

Werkzeuge zur Analyse von Bibliotheksdaten

Jakob Voß (VZG) Nico Wagner (DNB)

Bibliothekskongress 2022

2022-05-31

Bibliotheksdaten

Bibliotheksdaten

- ▶ Grundsätzlich nicht anders als andere Daten
- ▶ Aber oft Bibliotheksspezifische **Metadatenformate**
(MARC21, MAB2, PICA, ASEQ, METS/MODS, ONIX...)
- ▶ Außerhalb von Bibliotheken irrelevant
⇒ wenig Support allgemeiner Datenwerkzeuge
- ▶ Datenformate altern viel langsamer als Software

Neue Anforderungen an Bibliotheksdaten

- ▶ Öffnung und Verknüpfung mit anderen Datenbeständen
- ▶ Data Science, Big Data, Visualisierung, Machine-Learning...

Wie mit Bibliotheksdaten umgehen?

Werkzeuge

Werkzeuge für Bibliotheksdaten

1. Allgemeine Datenwerkzeuge (z.B. Excel, OpenRefine)
2. Freie Werkzeuge für bibliothekarische Datenformate
 - ▶ Programmierbibliotheken
 - ▶ Anwendungsprogramme
3. Integrierte/Interne Werkzeuge (WinIBW, FALCON, BibControl...)

Allgemeine Datenwerkzeuge

Data Science Werkzeugkasten

- ▶ Interaktive Notebooks (Jupyter, Observable...)
- ▶ Kommandozeile (grep, head, sort...)
- ▶ OpenRefine
- ▶ GitHub/GitLab
- ▶ ...

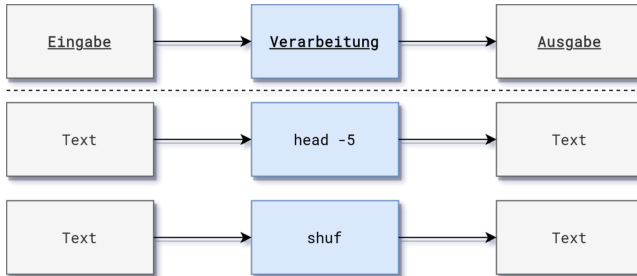
Freie Werkzeuge für Bibliotheksdaten

- ▶ Catmandu
- ▶ Metafacture
- ▶ pica-rs / picadata
- ▶ MABLE+/Marcel
- ▶ QA Catalogue, Cocoda...

Beispiele

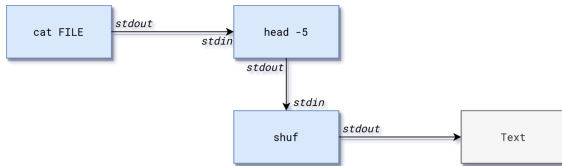
Arbeiten auf der Kommandozeile

- ▶ Programme auf der Kommandozeile folgen im Allgemeinen dem EVA-Prinzip



Verkettung von Programmen durch Pipes

- ▶ Programme können durch Pipes “|” miteinander verkettet werden



```
$ cat FILE | head -5 | shuf
```

Vorteile

- ▶ Integration in Data Science Workflows (Shell-Skripte, Makefiles, Cron-Jobs, DVC)
- ▶ Es stehen eine große Anzahl an Standardwerkzeugen zur Verfügung (head, grep)
- ▶ Umfangreiches Literatur- und Fortbildungsangebot

Zählen von PICA-Datensätzen

```
$ wc -l DUMP.dat
```

```
1000
```

```
$ picadata -f plus DUMP.dat
```

```
1000 records
```

```
36111 fields
```

```
$ pica count DUMP.dat
```

```
1000 records
```

```
36111 fields
```

```
83814 subfields
```

Filtern von PICA-Datensätzen

```
$ pica filter -s \  
    "002@.0 =^ 'Tp' && 028A.a == 'Goethe'" \  
    | pica count --records
```

14

```
$ pica filter -s "002@.0 =^ 'Tp' && 028A.a == 'Goethe'" \  
    | picadata -2 -f plain -t json "028A"  
[["028A","", "d", "Friedrich", "a", "Goethe"]]  
[["028A","", "d", "August", "c", "von", "a", "Goethe"]]
```

Tabellieren von Daten

```
$ pica filter -s  
    "002@.0 =^ 'Tp' && 028A.a == 'Goethe'" GND.dat \  
    | pica select "003@.0, 028A{a, d}"  
117749346,Goethe,Friedrich  
11854022X,Goethe,August  
118540246,Goethe,Katharina Elisabeth  
118628011,Goethe,Christiane  
...
```

Häufigkeitsverteilung eines Unterfelds

```
$ pica filter -s "002@.0 =^ 'Tg'" GND.dat \  
    | pica frequency -l 3 -H "code,count" "042B.a"  
code,count  
XA-IT,30149  
XA-DE-BY,26694  
XA-FR,17452
```


Abfrage per SRU-API

```
$ catmandu convert SRU \  
  --base http://sru.k10plus.de/opac-de-627 \  
  --recordSchema picaxml \  
  --parser picaxml \  
  --query pica.sgd=590 \  
  to PICA --type plain
```

Zwischenfazit

- ▶ Werkzeuge lassen sich miteinander kombinieren
 - ⇒ Stärken der unterschiedlichen Tools ergänzen sich
- ▶ Erzeugen von Standardformaten (CSV, JSON)
 - ⇒ Programmiersprachen und -bibliotheken nutzen
 - ⇒ Nutzen von allgemeinen Datenwerkzeuge

GND-Dashboard

- ▶ Statistische Auswertungen (Datenbasis PICA+)
- ▶ Selektion und Transformation der Daten mit *pica-rs*
- ▶ Datenaufbereitung und Visualisierung mit Python

Filtern des Gesamtabzugs

- ▶ Gesamtabzug der DNB ist ca. 44GB groß
 - ⇒ Titeldaten (title.dat): 38GB
 - ⇒ Normdaten (gnd.dat): 5.5GB

```
$ pica filter -s "002@.0 =^ 'T' && !008@.a?" DUMP.dat \  
-o gnd.dat
```

```
$ pica filter -s --invert-match "002@.0 =^ 'T'" DUMP.dat \  
-o title.dat
```

GND-Entitäten gesamt

GND Statistik allgemein

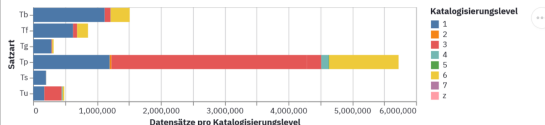
GND-Entitäten gesamt: 9.105.935

```
$ pica count --records gnd.dat  
9.105.935
```

Entitäten und Katalogisierungslevel

Entitäten und Katalogisierungslevel

Alle GND-Entitäten können in verschiedenen Katalogisierungsleveln (1-7) angelegt werden. Je niedriger das Katalogisierungslevel, desto verlässlicher die Daten, weil Sie dann von qualifizierten Personen erstellt bzw. überprüft wurden.



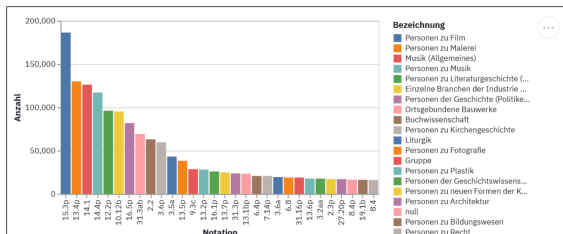
```
$ pica frequency --limit 3 "002@.0" gnd.dat
```

```
Tp3,3288170
```

```
Tp1,1197189
```

```
Tb1,1116481
```

GND-Systematik



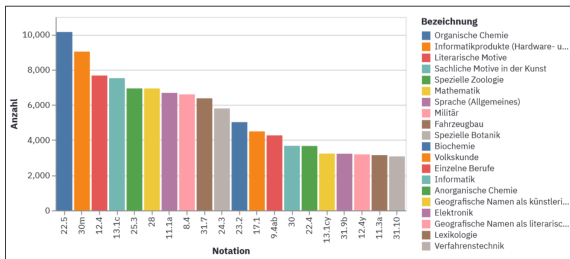
```
$ pica frequency --limit 3 "042A.a" gnd.dat
```

```
15.3p,186402
```

```
13.4p,129972
```

```
14.1,126224
```

GND-Systematik (nur Ts)



```
$ pica filter "002@.0 =^ 'Ts'" gnd.dat \
```

```
| pica frequency "042A.a"
```

```
22.5,10146
```

```
30m,9029
```

```
12.4,7671
```


Ausblick

Interoperabilität und Standards

- ▶ Werkzeuge sollten sich gut mit anderen kombinieren lassen
- ▶ Einigung auf gemeinsame Standards (Avram, PICA Path, Fix...)

Herausforderungen

- ▶ Dokumentation, Fortbildung, Einarbeitung
- ▶ Softwareentwicklung ist Daueraufgabe
- ▶ Entwicklung hängt oft an einzelnen Personen
(Beispiel: MarcEdit, zumal nur Freeware)
- ▶ Projektgetriebene Entwicklung selten nachhaltig
(Beispiel: d:swarm)

Weiterführende Ressourcen

- ▶ Dokumentation der jeweiligen Programme
- ▶ Einführung in die Verarbeitung von PICA-Daten
- ▶ Processing MARC with open source tools
- ▶ [it-in-bibliotheken]<https://it-in-bibliotheken.de> (geplant)
- ▶ Magnus Pfeffer (2016): Open Source Software zur Verarbeitung und Analyse von Metadaten. Präsentation auf dem 6. Bibliothekskongress. urn:nbn:de:0290-opus4-24490

Guter Rat zum Schluss

Datenverarbeitung ist keine Wissenschaft sondern Handwerk!

Übung macht die Meister*in!