

Themenbereich 8: Daten des Universitätsarchivs

Robert Noack, Stefan Schaub, Ramon Bernert

6. Juli 2015

- 1 Einleitung
- 2 Strukturierung
- 3 Quelldatenfehler
- 4 **Erweiterte Struktur**

Datenherkunft

- Daten über zurückgelassene Dokumente von Studenten
- stammen aus dem Universitätsarchiv
- Reichen bis ins 19. Jhr. zurück
- liegen in xls-Format vor
- Tabelle mit 41 Zeilen
- Informationen zu mehreren Studenten in einer Zelle

Vorgehen

- Zellen aus xls-Datei in Textdatei kopiert
- Python-Script analysiert diese Datei
- Ordnet zunächst die Daten anhand einfacher Muster den einzelnen Studenten zu
- Danach detaillierte Analyse der Zeichenkette pro Student
- Ausgabeformat: JSON

Struktur der Ausgangsdaten

3 Informationsbereiche

- ① Namensbereich
- ② persönliche Informationen
- ③ Dokumente

Beispiele

- Namensbereich: persönliche Informationen-- Dokumente.-
- Namensbereich: Dokumente.-

Namensbereich

Nachname, Vorname₁ Vorname₂ ... Vorname_n:

Beispiel

Ahnemüller, Gottlob Wilhelm:

Struktur

```
{  
  "surname": "Ahnemüller",  
  "prenome": "Gottlob Wilhelm",  
}
```

persönliche Informationen

: akademischer Titel, Geburtsort Geburtsjahr--

Beispiele

- stud. oecon., geb. Kiel 5.9.1897--
- geb. 5.7.1860 in Zöblitz--
- aus Saalfeld--

Struktur

```
"academic title": "stud. oecon.",  
"birthplace": "Kiel",  
"birthdate": "5.9.1897",  
"additional_information_of_birthplace": "null",
```

Dokumente

-- Dokument₁, Dokument₂, ..., Dokument_n.-

Beispiel

- Entlastungsschein Universitätsbibliothek 1910, Abgangszeugnis Universität Leipzig 1910.-
- Abgangszeugnis Universitätsgericht Leipzig 1842.-
- G, M, väterliche/vormundschaftliche Studiengenehmigung 1872.-

Dokumenttypen

- Zeugnis
- Buch
- Schein
- Protokoll
- Genehmigung
- Zuweisung
- Diplom
- Quittung

Dokumente

- statische Methode
Certificate.getCertificates(value) erzeugt aus String entsprechende Certificate-Instanzen
- Zerlegt String anhand von Komma
- Regex `.*(\d{4}([/]\d{2}))*.*` sucht nach Jahreszahl
- Jedes Wort wird an `Location.getLocation(city)` übergeben um Ort zu finden
- `Location.getYear()` gibt Jahreszahl zurück

Dokumente

Struktur

```
"certificate": [{  
  "source": "Reifezeugnis 1866",  
  "name": "Reifezeugnis",  
  "type": "Zeugnis",  
  "year": "1866"  
}], {  
  "source": "Abgangszeugnis Universität Leipzig 1873",  
  "name": "Abgangszeugnis Universität Leipzig",  
  "location": {  
    "name": "Leipzig",  
  },  
  "type": "Zeugnis",  
  "year": "1873" } ]
```

Gewünschte Struktur

Beispiel

Abb, Edmund: geb. Trennfurt, Bez. Obernberg 8.6.1878--
Prüfungszeugnis 1902.-

Struktur

```
{  
  "surname": "Abb",  
  "prename": "Edmund",  
  "academic title": "null",  
  "birthplace": "Trennfurt",  
  "birthdate": "8.6.1878",  
  "additional_information_of_birthplace": "Bez. Obernberg",  
  "certificate": [{  
    "source": "Prüfungszeugnis 1902",  
    "name": "Prüfungszeugnis",  
    "type": "Zeugnis",  
    "year": "1902"} ]  
  "object_under_investigation": "Abb, Edmund: geb. Trennfurt,  
  Bez. Obernberg 8.6.1878-- Prüfungszeugnis 1902"  
}
```

Fehler in den Ausgangsdaten

- Zusätzlicher Mädchenname

Beispiel

von Garnier, Katharina: geb. Möwes, stud. philol., geb. in Berlin--

- Doppelter Herkunftsort

Beispiel

Golembiewski, Alexander: stud. oecon., aus Warschau, aus Kowal--

Fehler in den Ausgangsdaten

- Bestehende Anmerkungen

Beispiele

- Pause, Erwin Adelbert (Albin?):
- Sackellar (ios), Polyvius:
- Rabinowicz, Heinrich: (aus Warschau?)--

- Falsche Struktur

Beispiel

Schotsch, Gustav: aus Clausenburg: Abgangszeugnis
Universitätsgericht Leipzig 1869

Fehler in den Ausgangsdaten

- Fehlerhafter Inhalt

Beispiele

- 1von Adamski, Josef: aus Warschau--
- Jäneke, Johann Martin Eduard: aus Glauchau--
Matrikelscheine 1863, väterliche/vormu\"\\n
- Mann, Conrad: stud. pharm., \"Servestanuns\"--
Matrikelscheine 1881

Fehler in den Ausgangsdaten

- Abgeschnittene Bereiche

Beispiel

- R\n
- Schrag, Emil Richard:\n
- Legitimationskarte, Kollegienbuch, Entlastungsschein
Universitätsbibliothek 1900, Abgangszeugnis Universität
Leipzig 1900
- Seyler, Georg: aus Harthau b. Bischofswerda-- Zuweisung
zur Immat\n

Geotagging

- Zur Identifizierung von Orten: geonames.org
- Bietet einfache API
- gibt JSON zurück
- enthält geonameID, Längen- und Breitengrad
- Aufruf des Webservices erfolgt in Klasse Location
- Verwendung zum Parsen des Geburts- und Dokumentenorts

Geotagging

Aufruf

```
http://api.geonames.org/searchJSON?q=Leipzig&username=...
```

Ausgabe

```
{ "totalResultsCount": 208,  
  "geonames": [{  
    "lng": "12.37129",  
    "name": "Leipzig",  
    "geonameId": 2879139,  
    "lat": "51.33962"  
  }]  
}
```

Geotagging

- statische Methode `Location.getLocation(city)` erzeugt aus String eine Location-Instanz
- Instanzen werden gecached
- `Location.getLng()` gibt Längengrad zurück

Erzeugtes JSON

```
{  "name": "Leipzig",  
    "latitude": "51.33962",  
    "geonameld": 2879139,  
    "longitude": "12.37129",  
    "url":  
    "https://maps.google.de/maps?q=51.33962,12.37129"  }
```

zukünftige Verfahren

- Statistische Analysen für beispielsweise:
 - Aus welchem Jahr sind die meisten Studenten?
 - Geografische Verteilung
 - Häufigsten Dokumentenarten
 - ...

Vielen Dank für Ihre Aufmerksamkeit.
Noch Fragen?