftige Titel. Sie werden ebenfalls mittels einer Blacklist vom Verfahren ausgeschlossen. Dies soll ein falsches Matching, insbesondere von unterschiedlichen Ausgabeformen und Hierarchien, verhindern.

Anwendungsschritte *(Kapitel 3.2.2):* Beim ersten Schritt (Suchfenster bestimmen) handelt es sich um ein «Search & Match»-Verfahren ähnlich dem des KOBV (*Kapitel 3.3.5*). Ein Sorted-Neighbourhood-Verfahren kann hier nicht angewendet werden, da kein Zugriff auf den gesamten Datenbestand von swissbib oder GVI bzw. dessen Indexe (Sortierschlüssel) besteht. Das Suchfenster wird für jedes Dokument mit einer SRU-Abfrage bestimmt. Die Fenstergrösse kann über eine Konfigurationsdatei bestimmt werden.

Auswahl der Felder *(Kapitel 3.2.3):* Zusätzlich zu den im Kriterienkatalog erwähnten Feldern werden Angaben aus dem Feld «Zusatz» ausgewertet, in welchem sich ein Sammelsurium von Informationen versteckt (z.B. Bandtitel, Quelle, Titelzusatz, etc.).

Es hat sich gezeigt, dass die Seitenzahlen der Originaldaten oft (nach bibliothekarischen Regeln) falsch erfasst wurden und daher zum Vergleich nicht geeignet

sind. Daher wird das Element «Seite» nur evaluiert, um festzustellen, ob es sich um eine Monografie oder ein Analyticum handelt.

Evaluation *(Kapitel 3.2.5):* Die Gewichtungswerte wurden basierend auf Werten von bestehenden Verfahren sowie eigenen Erfahrungswerten durch Analysen von Testdateien ermittelt.

Felder mit verlässlichen Werten in den IFF-Daten wie etwa dem Titelfeld oder der Materialart wurde eine hohe Gewichtung vergeben, um auch für Dokumente mit wenigen Angaben den Schwellenwert für ein sicheres Matching zu erreichen. Hingegen zeigte sich, dass für Felder, welche in den IFF-Daten nicht zuverlässig erfasst wurden, wie etwa das Jahr, auch leicht abweichenden Werten ein kleineres Gewicht vergeben werden musste, um nicht zu viele gute Datensätze auszuschliessen. Da die Werte für Verlag und Verlagsort bei den IFF-Daten oft mit einer gewissen Kreativität eingetragen wurden, konnte diesen Attributen kein hohes Gewicht zugeordnet werden. Die Gewichtungen für die Herkunft der Daten (Katalogisierungsquelle) wurden ausführlich getestet, um ein bestmögliches Matching zu erreichen (*Kapitel 5.2.4)*.

Bei den Vergleichen wurde kein Ähnlichkeitsmass verwendet, jedoch durch die «ANY»-Suche (CQL) sichergestellt, dass einige Eingabefehler in den Originaldaten aufgehoben wurden (Bsp. Einzahl/Mehrzahl-Fehler im Titel), da mehrere Indexe durchsucht wurden.[46]

Merging/Datenfusion *(Kapitel 3.2.6):* In einigen Fällen wird der bereits eingespielte Original-IFF-Datensatz von Mai 2018 aus swissbib neu importiert, da sich herausstellte, dass er mittlerweile von anderen Bibliotheken angereichert wurde.

In allen Fällen ersetzt der Gewinner-Datensatz den Verlierer-Datensatz, es werden nur die Stichworte des Verlierer-Datensatzes behalten. Bestehende Exemplardaten werden umgehängt.

Qualitätsmessung *(Kapitel 3.2.7):* Für die Qualitätsmessung wurden diverse Testsamples erstellt. Bei der Abfrage in swissbib war die Anzahl aller Dubletten bekannt, somit konnten die Werte für Recall und Precision berechnet werden. Bei

[46] Siehe sru.swissbib.ch/sru/explain?operation=explain welche Indexe mit dc.anywhere durchsucht werden.

der Abfrage im GVI konnten nur die swissbib-Daten als Vergleichswert hinzugezogen werden, da die Anzahl der möglichen Dubletten nicht bekannt war. Die Ergebnisse werden im *Kapitel 5.4* besprochen.

5.3.2 Ablaufbeschreibung

Abbildung 23 zeigt den Ablauf des Programms[47]. Eine grössere Darstellung des Ablaufdiagramms sowie das vollständige Programm befinden sich im *Anhang C.*

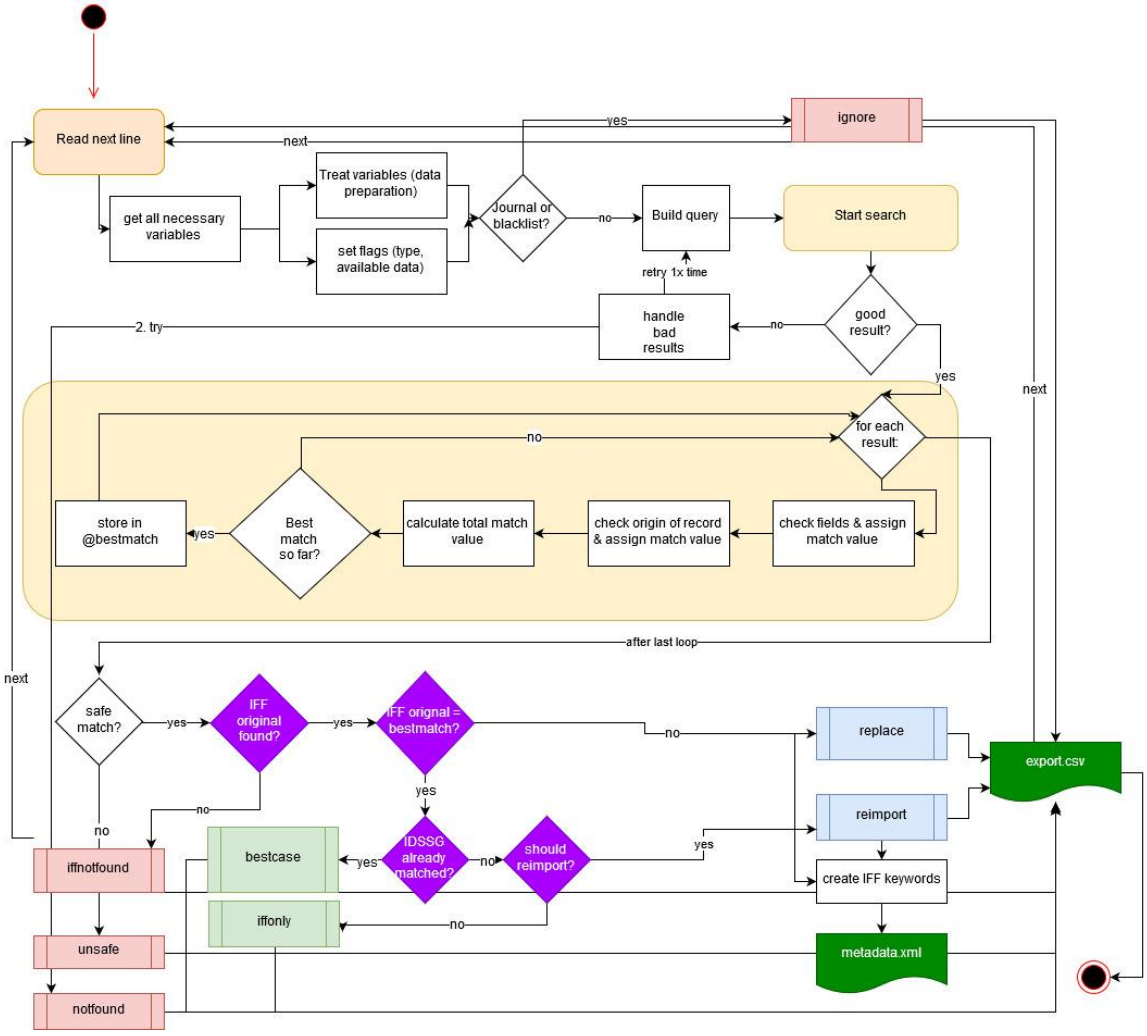


Abbildung 23: Ablauf des Programms dedup.pl (eigene Darstellung)

Über eine Konfigurationsdatei können bestimmte Werte für das Programm definiert werden, z.B. Fenstergrösse und Match-Werte. Für jede implementierte SRU-

[47] Bezeichnung des Perl-Skripts auf der DVD: v4_combined/dedup.pl

Schnittstelle wird eine eigene Konfigurationsdatei benötigt, da sich auch standardisierte Schnittstellen leicht unterscheiden. Die Konfigurationsdateien befinden sich auf der beiliegenden DVD (siehe *Anhang C*)., ein Auszug findet sich im *Anhang B*.

Das Programm wurde mit der swissbib- sowie GVI-Schnittstelle getestet und ist für beide Schnittstellen geeignet. Die gewünschte Schnittstelle wird über die Kommandozeile angegeben, zusammen mit den gewünschten Daten, welche dedupliziert werden sollen (CSV-Datei).

Das Programm liest zunächst die Argumente ein und setzt die entsprechenden Werte (Konfiguration der Schnittstelle). Dann werden die Daten der übergebenen CSV-Datei eingelesen. Die CSV-Datei muss einem bestimmten Muster entsprechen, d.h. das Programm sucht die Daten in bestimmten Spalten (z.B. Titel in Spalte 4, ISBN in Spalte 8). Im *Anhang A* wird der Aufbau der Datei genauer erläutert. Ein Auszug aus den Originaldaten findet sich ebenfalls im *Anhang A*.

Alle benötigten Spalten werden in einzelnen Variablen eingelesen. Nacheinander werden die Daten nun bereinigt und zu einem gewissen Grad normalisiert (*Kapitel 4.2*). Gleichzeitig werden für bestimmte Variablen Flags gesetzt (z.B. ob ein zweiter Autor oder ein Titelzusatz vorhanden ist). Aufgrund der Datenbereinigung können im nächsten Schritt nun die Zeitschriften, Jahrbücher sowie eine Liste schwieriger Titel (Blacklist) ausgeschlossen werden. Diese können in der Ausgabedatei herausgefiltert und manuell weiterbearbeitet werden.

Für alle andern Zeilen wird nun die SRU-Abfrage erzeugt. Ein erster Versuch wird immer aus ISBN oder Titel, Autor sowie – wenn vorhanden – Jahr gebildet.

Die Suchabfrage wird nun an den SRU-Server der gewählten Schnittstelle geschickt und es werden die Anzahl Treffer evaluiert. Liegen diese bei 0 oder über der konfigurierten Fenstergrösse, wird nochmals eine neue Suche zusammengebaut: Bei 0 Treffern wird eine breitere Suche (weniger Kriterien, allgemeinere CQL-Abfrage) gemacht, bei zu vielen Treffern wird eine engere Suche (mehr Kriterien) gemacht. Die erste oder zweite Suche führt in den meisten Fällen zu einer Trefferzahl innerhalb der gewünschten Fenstergrösse und kann weiterverarbei-

tet werden. Ansonsten wird dieser Datensatz in der Ausgabedatei als nicht gefunden markiert und kann manuell weiterverarbeitet oder in einem anderen Datenpool gesucht werden.

Bei einem Resultat zwischen 1 und Fenstergrösse wird jedes Dokument in der Trefferliste geprüft. Mithilfe der Flags und dem Schema Mapping werden die Felder aus den Originaldaten mit den entsprechenden Datenfeldern der gefundenen Dokumente verglichen. Es wird ein Matching-Wert zugeteilt, wenn diese übereinstimmen (*Kapitel 5.3.3*). Es wird dabei sehr grosszügig geprüft, ob die Strings aufeinanderpassen, jedoch werden keine Abweichungen toleriert (d.h. es werden keine Ähnlichkeitsmasse benutzt). Im Entstehungsprozess des Programms wurde noch mit einer erweiterten Subroutine gearbeitet, welche bei Misserfolg jeweils nur einen Teil des Strings vergleicht und die Hälfte der Punkte vergibt bei einem Teil-Match. Dies erwies sich jedoch nicht als zielführend. Bei Zahlen werden Abweichungen geprüft. Ein Beispiel für eine solche Evaluation findet sich im *Kapitel 5.3.3.*

Zusätzlich zu den Datenfeldern wird auch die Herkunft des Datensatzes überprüft und bewertet. Dies ist bei diesem Verfahren - insbesondere bei der swissbib-Suche - wichtig, da es eine klare Reihenfolge der Präferenzen gibt, unabhängig von der sonstigen Qualität des Datensatzes. Ein Datensatz aus dem IDSSG (d.h. eine lokale Dublette) soll immer höher gewertet werden als ein Datensatz aus einer anderen Bibliothek, auch wenn dieser vielleicht qualitativ bessere Daten enthält. Ziel des Verfahrens ist es ja, keine weiteren lokalen Dubletten zu verursachen. Die Höhe dieses Match-Wertes wurde durch ausführliches Testen ermittelt, damit er weder zu hoch (falsches Matching zu einem IDSSG-Datensatz) noch zu tief (ein anderer Datensatz aus swissbib wird höher gewichtet) ausfällt.

Die Match-Werte werden addiert. Gänzlich unsichere Matchings (d.h. Titelvergleich und Autorenvergleich ergeben 0 Punkte) werden aussortiert. Der Gesamtwert sowie der dazugehörige Datensatz werden gespeichert, wenn der bisher höchste Wert erreicht wurde. So befindet sich in der Liste @bestmatch immer der aktuell beste Treffer des Resultate-Fensters.

Sind alle Treffer-Dokumente evaluiert, wird geprüft, ob der Wert in @bestmatch einen Mindestwert erreicht und somit zuverlässig als Dublette gewertet werden

kann. Liegt der Wert darunter, konnte keine oder keine zuverlässige Dublette gefunden werden und der Vermerk «unsafe» wird in die Ausgabedatei geschrieben. Dieser Mindestwert wurde durch Vergleich aller @bestmatch-Werte der Testdaten ermittelt und liegt in dem Bereich, ab welchem sich die @bestmatch-Werte häufen, darunter kommen bedeutend weniger Matches vor und sind oft auch nicht mehr korrekt (*Abbildung 24*).

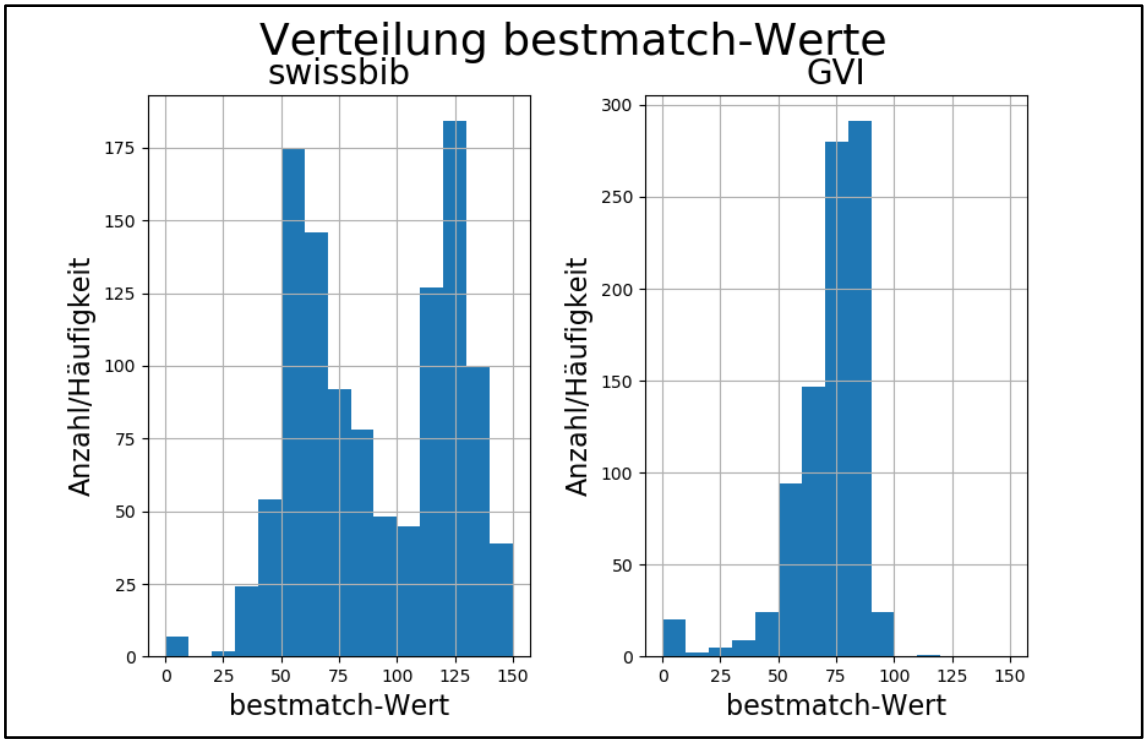


Abbildung 24: Verteilung der bestmatch-Werte

Der Mindestwert unterscheidet sich deutlich bei den beiden Schnittstellen. Die zweite (rechte) Spitze bei den swissbib-Daten lässt sich dadurch erklären, dass den Dubletten aus dem IDSSG sehr hohe Punkte vergeben werden, damit sie auf jeden Fall als Gewinner hervorgehen. Dadurch entstehen diese zweite Spitze und der deutliche Abfall dazwischen (ca. bei bestmatch-Wert 100).

Der gewählte Schwellenwert liegt für swissbib bei 31 Punkten, für den GVI bei 41, er jedoch kann in der Konfiguration geändert werden.

Bei der swissbib-Abfrage wird als nächstes geprüft, ob das Original-IFF-Dokument unter den Treffern war, denn diese Datensatz-Nummer wird benötigt, um die Einspielung von Mai 2018 zu ersetzen. Ist dies nicht der Fall, wird dies ebenfalls auf der Ausgabedatei vermerkt. Diese Fälle können nur manuell

weiterverarbeitet werden, auch wenn theoretisch ein passender Datensatz gefunden wird.

Danach wird gemäss der Entscheidungstabelle (*Kapitel 5.2.4*) vorgegangen. Wenn es eine Dublette zu bereinigen gibt, landet die Zeile inklusive der benötigten Datensatz-Nummern auf der Ausgabedatei.

Für die Suche im GVI gibt es nur die Variante «guter Treffer gefunden» oder «nicht gefunden». Die auf dem Ablaufdiagramm (*Abbildung 23)* lila markierten Entscheidungen fallen weg.

5.3.3 Evaluation

Die Treffer aus der SRU-Abfrage werden mit dem Originaldatensatz aus dem IFF verglichen. Dazu wird aus der XML-Datei das gewünschte MARC-(Unter)-Feld (gemäss Mapping) als String extrahiert und dem normalisierten String aus den Originaldaten gegenübergestellt. So werden die einzelnen, vorhandenen Datenfelder geprüft und es wird ihnen ein Matching-Wert zugeteilt, wenn diese übereinstimmen. Dies passiert mit der Subroutine getMatchValue (*Abbildung 25*).

Abbildung 26 sowie *Abbildung 27* zeigen als Beispiel den Vergleich Titel / Band (wenn zutreffend, d.h. wenn die Flags für die entsprechenden Felder gesetzt sind). Je nach Feld und Konfiguration werden unterschiedliche Gewichtungen gesetzt. Es werden grundsätzlich nur positive Gewichtungen verwendet, mit Ausnahme der negativen Gewichtung für die Herkunft, falls ein Dokument als Original-IFF-Dokument identifiziert wird. Dies hat sich als notwendig erwiesen, damit diese Dokumente nicht als Bestmatch identifiziert werden, wenn ein anderer guter Treffer vorhanden ist.

```
1015
1016  sub getMatchValue {
1017      my $df_content     = shift;
1018      my $sf_content     = shift;
1019      my $hash_ref       = shift;
1020      my $key            = shift;
1021      my $conf           = shift;
1022      my $record         = shift;
1023      my $xpath          = shift;
1024      my %norm               = %{$hash_ref};
1025      my $originalstring = $norm{$key};
1026          my $matchvalue     = $conf->{match}->{$key};
1027          my $datafield      = $conf->{sru}->{datafield};
1028      my $subfield       = $conf->{sru}->{subfield};
1029      my $CLEAN_TROUBLE_CHAR = qr/\.|\(|\)|\'|\"|\/|\+|\[|\]|\?|\,|/; #clean following characters: .()'"/+[]?,
1030
1031      if (defined $originalstring ) {
1032          # continue with comparison
1033          foreach my $el ( $xpath->findnodes($datafield.'[@tag="' . $df_content . '"]', $record  ) ) {
1034                          my $marcfield = $xpath->findnodes( $subfield.'[@code="' . $sf_content . '"]', $el )->to_literal;
1035                  $marcfield =~ s/$CLEAN_TROUBLE_CHAR//g;    # clean fields from special characters
1036                  if ( $marcfield =~ /\A\Z/ ) {
1037                      # subfield is empty and therefore does not exist:
1038                      return 0;
1039                  } else {
1040                          print $fh_report "$df_content $sf_content $marcfield\n";
1041                          $originalstring =~ s/$CLEAN_TROUBLE_CHAR//g;    # clean fields from special characters
1042                          if (( $originalstring =~ m/$marcfield/i ) || ( $marcfield =~ m/$originalstring/i ) ) {
1043                              #Marc data matches original data
1044                              return $matchvalue;
1045                          } else {
1046                                  #Marc data does not match original data
1047                                  return 0;
1048                          }
1049                  }
1050          }
1051      } else {
1052          # original string is not initialized, abort
1053          return 0;
1054      }
1055  }
```

Abbildung 25: Subroutine getMatchValue

```
# compare title & subtitle fields
my $t_match  = 0;
my $st_match = 0;
if ( hasTag( "245", $config, $rec, $xpc ) ) {
        $t_match = getMatchValue( "245", "a", \%norm, "tit", $config, $rec, $xpc );
        if ($flag{stit}) {
                $st_match = getMatchValue( "245", "b", \%norm, "stit", $config, $rec, $xpc );
        }
} elsif ( hasTag( "246", $config, $rec, $xpc ) ) {
        $t_match = getMatchValue( "246", "a", \%norm, "tit", $config, $rec, $xpc );
        if ($flag{stit}) {
                $st_match = getMatchValue( "246", "b", \%norm, "stit", $config, $rec, $xpc );
        }
}
```

Abbildung 26: Prüfung auf Titel, Untertitel und Alternativtitel

```
    # check volume titles:
my $tv_match = 0;
my $av_match = 0;
if ($flag{tvol}) {
    if ( hasTag( "505", $config, $rec, $xpc ) ) {
            $tv_match = getMatchValue( "505", "t", \%norm, "tvol1", $config, $rec, $xpc );
    } elsif ( hasTag( "245", $config, $rec, $xpc ) ) {
        $tv_match = getMatchValue( "245", "a", \%norm, "tvol1", $config, $rec, $xpc );
    }
}
if ($flag{avol}) {
    if ( hasTag( "505", $config, $rec, $xpc ) ) {
        $av_match = getMatchValue( "505", "t", \%norm, "avol_tit", $config, $rec, $xpc );
    } elsif ( hasTag( "245", $config, $rec, $xpc ) ) {
        $av_match = getMatchValue( "245", "a", \%norm, "avol_tit", $config, $rec, $xpc );
        if ( $av_match == 0 ) {
            $av_match = getMatchValue( "245", "b", \%norm, "avol_tit", $config, $rec, $xpc );
        }
    }
}
```

Abbildung 27: Prüfung auf Bandtitel

5.3.4 Datenfusion / Export

Die unterschiedlichen Fälle des Exports werden im Ablaufdiagramm (*Abbildung 23*) im unteren Drittel der Grafik gezeigt. Die lila markierten Entscheidungen betreffen nur die Abfrage in swissbib.

Wird ein besserer Datensatz als der Original-Datensatz des IFF gefunden, so müssen die Dokument-Nummern des zu ersetzenden IFF-Originals (aus swissbib) sowie die Dokument-Nummern des Ersatzes aus dem entsprechenden Datenpool sowie die zu ergänzenden Stichworte des IFF-Thesaurus geliefert werden. Dies geschieht über eine Export-Datei (in *Abbildung 23* unten rechts, dunkelgrün markiert) mit der Kennzeichnung der Titel mit dem entsprechenden Case sowie der dazugehörigen Datensatznummern. Die Exemplardaten des IFF-Katalogisats werden im Fall einer Dublette im IDSSG umgehängt und der IFF-Datensatz gelöscht. Kommt die Dublette aus einer anderen Quelle (swissbib oder GVI), so werden die Daten aus dem SRU-Resultat als XML-Datensatz exportiert. Auch hier müssen nur die Stichworte des IFF mitgeliefert werden (als Teil der XML-Daten). Diese ersetzen dann die Daten der Originaleinspielung von 2018, die Exemplardaten bleiben bestehen. Wird keine Verbesserung erzielt, so wird

der Datensatz der Einspielung von Mai 2018 belassen und es ist keine weitere Aktion notwendig.

Die Export-Daten, welche das Programm generiert, werden für den Empfänger (IDSSG) in der Readme-Datei auf GitHub dokumentiert. Die Readme-Datei befindet sich auf der DVD (siehe *Anhang C*). Die Datenfusion ist nicht Teil der technischen Umsetzung des IFF-Verfahrens, da dazu im IDSSG bereits Programme vorliegen. Im Juni 2019 wurden die IFF-Daten mit dem Programm (Skript 2, *siehe 5.4*) aus swissbib dedupliziert.

5.4 Ergebnisse

Es wurden vier Versionen des Programms erstellt (nachfolgend als Skript 1-4 bezeichnet):

- Skript 1: nicht normalisierte Daten, swissbib-Schnittstelle
- Skript 2: normalisierte Daten, swissbib-Schnittstelle
- Skript 3: nur GVI-Schnittstelle, nicht ausführlich getestet
- Skript 4 (finale Version): normalisierte Daten, beide Schnittstellen

Die meisten Tests wurden mit Skript 1 und 2 durchgeführt. Skript 4 ist für die swissbib-Schnittstelle eine refaktorisierte Version (Code-Restrukturierung), aber prozedural unverändert, die GVI-Schnittstelle ist neu hinzugekommen. Tests für die GVI-Schnittstelle mit Skript 4 wurden nachträglich hinzugefügt, als die Tests für Skript 1 und 2 bereits abgeschlossen und die IFF-Daten über swissbib dedupliziert waren.

5.4.1 Testdateien

Für die ersten Tests wurde mit kleinen, manuell ausgewählten Testdatensätzen gearbeitet. Durch das Variieren unterschiedlicher IFF-Datensätze konnten wichtige Hinweise auf Datenanomalien und Erkenntnisse zur Verbesserung des Programms gefunden werden. Beispiele solcher Testdateien sind im Unterordner data abgelegt (siehe *Anhang C*).

Danach wurde ein Testdatensatz von 1526 Dokumente erstellt, welcher jede 10. Zeile der Originaldaten und somit 10% der Gesamtdaten und einen guten Querschnitt durch alle Dokumenttypen enthält. Die Sortierung der Originaldaten ist willkürlich, es tauchen aber blockweise ähnliche Datensätze (z.B. Zeitschriften oder Reihen) auf, daher erwies sich diese 10%-Datei als gute Lösung, um die Funktionsfähigkeit des Programms für alle Datentypen zu testen. Die Datei heisst test1526.csv und befindet sich ebenfalls im «data» Ordner (siehe *Anhang C*). Wenn im Verlauf des Kapitels von Testdatensatz die Rede ist, so handelt es sich um diesen beschriebenen Datensatz mit 10% aller Daten.

5.4.2 Tests und Testergebnisse

Im Verlauf der Programmentwicklung gab es zwei Ansätze:

- Einfache Datenbereinigung im Programm mittels regulärer Ausdrücke und einiger Perl-Funktionen, ohne grosse Kenntnisse des Dateninhalts. Die Originaldaten wurden nicht verändert.
- Datenbereinigung mittels OpenRefine, wobei konkrete Änderungen an den Originaldaten vorgenommen wurden (*Kapitel 4.1*). Die Daten wurden jeweils nach Häufigkeit geclustert und bei allen zum Matching verwendeten Feldern die häufigsten Formen durchgesehen und normalisiert oder korrigiert. Danach wurde ein zweites Programm verfasst, welches nur noch minimale Datenbereinigungen enthielt.

Für einen Datenbestand in der vorliegenden Grösse ist das manuelle Bereinigen mithilfe von OpenRefine eine gute Möglichkeit, die Qualität deutlich zu verbessern. Ein Vergleich der abweichenden Ergebnisse von Skript 1 (Bereinigung im Skript) und Skript 2 (mit vorbereinigten Daten) ergab mit dem Testdatensatz (1526 Datensätze) folgende Resultate (*Tabelle 21*):

Starke Verbesserung	**34 Datensätze**
Leichte Verbesserung	**112 Datensätze**
Anderes Resultat (weder Verbesserung noch Verschlechterung)	**42 Datensätze**
Leichte Verschlechterung	**28 Datensätze**
Starke Verschlechterung	**6 Datensätze**

Tabelle 21: Unterschiede Skript 1 - Skript 2 (eigene Darstellung)

Unter einer starken Verbesserung versteht sich z.B. das Auffinden einer Dublette im Bestand des eigenen Verbunds oder swissbib, welche mit Skript 1 nicht entdeckt wurde. Unter einer leichten Verbesserung versteht sich z.B. das Auffinden des Originals, auch wenn sich herausstellte, dass dieses nicht verbessert werden konnte, oder ein sicherer Match im Vergleich zu einem unsicheren aus Skript 1. Bei einem anderen Resultat (ohne Verbesserung oder Verschlechterung) handelte es sich in der Regel um Fälle, wo eine Dublette einer anderen bevorzugt wurde, die Unterschiede jedoch auf die Datenqualität keinen Einfluss hatten. Unter einer leichten Verschlechterung versteht sich z.B. der Fall, dass eine Dublette nicht mehr erkannt wurde, oder eine Dublette aus swissbib einer Dublette des eigenen Verbunds bevorzugt wurde. Dies lag hauptsächlich daran, dass einige Dokumente sehr lange und komplizierte Titel hatten. Unter den starken Verschlechterungen befanden sich einige falschen Matchings, bei denen ein falsches Dokument als Dublette erkannt wurde und somit das Original ersetzt hätte. Hauptgrund war dabei der zu hohe, vergebene Wert für ein Dokument aus dem IDSSG. Dieser Fehler konnte mit einer Korrektur der Gewichtung für die IDSSG-Herkunft entfernt werden.

Zwei Beispiele für solche **falschen Matches**:

IFF-Original:

```
Shubik Martin, Game-Theory in the Social Sciences, Vol. 2,
0.262.19219.5, 1984
```

wäre ersetzt worden mit Dokument aus dem IDS-St.Gallen:

Shubik, Martin. - Concepts and solutions. - (Game theory in the social sciences / Martin Shubik ; 1). – 1985

IFF-Original:

OECD, Positive Adjustment Policies, 92.64.12402.0, 1983

wäre ersetzt worden mit Dokument aus dem IDS-St.Gallen:

Jenness, R.A. - Positive adjustment in manpower and social policies / by R.A. Jenness ; Organisation for Economic Co-operation and Development [ed.]. - Paris : OECD, 1984

Alle fälschlicherweise zusammengeführten Titel wiesen eine hohe Ähnlichkeit auf. Das Problem konnte mittels Anpassens der Matching-Werte oder leichten Korrekturen der Originaldaten behoben werden.

Es wurde daher nur noch mit Skript 2 weitergearbeitet und Skript 1 verworfen.

Für den Testdatensatz test1526.csv wurden alle Treffer, bei denen ein Ersatz gefunden und somit ein Datensatz ausgetauscht werden sollte, manuell überprüft.

Dabei wurden dieselben 6 Treffer (ca. 0.4 % aller Testdaten) entdeckt, welche im obigen Vergleich gänzlich falschen Dokumenten zugeordnet worden wären – diese Fehler wurden behoben. Im Testdatensatz sollten folglich keine falschen Matches mehr vorhanden sein. Es wird davon ausgegangen, dass dies mehrheitlich auch auf den Gesamtbestand zutrifft. Da sich dies nicht überprüfen lässt, muss schlimmstenfalls von einem falschen Matching von ca. 0.4 % ausgegangen werden. Dies lässt sich ohne manuelles Nachprüfen aller Matches weder verhindern noch überprüfen. Es wird davon ausgegangen, dass aufgrund der guten Performanz auf den Testdatensatz weniger als 0.4% betroffen sind.

Bei 12 Fällen wurden **leichte Matching-Fehler** festgestellt.

Zwei Beispiele eines leichten Matching-Fehlers: Das IFF-Dokument (Print-Ausgabe) wurde mit einer Online-Ausgabe aus swissbib oder IDSSG ersetzt:

Beispiel 1: Original des IFF:

Böhret Carl, Grundriss der Planungspraxis, 3.531.11199.X, 223, Druckerzeugnis, Mittelfristige Programmplanung und angewandte Planungstechniken, Opladen, Westdeutscher Verlag, 1975

Beispiel 1: Gefundener Match im IDSSG:

Böhret, Carl. - Grundriss der Planungspraxis : mittelfristige Programmplanung und angewandte Planungstechniken / Carl Böhret. - Opladen : Westdeutscher Verlag, 1975. - 1 Online-Ressource (228 S.). - ISBN 3-531-11199-X978-3-322-84050-9 (eISBN)

Beispiel 2: Original des IFF:

Stachels Elmar, Das Stabilitätsgesetz im System des Regierungshandelns, 209, Druckerzeugnis, Berlin, de Gruyter, 1970

Beispiel 2: Gefundener Match in swissbib:

Stachels, E. (1970). Das Stabilitätsgesetz Im System des Regierungshandelns. De Gruyter, Inc.. online: https://www.degruyter.com/viewbooktoc/product/65331

Zwei Beispiele eines leichten Matching-Fehlers: Das IFF-Dokument (Einzelband oder -ausgabe) wurde einer Aufnahme eines mehrbändigen Werkes oder Zeitschriftenaufnahme ersetzt (teilweise einer anderen Übersetzung):

Beispiel 1: Original des IFF:

OECD, Higher Education Management Vol.4/No.2, 92-64-13759-9, 127-260, Druckerzeugnis, Journal of the Programme on Institutional Management in higher Education, Paris, OECD, 1992

Beispiel 1: Gefundener Match im IDSSG:

Gestion de l'enseignement supérieur / <ed.>: Centre pour la Recherche et Innovation dans l'Enseignement. - Paris : OECD. - Erscheint in frz. und engl. Sprache. - ISSN 1013-8501

Beispiel 2: Original des IFF:

Banca d'Italia, I bilanci degli istituti di emissione italiani dal 1845 al 1936, Tomo II, 700, Druckerzeugnis, Tomo II: altre serie storiche di interesse monetario e fonti, Rom, 1967

Beispiel 2: Gefundener Match im IDSSG:

Studi e ricerche sulla moneta / a cura di Renato de Mattia. - Roma : Banca d'Italia, 1967-1990. - 3 Bde. in 7.
Vol. 3, tomo 1-2: Storia del capitale della Banca d'Italia e degli istituti predecessori
Vol. 1, tomo 1-2: I bilanci degli istituti di emissione italiani dal 1845 al 1936, altre serie storiche di interesse monetario e fonti
Vol. 2, tomo 1-3: Storia delle operazioni degli istituti di emissione italiani dal 1845 al 1936 attraverso i dati dei loro bilanciNE: De Mattia, Renato

Die Bereinigung dieser Fehler hätte zu sehr grossen Verschlechterungen der allgemeinen Erfolgsrate des Programms geführt (d.h. man hätte sehr viele Fälle von einem Matching ausnehmen müssen, die tatsächlich korrekt zusammengeführt wurden).

Diese Fälle sind zwar gemäss Regelwerk falsch, jedoch gehören die Dokumente inhaltlich zum selben Titel und somit fällt der Informationsverlust bei Ersatz des Originaldokuments gering aus. Diese Art von Fehlern wird in Kauf genommen und nicht weiter optimiert.

Für den Testdatensatz traf dies auf 12 erkannte Fälle zu, was weniger als 1 % (genau: 0.8%) des Datensatzes entspricht. In der Annahme, dass der Testdatensatz repräsentativ für den Gesamtbestand ist, kann davon ausgegangen werden, dass mit der Endfassung des Skript 2 etwa 1 % aller Daten nicht korrekt und mit oben beschriebenen Fehlern ersetzt werden. In genauen Zahlen ausgedrückt: es muss bei etwa 152 Dokumenten mit Fehlern und leichtem Informationsverlust gerechnet werden.

5.4.3 Effektivität für swissbib

Das Skript 2 (Stand Mai 2019) liefert für die swissbib-Schnittstelle die Ergebnisse in *Tabelle 22*.

Um die Effektivität eines Verfahrens zu messen, werden die Formeln für Recall und Precision aus *Kapitel 3.1.7* verwendet.

Da alle Daten bereits eingespielt sind, ist die Anzahl der **True Positives** bekannt: Alle gefundenen IFF-Datensätze sind Dubletten, abzüglich derjenigen, bei welchen nur das IFF-Original gefunden wird (siehe True Negatives).

Für die Auswertung wird nicht unterschieden zwischen Exact Match und Partial Match (siehe *Kapitel 3.1.7*). Als Match wird ein Ersatz oder Rückimport aus dem Datenpool (swissbib) oder dem eigenen Verbund (IDSSG) verstanden, aber auch das Best-Case-Szenario, wo der Match bereits vollzogen wurde. Es gibt natürlich gute und weniger gute Matches, wie im *Kapitel 5.4.2* beschrieben.

	Testdatensatz test1526.csv		Gesamter Datenbestand	
	Prozent (%)	**Datensätze**	**Prozent (%)**	**Datensätze**
Total	**100**	**1526**	**100**	**15260**
Total Ersatz gefunden (TP)	**49.49**	**755**	**48.47**	**7395**
Ersatz aus swissbib	12.65	193	11.97	1826
Rückimport aus swissbib	2.69	41	3.97	605
Ersatz innerhalb IDSSG	16.19	247	16.04	2448
Bereits dedupliziert	17.96	274	16.49	2516
Total nur IFF-Original gefunden (TN)	**22.28**	**340**	**23.78**	**3629**
Total manuell zu erledigen (FN)	**4.12**	**63**	**4.58**	**698**
Nicht gefunden	2.42	37	2.41	367
Original nicht gefunden	1.57	24	1.70	260
Unsicherer Match	0.13	2	0.47	71
Vom Verfahren ignoriert: Zeitschriften/Blacklist	**24.12**	**368**	**23.19**	**3538**

Tabelle 22: Effektivität mit swissbib Schnittstelle (eigene Darstellung)

Legende zur Tabelle: TP = True Positives / FN = False Negatives / TN = True Negatives

Als **True negatives** gelten bei diesem Datenbestand die Datensätze, bei denen nur das IFF-Original gefunden wurde, da sie ja ohne die Einspielung im Mai 2018 nicht existieren würden.

Unter **False positives** werden die möglichen falschen Treffer verstanden, welche im vorhergehenden Kapitel erwähnt sind. Zwar wurden diese im Testdatensatz behoben, es wird aber hier mit dem Worst-Case-Szenario von **6 Dokumenten** für

den **Testbestand** gerechnet, da durchaus möglich ist, dass im Gesamtbestand solche falschen Matches verbleiben. Hochgerechnet auf den **Gesamtdatenbestand** bedeutet dies eine geschätzte Anzahl von maximal 60 Datensätzen.

Als **False negatives** werden alle nicht erkannten Dubletten gezählt:

- nicht gefundene Dokumente
- nicht gefundene Original-IFF-Dokumente
- unsichere Matchings

Die Zeitschriften/Blacklist-Datensätze, welche vom Verfahren ausgenommen sind, werden für die Berechnung der Effektivität nicht berücksichtigt.

Für die Berechnung der Effektivität werden somit folgende Werte verwendet (*Tabelle 23*):

	Testdatensatz	**Gesamtbestand**
True positives	**755**	**7395**
False positives	**6**	**60**
False negatives	**63**	**698**
True negatives	**340**	**3629**

Tabelle 23: Werte für die Effektivitätsberechnung (eigene Darstellung)

Zur Erinnerung hier nochmals die verwendeten Formeln:

$$precision = \frac{|true\ positives|}{|true\ positives| + |false\ positives|} = \frac{|true\ positives|}{|gefundene\ Dubletten|}$$

$$recall = \frac{|true\ positives|}{|true\ positives| + |false\ negatives|} = \frac{|true\ positives|}{|tatsächliche\ Dubletten|}$$

Für den Testdatensatz ergeben sich folgende Werte:

$$precision = \frac{755}{761} = \mathbf{0.9921}$$

$$recall = \frac{755}{818} = \mathbf{0.9229}$$

Für den Gesamtbestand ergeben sich folgende Werte:

$$precision = \frac{7395}{7455} = \mathbf{0.9919}$$

$$recall = \frac{7395}{8093} = \mathbf{0.9138}$$

Die Werte für Precision und Recall sind sehr hoch, das IFF-Verfahren hat somit eine gute Effektivität.

5.4.4 Effektivität für GVI

Da die Einbindung der GVI-Schnittstelle in das Programm erst nach der erfolgreichen Deduplizierung aus swissbib vorgenommen wurde, lassen sich die Zahlen zwischen GVI und swissbib schwer vergleichen. Für die GVI-Schnittstelle ergibt die Deduplizierung (Skript 4, Stand August 2019) folgende Resultate (*Tabelle 24*):

	Testdatensatz test1526.csv		**Datensatz swissbib_iffonly.csv**	
	Prozent (%)	**Datensätze**	**Prozent (%)**	**Datensätze**
Total gefunden	**56.23**	**858**	**47.37**	**1583**
Total manuell zu erledigen	**43.78**	**668**	**52.64**	**1759**
Nicht gefunden	16.84	257	43.81	1464
Unsicherer Match	2.56	39	8.83	295
Zeitschriften/Blacklist	24.38	372	-	-
Total	**100**	**1526**	**100**	**3342**

Tabelle 24: Effektivität mit GVI-Schnittstelle (eigene Darstellung)

Eine Auswertung des Gesamtdatensatzes mit der GVI-Schnittstelle macht wenig Sinn, daher wird in *Tabelle 24* nur ein sinnvoller Teilbestand (siehe Empfehlung

in *Kapitel 5.2.6)* verglichen. Der Datensatz swissbib_iffonly.csv enthält die Datensätze, welche im ersten Durchlauf mit swissbib nicht dedupliziert werden konnten. Er wurde jedoch mit einer anderen Version (Skript 4) erstellt als die Daten der Auswertung in *Kapitel 5.4.3* (Skript 2). Daher unterscheiden sich die Zahlen der Dokumente, für die nur ein IFF-Original gefunden wurde. Es gibt auch einige Abweichungen im Programm (z.B. Titeländerungen auf der Blacklist).

Da die Anzahl der True Positives (= Anzahl aller Dubletten) nicht bekannt ist, kann die Effektivität für den GVI nicht berechnet werden. Im Vergleich mit den Zahlen von swissbib lässt sich aber ableiten, dass der GVI mit demselben Testdatensatz besser abschneidet, da das Programm in diesem Datenpool grundsätzlich mehr Dubletten findet. Dies lässt sich durch den grösseren Datenbestand leicht erklären. Dass über den GVI mit dem Datensatz swissbib_iffonly.csv prozentual weniger Titel gefunden werden, liegt vor allem an den Fachgebieten. Viele der Dokumente aus dieser Datei kommen aus dem juristischen Bereich (Schweizer oder St. Galler Recht), auch hat es in den Originaldaten des IFF viele analytische Aufnahmen und Graue Literatur, welche das IFF selber produziert (eigene Seminare, Kongresse, etc.). Es ist nachvollziehbar, dass sich in den deutschen Verbünden wenige solche Dokumente finden lassen.

5.4.5 Effizienz

Zur Messung der Laufzeiten in *Tabelle 25* wurde das Perl-Modul «Time::HiRes» verwendet[48]. Da für jedes Dokument eine Abfrage an den SRU-Service von swissbib oder GVI gemacht wird, ist die gesamte Laufzeit des Programms stark vom verwendeten Netzwerk und Verfügbarkeit des SRU-Servers abhängig. Daher wurde die Zeit vom Absenden der SRU-Anfrage bis zum Zurückliefern der Resultate separat gemessen (in *Tabelle 25* als «Remote-Prozesse» aufgeführt). Die verbleibende Zeit entspricht den lokalen Prozessen, welche unabhängig von Netzwerkabfragen funktionieren (z.B. die Normalisierungsroutinen, die Vergleiche der Strings, das Schreiben der Exportdateien). Diese Zeiten sind in *Tabelle 25* als «Lokale Prozesse» gekennzeichnet und können verglichen werden. Mithilfe

[48] Time::HiRes: perldoc.perl.org/Time/HiRes

der lokalen Prozess-Zeiten kann die Effizienz des Programms beurteilt werden. Alle Zeiten wurden vom selben Rechner, im selben Netz und mit demselben Programm (Skript 4) durchgeführt.

	GVI				swissbib			
Anz. Datensätze	15	152	1526	15260	15	152	1526	15260
Total Laufzeit (Sek.)	6.54	83.42	657.43	6330.85	1.50	16.33	116.91	1283.42
Remote-Prozesse (Sek.)	6.14	78.81	617.29	5940.78	1.14	13.24	100.24	1034.31
Lokale Prozesse (Sek.)	0.40	4.61	40.14	390.07	0.36	3.09	16.67	249.12

Tabelle 25: Effizienzmessung GVI/swissbib (eigene Darstellung)

Wie aus *Tabelle 25* ersichtlich, wächst die Laufzeit des Programms weitgehend linear. Leichte Abweichungen ergeben sich bei beiden Schnittstellen je nach Tageszeit und Netzwerk. Die Zahlen in *Tabelle 25* entsprechen einer durchschnittlichen Messung. Die Remote-Prozesse unterscheiden sich stark aufgrund der sehr unterschiedlichen Antwortzeiten der beiden SRU-Server. Die unterschiedlichen Laufzeiten der beiden Schnittstellen für die lokalen Prozesse waren zunächst überraschend, konnten jedoch durch eine zweite Messmethode erklärt werden. Dazu wurde in einer Kopie des Skripts 4 (Datei dedup_timer.pl, siehe *Anhang C*) eine andere Zeitmessung angewendet. Der Aufruf dieses Skripts funktioniert prozedural gleich wie Skript 4, gibt jedoch die Zeitmessung jedes einzelnen Schrittes auf der Konsole aus. Dabei lässt sich leicht feststellen, dass die Suche im GVI für jeden IFF-Datensatz deutlich mehr Treffer liefert als die swissbib-Suche. Damit erhöht sich die lokale Verarbeitungszeit. Auch muss im GVI häufiger eine zweite Suchanfrage abgesetzt werden. Dies erklärt die höhere Laufzeit für die lokalen Prozesse mit der GVI-Schnittstelle. Ein Ausschnitt dieser Zeitmessung ist in *Abbildung 28* dargestellt. Die Prozesse wurden in Sekunden gemessen (auf zwei Nachkommastellen gerundet).

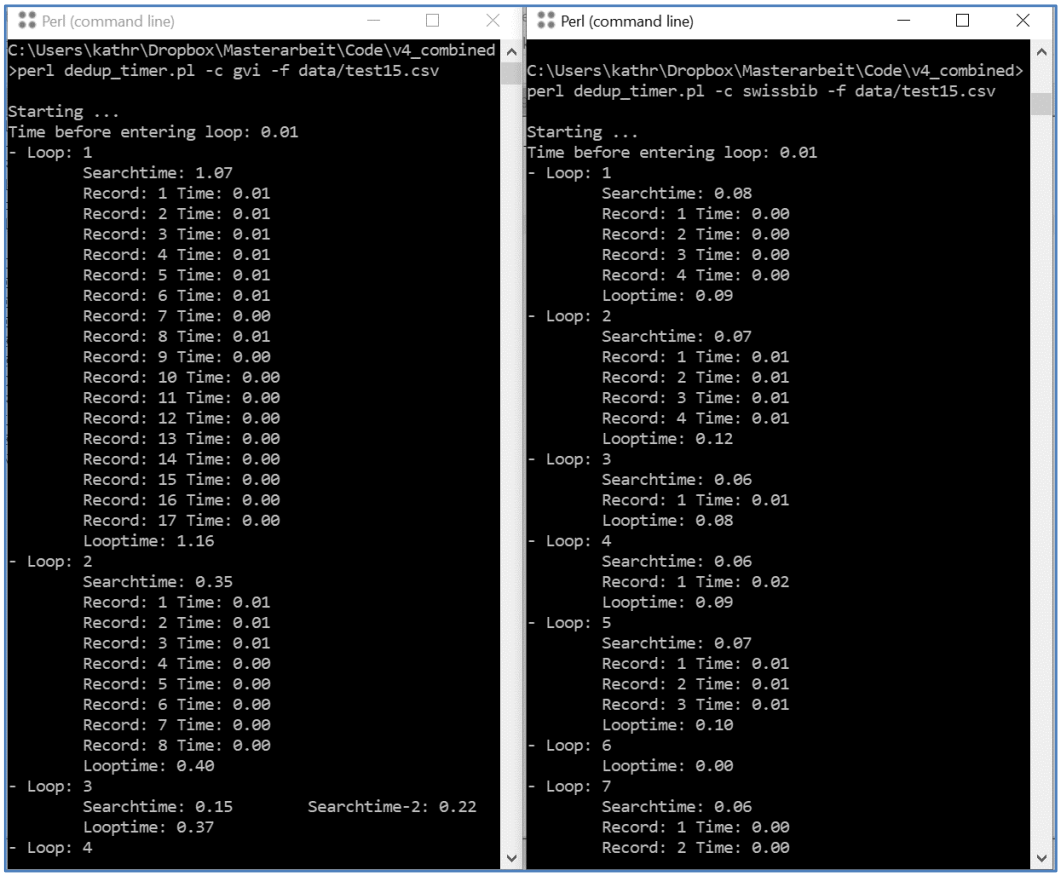


Abbildung 28: Laufzeitmessungen GVI (links) und swissbib (rechts)

Für die Deduplizierung der IFF-Daten ist die benötigte Zeit kein Problem, für einen sehr grossen Datenbestand müsste dies jedoch beachtet werden. Grundsätzlich kann davon ausgegangen werden, dass das Programm mit einem zeitgemässen Rechner und schnellen Internetverbindung für den gesamten Datenbestand des IFF mit der swissbib-Schnittstelle ca. 10-30 Minuten (abhängig vom Netzwerk) benötigt, was kein Problem darstellt. Aber auch bei einer langsameren Schnittstelle wie dem GVI kann der gesamte Bestand des IFF (über 15'000 Dokumente) mit einer guten Internetverbindung unter 2 Stunden dedupliziert werden. Dies ist immer noch ein Vielfaches schneller als die gesamte swissbib- oder GVI-Datenbank zu laden und indizieren.

5.4.5.1 Effizienz im Vergleich mit anderen Verfahren

Der Vergleich des IFF-Verfahrens mit anderen Verfahren ist etwas schwierig, da für jeden Datensatz eine Anfrage an eine Webschnittstelle geschickt wird, dafür jedoch keine Indexierung von Sortierschlüsseln vorgenommen werden muss.

Dennoch zeigen die Zahlen der Deduplizierung von SLSP (*Kapitel 3.3.6.1*) oder KOBV/GVI (*3.3.5.1*), wie viel schneller ein modernes Big-Data-Framework mit parallelisierter Deduplizierung arbeitet (*Tabelle 26,* Zahlen IFF mit swissbib-SRU-Schnittstelle).

Netzwerk	Datensätze	Benötigte Zeit	Datensätze pro Minute
swissbib	23'011'993	1320 Min. (22 h)	17'433
KOBV	30'000’000	2880 Min. (48 h)	10'416
IFF-Verfahren	15’260	21 Min. (1283 s)	726

Tabelle 26: Vergleich Datensätze pro Minute (eigene Darstellung)

5.4.6 Fazit zur Qualität des Verfahrens

Alles in allem sind die Ergebnisse des IFF-Verfahrens zufriedenstellend. Die Datenqualität wurde für einen grossen Teil der IFF-Daten verbessert.

Im Hinblick auf die erneute Datenmigration und Deduplizierung für SLSP Ende 2020 (*Kapitel 3.3.6.1*) kann auch festgehalten werden, dass allfällige, mit diesem Verfahren übersehene Dubletten, in einem Jahr erneut dedupliziert und möglicherweise vom swissbib-Verfahren als Dubletten erkannt werden.

6 Zusammenfassung und Ausblick

In diesem Kapitel findet ein Rückblick auf das erarbeitete IFF-Verfahren statt. Die verwendeten Werkzeuge werden beschrieben. Die Zielsetzung wird überprüft und gewonnene Erkenntnisse werden betrachtet.

Ein Ausblick auf die mögliche Weiterentwicklung des Verfahrens sowie Erwägungen für zukünftige Datenintegrationen bilden den Abschluss dieser Masterarbeit.

6.1 Rückblick

6.1.1 Verwendete Werkzeuge

Das Werkzeug OpenRefine für die Datenbereinigung wird in *Kapitel 4.1* beschrieben.

Als Werkzeug für das Erstellen des Programms wurde die Programmiersprache Perl gewählt. Der Entscheid basiert massgeblich auf persönlicher Vorliebe sowie Programmierkenntnissen. Die Umsetzung wäre auch mit anderen Programmiersprachen (z.B. Python oder Java) möglich gewesen. Es wurden verschiedene Perl-Module verwendet, u.a. LibXML[49] zur Verarbeitung von XML-Dateien. Die Verwendung von LibXML hat Vor- und Nachteile. Einerseits wird LibXML empfohlen für das Parsen von XML-Dateien und ist recht gut dokumentiert, es gibt Tutorials und praktische Beispiele[50]. Andererseits ist die Installation von LibXML je nach Betriebssystem ziemlich aufwändig. Da das Programm auf einer Windows-Umgebung (mit Strawberry Perl[51]) entwickelt wurde, zeigten sich diese Probleme erst bei der Übergabe an den IDSSG. Das Installieren von LibXML auf dem RedHat-Server des Bibliotheksverbundes bereitete grosse Probleme. Bei Strawberry Perl hingegen ist LibXML bereits vorinstalliert, die Installation von Strawberry Perl erfordert keine besonderen Kenntnisse.

[49] XML::LibXML: metacpan.org/pod/XML::LibXML

[50] z.B. hier: grantm.github.io/perl-libxml-by-example

[51] Strawberry Perl: strawberryperl.com

Um diesen Problemen vorzubeugen, liegt dieser Arbeit auf der DVD ein Linux-Image (virtuelle Maschine) bei, auf welcher alle benötigten Perl-Module installiert sind (siehe *Anhang C*). Sämtliche weiteren benutzten Perl-Module sind ebenfalls auf der Beilage-DVD dokumentiert.

Für grössere Projekte, insbesondere auch in Teams und im Austausch mit anderen Bibliotheken, empfiehlt sich auch Catmandu (LibreCat)[52]. Leider ist das Modul für SRU schlecht dokumentiert und kann von Nicht-Experten, entgegen den Behauptungen der Entwickler, nicht so einfach angewendet werden. Grundsätzliche Bash-Programmierung wird vorausgesetzt. Die Installation der Umgebung scheitert auch mangels Dokumentation von möglichen Problemen. Ein Ansatz, mit Catmandu zu programmieren, wurde daher verworfen. Auch Metafacture (Culturegraph)[53] wurde in Betracht gezogen, aber verworfen, da viel zu komplex für die vorliegende Situation. Der Aufwand für die Einarbeitung in diese Software Suite wäre für den Anwendungsfall nicht vertretbar gewesen.

6.1.2 Zielerreichung

Das Entwickeln eines Deduplizierungsverfahrens im Allgemeinen sowie speziell für den vorliegenden Datenbestand wurde in den *Kapiteln 3.1* und *5* aufgezeigt. Im Laufe der Entwicklung des Verfahrens wurden die Daten nochmals analysiert und bereinigt, was erhebliche Verbesserungen erbrachte (*Kapitel 4*). Mithilfe der in *Kapitel 3.2* festgelegten Kriterien und Erfahrungswerten aus anderen Verfahren konnte somit ein eigenes Verfahren definiert und realisiert werden. Die Ergebnisse sind zufriedenstellend und in der Praxis erprobt: die IFF-Daten wurden mit dem in dieser Arbeit beschriebenen Verfahren mithilfe der swissbib-Schnittstelle im Juni 2019 dedupliziert. Das Ziel gemäss *Kapitel 1.2* ist somit erfüllt. Nachfolgend werden einige Anforderungskriterien aus *Kapitel 3.2* kurz evaluiert und kommentiert. Bei den nicht erwähnten Kriterien werden die Anforderungen erfüllt.

Designziel: Ebene (*3.2.1.1*): Dieses Kriterium ist - mit den akzeptierten Einschränkungen der tolerierten Unterschiede in der Expression – erfüllt.

---

[52] Catmandu: librecat.org/catmandu

[53] Culturegraph: hub.culturegraph.org

Designziel: Dokumenttyp (*3.2.1.2*): Dieses Kriterium wird nur teilweise erfüllt. E-Books werden nicht dedupliziert, sind jedoch von der Menge her vernachlässigbar. Andere Dokumentarten werden zusätzlich zu den Vorgabekriterien berücksichtigt (z.B. Loseblattwerke).

Evaluation *(3.2.5)*: Dieses Kriterium wird erfüllt. Die verwendeten Gewichte wurden durch Testreihen ermittelt und sind speziell an die vorliegende Situation angepasst (insbesondere Gewichte für Herkunft der Daten), siehe *Kapitel 5.3.3*. Die Testreihen hätten besser dokumentiert werden sollen, um die erzielten Verbesserungen für zukünftige Parametrisierungen zu dokumentieren (*Kapitel 5.4.2*).

Merging/Datenfusion (*3.2.6*): Das Kriterium wird teilweise erfüllt. Es konnte nicht für alle Dokumente ein guter Ersatz gefunden werden (aus diversen Gründen, welche u.a. im *Kapitel 5.2.5* erläutert werden).

Qualitätsmessung: Effektivität / Effizienz (*3.2.7*): Die Effektivität kann nur für die Suche in swissbib gemessen werden, dort ist die Effektivität gut. Für den GVI können die Werte nur mit den swissbib-Werten verglichen werden. Auch dort ist die Effektivität gut. Der Programmieraufwand war hoch, aber angesichts des Lernaspekts vertretbar. Die Laufzeit und Rechenleistung ist für die vorliegenden Daten gut. Das Verfahren ist jedoch nicht skalierbar für sehr grosse Datenbestände.

6.1.3 Gewonnene Erkenntnisse

Der relativ kleine Datenbestand des IFF war eine gute «Sandbox» für die Entwicklung eines Deduplizierungsverfahrens. Die Voraussetzungen zum Lernen und Verbessern waren durch die Tatsache, dass die Daten bereits eingespielt waren, sehr gut und erlaubten einfache Testmöglichkeiten. Es erforderte jedoch zusätzliche Überlegungen und teilweise komplizierte Abläufe im Programm, um diese Original-Einspielung von Mai 2018 wieder zu bereinigen.

Im Hinblick auf die Datenbereinigungen für die Migration nach Alma und Deduplizierung in SLSP wurden grosse Lernfortschritte gemacht und viel über die Deduplizierung in swissbib und anderen Verbünden gelernt.

Als wichtige Erkenntnis gilt auch, dass die manuelle Datenbereinigung enorme Verbesserungen bringt. Das Tool OpenRefine bietet dafür gute Möglichkeiten und kann auch von Nicht-Informatikern bedient werden. Es wird daher im IDSSG auch für andere Datenbereinigungen zukünftig zum Einsatz kommen.

Das Bewusstsein um die Herkunft und Vorgeschichte der vorliegenden Daten kann durchaus wichtig sein und soll für zukünftige Datenintegrationen von Anfang an stärker gewichtet werden.

Als Kritik am Vorgehen kann festgehalten werden, dass sich eine durchgehende und systematische Dokumentation aller Testfälle und Versuchsreihen im zeitlichen Verlauf lohnt. Das Speichern von Ergebnissen, insbesondere wie und wann bestimmte Werte geändert wurden, wurde nicht durchgehend dokumentiert. In der Abschlussphase erwiesen sich diese Zahlen aber als sehr nützlich für die wissenschaftliche Aufarbeitung, z.B. für die Auswertung der bestmatch-Distribution.

Für zukünftige Verfahren sollten auch automatisierte Tests für die Funktionen erstellt werden, um die Wartbarkeit zu erhöhen.

6.2 Ausblick

6.2.1 Ausbaumöglichkeiten des Programms

Das entwickelte Verfahren funktioniert nicht gut für Dokumente mit mehrdeutigen Titeln, welche zu viele Treffer generieren. Dies betrifft viele juristische Werke, welche zudem oft aus mehreren (Teil-)Bänden bestehen. Im IFF-Datenbestand gibt es sehr viele dieser Werke. Ein zweites Programm, welches speziell für solche Werke angepasst wurde und konkret nach bestimmten Eigenheiten von juristischen Werken sucht, hätte viele Dokumente bereinigen können, welche nun auf der Liste der manuell zu erledigenden Datensätze gelandet sind. Die Entwicklung eines zusätzlichen Spezialverfahrens würde insbesondere bei grösseren Datenmengen oder einem kontinuierlichen Verfahren Sinn machen.

Dasselbe gilt für Zeitschriften. Hier liegt jedoch der manuelle Aufwand beim vorliegenden Datenbestand im Rahmen. Ein zusätzliches Programm für die Zeitschriften hätte den zeitlichen Aufwand, der für die manuelle Deduplizierung erforderlich war, mit Sicherheit überschritten.

Für die analytischen Aufnahmen fehlen die Jahresangaben, was insbesondere für das Matching in swissbib einen Nachteil bildet, da das Jahr dort als Voraussetzung für eine Deduplizierung benötigt wird. Ausserdem wäre es wünschenswert, für die analytischen Aufnahmen den Uplink zur Quelle (MARC-Feld 773) zu prüfen und allenfalls mitzuliefern, wenn die Quelle im eigenen Verbund vorhanden ist. Eine solche Verbesserung macht nach der Migration zu SLSP mehr Sinn, wenn viel mehr Daten für solche Verlinkungen verfügbar sind.

Das Programm wurde für zwei Schnittstellen ausprogrammiert und lässt sich recht leicht ausbauen für weitere SRU-Schnittstellen. So könnten nicht gefundene Datensätze weiter optimiert werden. Für die vorliegenden Daten ist es fraglich, ob es einen Datenpool gibt, der die Resultate noch signifikant verbessern kann (viele verbleibende Dokumente sind Graue Literatur, die vom IFF selbst produziert wird). Für andere Datenmigrationen ist dies durchaus eine Option.

Eine weitere Ausbaumöglichkeit wäre eine Gültigkeitsprüfung der ISBN, um unnötige Vergleiche einzusparen oder möglicherweise Tippfehler in der ISBN zu korrigieren.

6.2.2 Anreicherungsmöglichkeiten

Im *Kapitel 4.2.4* wurde bereits die Möglichkeit erwähnt, die IFF-Daten, für welche kein Ersatz gefunden wurde, mit GND-Daten anzureichern, zumindest für die Autoren des IFF oder der Universität St.Gallen. Ein Ansatz, wie die GND-Nummer für häufig vorkommende Autoren einfach extrahiert werden kann, wird im erwähnten Kapitel beschrieben. Dieser Schritt wurde im Programm nicht umgesetzt und wäre eine Erweiterung.

Das Anreichern der bestehenden Daten (ob nun mit dem Pfeffer-Verfahren oder einem Linked-Data-Verfahren) wäre ein durchaus spannender Ansatz, konnte jedoch im Rahmen dieser Arbeit nicht umgesetzt werden.

6.2.3 Nutzung von Frameworks

Die für moderne Verfahren genutzten Frameworks, wie z.B. Metafacture *(Kapitel 3.3.4*) oder auch LibreCat[54] sind ausgezeichnete Open-Source-Frameworks, für die relativ kleine IFF-Datenintegration jedoch überdimensioniert. Für grössere Datenintegrationen und zur besseren Nachnutzung sollten für zukünftige Verfahren jedoch solche Frameworks benutzt werden. Dazu wäre es wünschenswert und motivierend, wenn auch für Metafacture ein einfach verständliches Tutorial für Einsteiger vorhanden wäre. Für LibreCat gibt es dies bereits[55], was den Einstieg erleichtert.

[54] LibreCat: librecat.org

[55] Tutorial siehe: librecatproject.wordpress.com/tutorial

7 Literaturverzeichnis

Baksik, C. M. & Koerber, J. (2019). *Alma Matching Algorithms,* Harvard College, Library Technology Services. Zugriff am 27.08.2019. Verfügbar unter https://wiki.harvard.edu/confluence/display/LibraryStaffDoc/Alma+Matching+Algorithms

Bleiholder, J. & Schmid, J. (2015). Datenintegration und Deduplizierung. In K. Hildebrand, M. Gebauer, H. Hinrichs & M. Mielke (Hrsg.), *Daten- und Informationsqualität. Auf dem Weg zur Information Excellence* (S. 121–140). Wiesbaden: Springer Vieweg. https://doi.org/10.1007/978-3-658-09214-6

Coyle, K. (1992). *Rules for Merging MELVYL Records. Technical Report No. 6. Revised*. Oakland: Division of Library Automation, University of California.

Deutsche Nationalbibliothek (Hrsg.). (2019). *GVI – Gemeinsamer-Verbünde-Index.* Zugriff am 26.08.2019. Verfügbar unter https://www.dnb.de/DE/Professionell/Standardisierung/AGV/_content/gvi.html

A dictionary of computer science. (2016) (7th ed.). Oxford: Oxford University Press. https://doi.org/10.1093/acref/9780199688975.001.0001

Ex Libris (Hrsg.). (2015a). *Duplicate Detection Vector*. Primo Technical Guide. Zugriff am 27.08.2019. Verfügbar unter https://knowledge.exlibrisgroup.com/Primo/Product_Documentation/Primo/Technical_Guide/030Duplicate_Detection_Process/030Duplicate_Detection_Vector

Ex Libris (Hrsg.). (2015b). *Managing Import Profiles*. Alma Product Documentation. Zugriff am 27.08.2019. Verfügbar unter https://knowledge.exlibrisgroup.com/Alma/Product_Documentation/010Alma_Online_Help_(English)/040Resource_Management/060Record_Import/020Managing_Import_Profiles

Ex Libris (Hrsg.). (2017). *Understanding the Dedup and FRBR Processes (Primo VE)*. Primo VE Product Documentation. Zugriff am 27.08.2019. Verfügbar unter

https://knowledge.exlibrisgroup.com/Primo/Product_Documentation/020Primo_VE/040Dedup_and_FRBR_for_Primo_VE/010Understanding_the_Dedup_and_FRBR_Processes_(Primo_VE)

Fischer, P. & Hofer, P. (2011). *Lexikon der Informatik* (15., überarb. Aufl.). Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-642-15126-2

Gatenby, J. (2012). GLIMIR: Manifestation and Content Clustering within WorldCat. *Code4Lib Journal*, (17, 2012-06-01). Zugriff am 27.08.2019. Verfügbar unter http://journal.code4lib.org/articles/6812

Hickey, T. B. & Rypka, D. J. (1979). Automatic detection of duplicate monographic records. *Journal of library automation*, *12*(2), 125–142.

Hickey, T. B. & Toves, J. (2009). *FRBR Work-Set Algorithm. version 2.0*, OCLC Online Computer Library Center, Inc. Zugriff am 27.08.2019. Verfügbar unter https://www.oclc.org/content/dam/research/activities/frbralgorithm/2009-08.pdf

Hildebrand, K., Gebauer, M., Hinrichs, H. & Mielke, M. (Hrsg.). (2015). *Daten- und Informationsqualität. Auf dem Weg zur Information Excellence*. Wiesbaden: Springer Vieweg. https://doi.org/10.1007/978-3-658-09214-6

Kuberek, M. (1999). *Dublettenbehandlung (Match-und Merge-Verfahren) in der KOBV-Suchmaschine. Grundlagen* (Preprint SC 99-16 (Juni / Dezember 1999)). Berlin: Konrad-Zuse-Zentrum für Informationstechnik.

Leu, F. (2009). *Laden von Katalogisaten mit halbautomatischer Doublettenkontrolle,* IDS St. Gallen. Zugriff am 27.08.2019. Verfügbar unter https://wiki.unisg.ch/doku.php?id=aleph_admin:doublettenkontrolle

Lohrum, S. (1999). *De-duplication in KOBV* (Preprint SC 99-05 (June 1999)). Berlin: Konrad-Zuse-Zentrum für Informationstechnik.

Lohrum, S., Kirchhoff, T., Reh, U. & Winkler, S. (2019, April). *De-Duplikationsverfahren und Einsatzszenarien im Gemeinsamen Verbündeindex (GVI).* KIM Workshop, Mannheim. Zugriff am 31.08.2019. Verfügbar unter https://wiki.dnb.de/download/attachments/146377939/2019-04-03_KIMWS19_GVI.pdf

Mattmann, B. (2018). *Was kommt die kommenden Jahre auf swissbib zu?,* Swissbib. swissbib-info: 22.11.2018. Zugriff am 30.08.2019. Verfügbar unter https://swissbib.blogspot.com/2018/11/was-kommt-die-kommenden-jahre-auf.html

Munzinger Archiv/Duden. Das grosse Wörterbuch der deutschen Sprache. (2012) (4. Aufl.). Berlin: Brockhaus; Munzinger Archiv GmbH. Zugriff am 11.05.2019. Verfügbar unter http://www.munzinger.de/search/query?f=query&qid=query-duden

Naumann, F. (2007). Methoden der Dublettenerkennung. Dubletten effektiv und effizient finden. *is report*, *11*(5), SPECIAL IQ report No.2.

Naumann, F. & Herschel, M. (2010). *An introduction to duplicate detection:* Morgan & Claypool. https://doi.org/10.2200/S00262ED1V01Y201003DTM003

Pfeffer, M. (2014). Using Clustering Across Union Catalogues to Enrich Entries with Indexing Information. In M. Spiliopoulou, L. Schmidt-Thieme & R. Janning (Hrsg.), *Data analysis, machine learning and knowledge discovery* (S. 437–445). Cham [u.a.]: Springer. https://doi.org/10.1007/978-3-319-01595-8

Pfeifer, B. & Polak-Bennemann, R. (2016). Zusammenführen was zusammengehört. Intellektuelle und automatische Erfassung von Werken nach RDA. *o-bib. Das offene Bibliotheksjournal*, *3*(4), 144–155. https://doi.org/10.5282/O-BIB/2016H4S144-155

Rusch, B. (1999). *Normierungen von Zeichenfolgen als erster Schritt des Match. Zur Dublettenbehandlung im Kooperativen Bibliotheksverbund Berlin-Brandenburg* (Preprint SC 99-13 (März - Dezember 1999)). Berlin: Konrad-Zuse-Zentrum für Informationstechnik. Zugriff am 28.08.2019. Verfügbar unter http://edok01.tib.uni-hannover.de/edoks/e001/322514606.pdf

Sitas, A. & Kapidakis, S. (2008). Duplicate detection algorithms of bibliographic descriptions. *Library Hi Tech*, *26*(2), 287–301. https://doi.org/10.1108/07378830810880379

Steeg, F. & Pohl, A. (2018). *GND reconciliation for OpenRefine.* Zugriff am 31.08.2019. Verfügbar unter http://blog.lobid.org/2018/08/27/openrefine.html

Stevenson, A. (2015). *New Oxford American dictionary* (3. ed.). New York: Oxford Univ. Press. https://doi.org/10.1093/acref/9780195392883.001.0001

Subasic, I., Gvozdenovic, N. & Jack, K. (2016). De-duplicating a large crowd-sourced catalogue of bibliographic records. *Program: electronic library and information systems*, *50*(2), 138–156. https://doi.org/10.1108/PROG-02-2015-0021

Swissbib (Hrsg.). (2007). *Deduplication study.* Zugriff am 26.08.2019. Verfügbar unter http://www.swissbib.org/wiki/index.php?title=Deduplication_study

Swissbib (Hrsg.). (2018). *Matching and merging.* Zugriff am 26.08.2019. Verfügbar unter http://www.swissbib.org/wiki/index.php?title=Matching_and_merging

Toney, S. R. (1992). Cleanup and Deduplication of an International Bibliographic Database. *Information Technology and Libraries*, *11*(1), 19.

Viegener, T. (2010). *Metadatenstandards und moderne Suchoberflächen. Befunde aus dem Projekt swissbib.* E-Lib.ch. Zugriff am 28.08.2019. Verfügbar unter https://de.slideshare.net/Aliverti/viegenerswissbib

Vorndran, A. (2018). Hervorholen, was in unseren Daten steckt! Mehrwerte durch Analysen grosser Bibliotheksdatenbestände. *o-bib. Das offene Bibliotheksjournal*, *5*(4), 166–180. https://doi.org/10.5282/O-BIB/2018H4S166-180

Wiesenmüller, H. & Pfeffer, M. (2013). Abgleichen, anreichern, verknüpfen. Das Clustering-Verfahren - eine neue Möglichkeit für die Analyse und Verbesserung von Katalogdaten. *BuB - Forum Bibliothek und Information*, *65*(09), 625–629.

Witzig, S. & Hipler, G. (2019, April). *Clustern von Daten auf der swissbib Plattform.* KIM Workshop 2019, Mannheim. Zugriff am 28.08.2019. Verfügbar unter https://wiki.dnb.de/download/attachments/146377939/2019-04-03_KIMWS19_Witzig-Hipler_swissbib.pdf

Zwirner, M. (2015). Datenbereinigung zielgerichtet eingesetzt zur permanenten Datenqualitätssteigerung. In *Daten- und Informationsqualität : Auf dem Weg zur Information Excellence* (S. 101–120). Wiesbaden: Springer Vieweg.

Anhänge

A. Rohdaten IFF

Die IFF-Daten liegen im CSV-Format vor (*Abbildung 29*). Die Originaldaten sind im leichter lesbaren Excel-Format in *Tabelle 27* abgebildet (Auszug), die bereinigten Daten in *Tabelle 28*.

Die gesamten Rohdaten befinden sich auf der DVD (siehe *Anhang C*).

Für das Deduplizierungsverfahren müssen die Daten in einer CSV-Datei in folgenden Spalten verfügbar sein:

1-3: Autor(en)

4-7: Titelinformationen

8: ISBN

9: Seitenzahlen

10: Materialart (kontrolliertes Vokabular, z.B. Druckerzeugnis, Loseblattwerk)

11: Zusatz

12-13: Standort/Signatur

14: Erscheinungsort

15: Verlag

16: Jahr

17-19: code1, code2, code3 (Stichwort-Code)

20-22: Schlagwörter

Spalte 24 ff. sollten nicht belegt sein mit Daten.

```
"author";"title";"isbn";"pages";"kind";"creation_date";"addition";"location";"signature";"place";"publisher";"year";"remarks";"subj1";"subj2";"subj3";"
subj1_t";"subj2_t";"subj3_t";
"Yerly Nadia";"The Political Economy of Budget Rules in the twenty-six Swiss Cantons";"978-2-8399-1414-7";"399";"Druckerzeugnis";"23.05.2014
11:35:46";"Institutional analysis, preferences and performances";"Haupt-Bibliothek";"";"Fribourg";"Uni Fribourg";"2013";"";"";"";"";"";"";
"Schmidhauser Bruno";"Der Begriff der 'mehreren Unternehmen' im Sinne von Art. 4 Abs. 2 KG";"3 7255 3887 5";"S. 429-446";"Druckerzeugnis";"05.12.2003
08:19:03";"In: Baldi, Marino, Baumann, Max & u.a., (Hrsg.): Der Einfluss des europäischen Rechts auf die Schweiz, Festschrift für Professor Roger
Zäch zum 60. Geburtstag";"Haupt-Bibliothek";"-";"Zürich";"Schulthess Polygraphischer Verlag";"-";"";"WB";"";"";"Uebrige Literatur";"Oeffentliches
Recht";"";
"Shultz William J./Harriss Lowell C.";"American Public Finance";"";"556";"Druckerzeugnis";"05.12.2003 08:19:03";"8. A.";"Haupt-Bibliothek";"AA
158";"New Jersey";"Prettice Hall Inc. Englewood Cliffs";"1965";"";"YB";"";"";"Wirtschaftswissenschaften";"Allgemeines. Methoden der W. Unterrichts-
und Ausbildungsmethoden in den W., Wirtschaftspädagogik";"";
"Hofer Hermann";"Die schweizerische Finanzhilfe an Lateinamerika";"";"123";"Druckerzeugnis";"06.09.2004";"Öffentliche Kolloquien, veranstaltet am
Lateinamerikanischen Institut a. d. HSG, WS 1967/68";"Haupt-Bibliothek";"AK
0507";"";"";"1968";"";"YM";"";"";"Wirtschaftswissenschaften";"Internationaler Handel. Zahlungsbilanz und Wechselkurse";"";
"Bohley/Peter";"Perspektiven des interkommunalen Finanzausgleichs";"-";"107-120";"Druckerzeugnis";"05.12.2003 08:19:03";"In: Eng- Franz- Glatthard-
Alexander- Koenig- Beat H.: Der interkommunale
Finanzausgleich";"Haupt-Bibliothek";"-";"-";"-";"-";"";"YL4";"";"";"Wirtschaftswissenschaften";"Finanzwissenschaft. Betriebswirtschaftliche
Steuerlehre";"Finanzausgleich";
"Haas Adrian";"Chancen und Gefahren des New Public Managements  für die Privatwirtschaft";"";"";"Druckerzeugnis";"05.12.2003
08:19:03";"";"Haupt-Bibliothek";"EF 230";"Bern";"ECOPLAN";"1996";"";"YQ";"";"";"Wirtschaftswissenschaften";"Spezielle Betriebswirtschaftslehren";"";
"OECD";"Die internationalen Bildungsindikatoren der OECD";"3-631-48240-X";"149";"Druckerzeugnis";"05.12.2003 08:19:03";"Ein
OECD/CERI-Bericht";"Haupt-Bibliothek";"EJ 150";"Frankfurt a/M";"Peter Lang";"1994";"";"YQ";"";"";"Wirtschaftswissenschaften";"Spezielle
Betriebswirtschaftslehren";"";
"Schneider Friedrich";"Der Einfluss von Interessengruppen auf die Wirtschaftspolitik";"3.258.03421.4";"204";"Druckerzeugnis";"06.09.2004";"Eine
empirische Untersuchung für die Schweiz";"Haupt-Bibliothek";"AK 0457";"Bern und Stuttgart";"Paul
Haupt";"1985";"";"YR";"";"";"Wirtschaftswissenschaften";"Gewerbepolitik. Einzelne Wirtschaftszweige";"";
"Flick Hans/Wassermeyer Franz/Baumhoff Hubertus/Schönfeld Jens";"Aussensteuerrecht - Bd. III";"3-504-26041-6";"-";"Loseblattwerk";"05.12.2003
08:19:03";"Kommentar";"HB Steuerrecht";"KB 308 3";"Köln";"Verlag Dr. Otto Schmidt";"2017";"";"KB";"";"";"Ausländisches Steuerrecht";"Deutschland";"";
"Nystad Arild N.";"Application of Control and System Theory to Macroeconomics: A Survey";"";"52";"Druckerzeugnis";"05.12.2003
08:19:03";"";"Haupt-Bibliothek";"AK 1063";"St. Gallen";"Forschungsgemeinschaft für
Nationalökonomie";"1981";"";"YC";"";"";"Wirtschaftswissenschaften";"Wirtschaftstheorie, einschliesslich Geldtheorie";"";
"Pälli Alfred";"Die sozio-ökonomischen Bestimmungsfaktoren der Bildung von Wohneigentum in der
```

Abbildung 29: IFF-Daten, Ansicht im CSV-Format (Originaldaten)

author	title	isbn	pages	kind	addition	place	publisher	year
Yerly Nadia	The Political Economy of Budget Rules in the twenty-six Swiss Cantons	978-2-8399-1414-7	399	Druckerzeugnis	Institutional analysis, preferences and performances	Fribourg	Uni Fribourg	2013
Schmidhauser Bruno	Der Begriff der 'mehreren Unternehmen' im Sinne von Art. 4 Abs. 2 KG	3 7255 3887 5	S. 429-446	Druckerzeugnis	In: Baldi, Marino, Baumann, Max & u.a., (Hrsg.): Der Einfluss des europäischen Rechts auf die Schweiz, Festschrift für Professor Roger Zäch zum 60. Geburtstag	Zürich	Schulthess Polygraphischer Verlag	-
Shultz William J./Harriss Lowell C.	American Public Finance		556	Druckerzeugnis	8. A.	New Jersey	Prettice Hall Inc. Englewood Cliffs	1965
Hofer Hermann	Die schweizerische Finanzhilfe an Lateinamerika		123	Druckerzeugnis	Öffentliche Kolloquien, veranstaltet am Lateinamerikanischen Institut a. d. HSG, WS 1967/68			1968
Bohley/Peter	Perspektiven des interkommunalen Finanzausgleichs	-	107-120	Druckerzeugnis	In: Eng- Franz- Glatthard- Alexander- Koenig- Beat H.: Der interkommunale Finanzausgleich	-	-	-
Haas Adrian	Chancen und Gefahren des New Public Managements für die Privatwirtschaft			Druckerzeugnis		Bern	ECOPLAN	1996
OECD	Die internationalen Bildungsindikatoren der OECD	3-631-48240-X	149	Druckerzeugnis	Ein OECD/CERI-Bericht	Frankfurt a/M	Peter Lang	1994
Schneider Friedrich	Der Einfluss von Interessengruppen auf die Wirtschaftspolitik	3.258.03421.4	204	Druckerzeugnis	Eine empirische Untersuchung für die Schweiz	Bern und Stuttgart	Paul Haupt	1985
Flick Hans/Wassermeyer Franz/Baumhoff Hubertus/Schönfeld Jens	Aussensteuerrecht - Bd. III	3-504-26041-6	-	Loseblattwerk	Kommentar	Köln	Verlag Dr. Otto Schmidt	2017
Nystad Arild N.	Application of Control and System Theory to Macroeconomics: A Survey		52	Druckerzeugnis		St. Gallen	Forschungsgemeinschaft für Nationalökonomie	1981

Tabelle 27: Auszug nicht bereinigte Daten: Datei IFF_Katalog_Full.csv (eigene Darstellung)

author 1	author 2	author 3	title 1	title 2	volume 1	volume 2	isbn	pages	kind	addition	place	publisher	year
Yerly Nadia			The Political Economy of Budget Rules in the twenty-six Swiss Cantons				978-2-8399-1414-7	399	Druckerzeugnis	Institutional analysis, preferences and performances	Freiburg	Uni Fribourg	2013
Schmidhauser Bruno			Der Begriff der 'mehreren Unternehmen' im Sinne von Art. 4 Abs. 2 KG				3 7255 3887 5	429-446	Druckerzeugnis	In: Baldi, Marino, Baumann, Max & u.a., (Hrsg.): Der Einfluss des europäischen Rechts auf die Schweiz, Festschrift für Professor Roger Zäch zum 60. Geburtstag	Zürich	Schulthess	
Shultz William J.	Harriss Lowell C.		American Public Finance					556	Druckerzeugnis	8. A.	New Jersey	Prettice Hall Inc. Englewood Cliffs	1965
Hofer Hermann			Die schweizerische Finanzhilfe an Lateinamerika					123	Druckerzeugnis	Öffentliche Kolloquien, veranstaltet am Lateinamerikanischen Institut a. d. HSG, WS 1967/68			1968
Bohley Peter			Perspektiven des interkommunalen Finanzausgleichs				-	107-120	Druckerzeugnis	In: Eng- Franz- Glatthard- Alexander- Koenig- Beat H.: Der interkommunale Finanzausgleich			
Haas Adrian			Chancen und Gefahren des New Public Managements für die Privatwirtschaft						Druckerzeugnis		Bern	ECOPLAN	1996
OECD			Die internationalen Bildungsindikatoren der OECD				3-631-48240-X	149	Druckerzeugnis	Ein OECD/CERI-Bericht	Frankfurt	Peter Lang	1994
Schneider Friedrich			Der Einfluss von Interessengruppen auf die Wirtschaftspolitik				3.258.03421.4	204	Druckerzeugnis	Eine empirische Untersuchung für die Schweiz	Bern	Paul Haupt	1985
Flick Hans	Wassermeyer Franz	Baumhoff Hubertus	Aussensteuerrecht	Bd. III			3-504-26041-6		Loseblattwerk	Kommentar	Köln	Otto Schmidt	2017
Nystad Arild N.			Application of Control and System Theory to Macroeconomics	A Survey				52	Druckerzeugnis		St. Gallen	Forschungsgemeinschaft für Nationalökonomie	1981

Tabelle 28: Auszug bereinigte Daten: Datei IFF_Katalog_FULL_normalized.csv (eigene Darstellung)

B. Konfiguration IFF

Die Schnittstelle sowie die Matching-Werte können über eine Konfigurationsdatei geändert werden. Die vollständigen Konfigurationsdateien sowie eine Kurzdokumentation befinden sich auf der beiliegenden DVD. Eine vollständige Dokumentation des verwendeten Perl-Moduls Config::Tiny ist in CPAN verfügbar.[56] Nachfolgend als Beispiel die swissbib-Konfiguration.

Matching-Werte (Auszug):

```
# weight values for a positive match

isbn=10

aut1=10

aut2=5

tit=15

stit=5

year=15

year_p1=5

year_m1=5

# minimum value for a safe total match:

safe=31
```

Gewichtung der Herkunft der Daten (Auszug):

```
# weights for origin of the data:
RERO=6
SGBN=4
IDS=11
# old idssg monograph:
IDSSG_OLD_M=40
# old idssg analytica:
IDSSG_OLD_A=20
# negative value for IFF original data, monograph:
IFF_M=15
# negative value for IFF original data, analytica:
IFF_A=3
```

[56] S.a. https://metacpan.org/pod/Config::Tiny

C. DVD

Auf der beiliegenden DVD befinden sich:

- **Masterarbeit**: Datei thesis.pdf
- **Programmcode**: Ordner iff-master
    - Beschreibung und Nutzungsweise: README.md
    - Ablaufdiagramm: Ablaufdiagramm.jpg und Ablaufdiagramm.svg
    - ***Finale (empfohlene) Version:*** Ordner v4_combined
        - Skript: dedup.pl
        - POD-Dokumentation: dedup_pod.md
        - Konfigurationsdateien: gvi.conf und swissbib.conf
        - MAP-Datei: iff_subject_table.map
        - Skript: dedup_timer.pl für die Laufzeitmessungen
    - ***Verwendete Daten:*** Unterordner v4_combined/data
        - Gesamte IFF-Daten: IFF_Katalog_FULL_normalized.csv
        - Testdatensatz (enthält 10% aller Daten): test1526.csv
        - Weitere Testdateien
    - Ältere, im Text erwähnte Versionen: Ordner old_versions
- Ein **Linux-Image**:
    - VM (2.3 GB): ZIP-Datei lubuntu-clean.7z
    - Infodatei zum Linux-Image: lubuntu-bibinfo-vm-howto.txt

Zum Zeitpunkt der Abgabe der Masterarbeit (September 2019) können folgende DVD-Inhalte in gleicher Form auch online heruntergeladen werden:

- **Programmcode** kann von Github geklont werden:
  https://github.com/kathrin77/iff.git
- **Linux-Image** sowie die dazugehörende Infodatei können von Switch-Drive heruntergeladen werden:
  https://drive.switch.ch/index.php/s/Tub2XFQLxU3NIXY

Selbständigkeitserklärung

Ich versichere, dass die vorliegende Arbeit von mir selbständig und ohne unerlaubte Hilfe angefertigt worden ist. Ich habe alle Stellen, die wörtlich oder sinngemäss aus Veröffentlichungen entnommen sind, durch Zitate bzw. Literaturhinweise als solche kenntlich gemacht.

Ort, Datum

Unterschrift