"WIR HABEN DAS IN EXCEL ..."

"... WIE BEKOMMEN WIR DAS IN DEN KATALOG?"

Stefan Schuh

Universitätsbibliothek Graz

Österreichischer Bibliothekartag

Graz, 11. September 2019

PROJEKTE

- Handschriftenkatalog der UB Graz
- Korrespondenzen aus der Nachlasssammlung

WERKZEUGE

- Python
- OpenRefine
- Emacs

HANDSCHRIFTEN

- Nachweis der Historischen Handschriften im Bibiothekssystem
- Metadaten für das Portal der Digitalisierten Handschriften

AUSGANGSDATEN

- Handschriftenkatalog der UBG
- Übertragung in MS-Excel

AUSGANGSDATEN IN MS-EXCEL (AUSZUG)

	A	B	C	D	E	F	G	H	I	P	R	S	T	U	V	Z	AA
1	Signatur modern	Bd.	Signatur alt	Format	Beschreibstoff	Umfang	Größe h : b	1. Dat. exakt	1. Dat. ex. fol.	1. Dat. Jh.	2. Dat. Jh.	2. Dat.Jh. Fol./von-bis	3. Dat. Jh.	3. Jh. Fol./von-bis	1. Dat. ca., um, vor, Anfang, Ende	1. VB natürl. Personen	1. Vorbesitz Institution
73	62		40/16	f°	Pergament	304	45 : 33			XV/1							Benediktinerstift St. Lambrecht
74	63		40/6	f°	Pergament	326	42 : 25	1390								Ulrich von Albeck	Chorherrenstift Seckau
75	64		33/1	f°	Papier	II, 406	41 : 28			XV/2							St. Georgs-Ritterorden in Millstatt
76	65		40/21	f°	Pergament	198	41 : 29			XII	XV	1 - 22	XIV	186 - 198			Chorherrenstift Seckau
77	66		38/17	f°	Papier	347	41 : 29			XV							Chorherrenstift Seckau
78	67		37/21	f°	Papier	III, 389	41 : 28			XVI							Chorherrenstift Seckau
79	68		40/2	f°	Pergament	II, 231	41 : 30			XII							Chorherrenstift Seckau
80	69		40/4	f°	Pergament	I, 284	42 : 29			XIII/1							Chorherrenstift Seckau
81	70		42/1	f°	Papier	I, 110	41 : 28			XV/2							Benediktinerstift St. Lambrecht
82	71		40/26	f°	Pergament	172	42 : 27			XIII							Benediktinerstift St. Lambrecht
83	72		40/10	f°	Pergament	242	42 : 27			XIV/1							Benediktinerstift St. Lambrecht
84	73		33/9	f°	Papier	VII, 39	43 : 28	1644								Johann Ferdinand von Herberstein	
85	74		37/18	f°	Pergament	362	41 : 31								vor 1477	Jakob Gerold	Chorherrenstift Seckau
86	75		33/40	f°	Papier	I, 249	41 : 29			XV/2							St. Georgs-Ritterorden in Millstatt

AUFBEREITUNG DER DATEN

- Konversion von xlsx nach CSV
- Vereinheitlichung der Spaltenköpfe
- Whitespace cleanup, etc.
- Recherche von Ansetzungsform und GND-Nummer von vorbesitzenden Institutionen und Personen
- Iterativer Prozess in Zusammenarbeit mit den Erschließerinnen

ERSTELLUNG VON MARC-DATEN

- Einlesen der Daten im csv-Format in eine geeignete Datenstruktur
    - Daten können über Spaltenköpfe adressiert werden:

```
beschreibstoff = row["Beschreibstoff"]
```

- Erstellen eines MARC-Records pro Zeile (falls Daten ausreichend)

EINFACH: UNVERÄNDERTES ÜBERTRAGEN VON WERTEN

- z. B. Beschreibstoff

```
beschreibstoff = row["Beschreibstoff"]
rec.add_ordered_field(
    Field(
        tag = "340",
        indicators = [" ", " "],
        subfields = ["a", beschreibstoff])
```

SCHWIERIGER: ZUSAMMENSETZEN MEHRERER WERTE

- z. B. Umfangs- und Formatangabe

```
# Umfangsangabe in 300
if "rolle" in row["Umfang"].lower():
    sfa = row["Umfang"]
else:
    sfa = f'{row["Umfang"]} Blätter'

sfc = f'{row["Format"]}, {row["Größe h : b   "].replace(":", "x")

if sfa.startswith(" "):
    sfa = ""
if sfc.startswith(", "):
    sfc = sfc[2:]
if sfc.endswith(", "):
    sfc = sfc[:-2]
rec.add_ordered_field(
    Field(
```

AUFWÄNDIG: DATUMSANGABEN

- verteilen sich auf mehrere Spalten
- liegen nicht in Maschinenlesbarer Form vor
- maschinenlesbare Form für Feld 008/07-10 notwendig
- menschenlesbare Form kann direkt in 264 geschrieben werden

DATUM 1: AUS DER TABELLE AUSLESEN

- Es wird nur auf das erste Datum geprüft
- Prüfung nach absteigender Genauigkeit
- Erster gefundener Wert wird verwendet

```
def get_date(row):
    """Return the raw date for 264 $$c and 008"""
    data = None

    if row["1. Dat. exakt"]:
        data = row["1. Dat. exakt"]
    elif row["1. Dat. Ex. Von - bis"]:
        data = row["1. Dat. Ex. Von - bis"]
    elif row['1. Dat. ca., um, vor, Anfang, Ende']:
        data = row['1. Dat. ca., um, vor, Anfang, Ende']
    elif row["1. Dat. Jh. "]:
        data = row["1. Dat. Jh. "]
    else:
        data = "Datum unbekannt"

    return data
```

DATUM 2: MAPPING DER JAHRHUNDERTANGABEN

- Mapping für Jahrhundertangaben notwendig, weil MARC ein vierstellig numerisches Datum erzwingt

```
map_dates_008 = {
    "VI": "501",
    "IX/1": "801",
    "Mitte IX": "851",
    "Ende IX": "881",
    "Anfang X": "901",
    #---8<---8<---8<---
    }
```

DATUM 3: ZUORDNUNG DES DATUMS ANHAND DES MAPPINGS

```
def date_008(date):
    year = None
    if date in map_dates_008.keys():
        year = map_dates_008[date].zfill(4)
    else:
        re_match = re.search(r'\d{3}\d?', date)
        if re_match is not None:
            year = re_match.group(0).strip().zfill(4)
    return year
```

EINSPIELUNG INS BIBLIOTHEKSSYSTEM

- Importprofil
- Bestand
- Weitere Bearbeitung erfolgt intellektuell

KORRESPONDENZEN AUS DER NACHLASSSAMMLUNG

AUSGANGSDATEN

- Alexius Meinong im alten Nachlassportal

URSPRÜNGLICH GEPLANTER WORKFLOW

- Daten aus dem Text der HTML-Site extrahieren
- In strukturierter Form (csv/Excel) zur Verfügung stellen
- Daten intellektuell in Excel ergänzen und mit OpenRefine anreichern
- Aus dem daraus resultierendem Excel MARC-Daten erstellen und in Alma laden

DONE: DATEN AUS DEM TEXT DER HTML-SITE EXTRAHIEREN

Text:

```
A

Adamek, 0. 1-8; Adickes, E. 9-10; Adler, Guido (Briefe aus den Jahren
1877-1920) 11-164 (Meinong an Adler: siehe LXVII); Akademie der
Wissenschaften in Wien: siehe XX/e (vgl. auch: Junk, Karabacek,
Radermacher, Redlich) (Meinong an die Akademie: siehe LXVII); Alexande
S. 165; Ameseder, Rudolf 166-185; Appunn, A. 186-187; Arleth, Emil
188-191; "Arminia" (Burschenschaft) 192; Artaria, J. 193; Aster, E. vo
194; Augustin, M. 195-197; Avenarius, Richard 198-213 (Meinong an
Avenarius: siehe LXVII)
```

DONE: DATEN AUS DEM TEXT DER HTML-SITE EXTRAHIEREN

Die einzelnen Einträge einlesen

```
with open("meinong_brief.txt") as fh:
    entries = []
    for line in fh:
        if re.match(r"(^[A-Z]$)|(^$)", line):
            continue
        else:
            entries += [entry for entry in line.split(";")]
```

"Datenfelder" innerhalb der Einträge trennen

```
re.search(r'^([a-zA-ZäüöÄÜÖéá., "ß]*)(\(.*\))?([\d ,-]+)(\(.*\))?
```

DONE: IN STRUKTURIERTER FORM (CSV/EXCEL) ZUR VERFÜGUNG STELLEN

	A	B	C	
1	Name	Anm. zum Namen	Nummer(n)	Anmerkungen
2	Adamek, O.		1-8	
3	Adickes, E.		9-10	
4	Adler, Guido	(Briefe aus den Jahren 1877-1920)	11-164	(Meinong an Adler: siehe LXVII)
5	Alexander, S.		165	
6	Ameseder, Rudolf		166-185	
7	Appunn, A.		186-187	
8	Arleth, Emil		188-191	
9	"Arminia"	(Burschenschaft)	192	
10	Artaria, J.		193	
11	Aster, E. von		194	

CANCELED: DATEN INTELLEKTUELL IN EXCEL ERGÄNZEN UND MIT OPENREFINE ANREICHERN

...

CANCELED: AUS EXCEL MARC-DATEN ERSTELLEN UND IN ALMA LADEN

...

NACHLASSDATENBANK

- Korrespondenzen wurden in Datenbank erfasst
- Datenbank nicht mehr benutzbar
- Excel-Export der Datenbank verfügbar

BISHERIGE AKTIVITÄTEN

- Auflösung der Namen aus den Schlüsseln
- Erstellung einer Datei pro Nachlass

IN ARBEIT

- Anreicherung von Personen und Orten mit der GND-ID
- Erstellung von MARC-Daten
  - Ein Datensatz pro Konvolut, d. h. pro Absender
  - Konformität mit RNAB

WERKZEUGE

PYTHON

WARUM PYTHON?

- Open Source
- Verfügbarkeit von Materialien zum Selbststudium
- Sehr umfangreiche Standard Library
- Drittpakete für jeden erdenklichen Anwendungsfall über `PyPI` und `pip`
- extrem praktische Methoden zur Textmanipulation

STANDARD-LIBRARY

- csv
- re

PYMARC

- https://github.com/edsu/pymarc
- Objektorientiertes Interface zur Manipulation von MARC-Daten

```
>>> title = record["245"]["a"]
>>> print(title)
"Automate the boring stuff with Python"
```

PYMARC_HELPERS

https://github.com/schuach/pymarc_helpers

Bequemlichkeitsfunktionen für pymarc, v. A. für die Aufbereitung von Verlagsmetadaten

- lesen/ausgeben von MARC-Batch-Dateien in verschiedenen Formaten
- einfache Feldstatistik
- oft benötigte Operationen
  - Entfernen von ISBD-Interpunktion
  - Einfügen von Nichtsortierzeichen
  - relator terms in Codes umwandeln
  - etc.

PYMARC_HELPERS: BEISPIEL

MARC-Batch in MARC-XML schreiben:

```
filename = "output.xml"
writer = pymarc.XMLWriter(open(filename, "wb"))
for record in reclist:
    writer.write(record)
    writer.close()
```

Mit `pymarc_helpers`

```
pymarc_helpers.write_to_file(reclist, "output", form="xml")
```

OPENREFINE

http://openrefine.org/

Freie, sehr mächtige Software Datenaggregation und Datenaufbereitung

- Ursprünglich von Google als "GoogleRefine" entwickelt und später als Open Source veröffentlicht
- Wird zur Anreichern der Daten mit GND-IDs verwendet

EMACS + ORG-MODE

https://www.gnu.org/software/emacs/
http://orgmode.org

- Verfassen von Code und Dokumentation in einer Datei (mit org-babel)
- Export des Scripts
- Export der Dokumentation

IN EMACS

```
**** DONE Beschreibstoff
     CLOSED: [2018-11-29 Do 17:05]
     :LOGBOOK:
     - State "DONE"       from "TODO"    [2018-11-29 Do 17:05]
     :END:
     Für Books gibt es in 007 keinen Code für physical medium,
     daher wird der Text unverändert übernommen. Umso besser,
     dann muss nichts geprüft werden.
     #+NAME: beschreibstoff
     #+BEGIN_SRC python
       beschreibstoff = row["Beschreibstoff"]
       rec.add_ordered_field(
           Field(
               tag = "340",
               indicators = [" ", " "],
               subfields = ["a", beschreibstoff])
```

IN DER EXPORTIERTEN DOKUMENTATION

```
rec.leader = "00000ntm#a22000005c#4500"
```

1. DONE Prüfen, ob Alma beim Import Datensatzlänge etc. einträgt.

2.4.5 DONE Beschreibstoff

Für Books gibt es in 007 keinen Code für physical medium, daher wird der Text unverändert übernommen. Umso besser, dann muss nichts geprüft werden.

```
beschreibstoff = row["Beschreibstoff"].strip()
rec.add_ordered_field(
    Field(
        tag = "340",
        indicators = [" ", " "],
        subfields = ["a", beschreibstoff])
)
```

LESSONS LEARNED

- Genauere Datenkonsistenzprüfungen im Vorfeld
- Sich umhören, was schon getan wurde und ob es vielleicht noch Daten in irgendwelchen "Schubladen" gibt

DANKE FÜR IHRE AUFMERKSAMKEIT!

FRAGEN?

Stefan Schuh

Universitätsbibliothek Graz

mailto:stefan.schuh@uni-graz.at
https://github.com/schuach