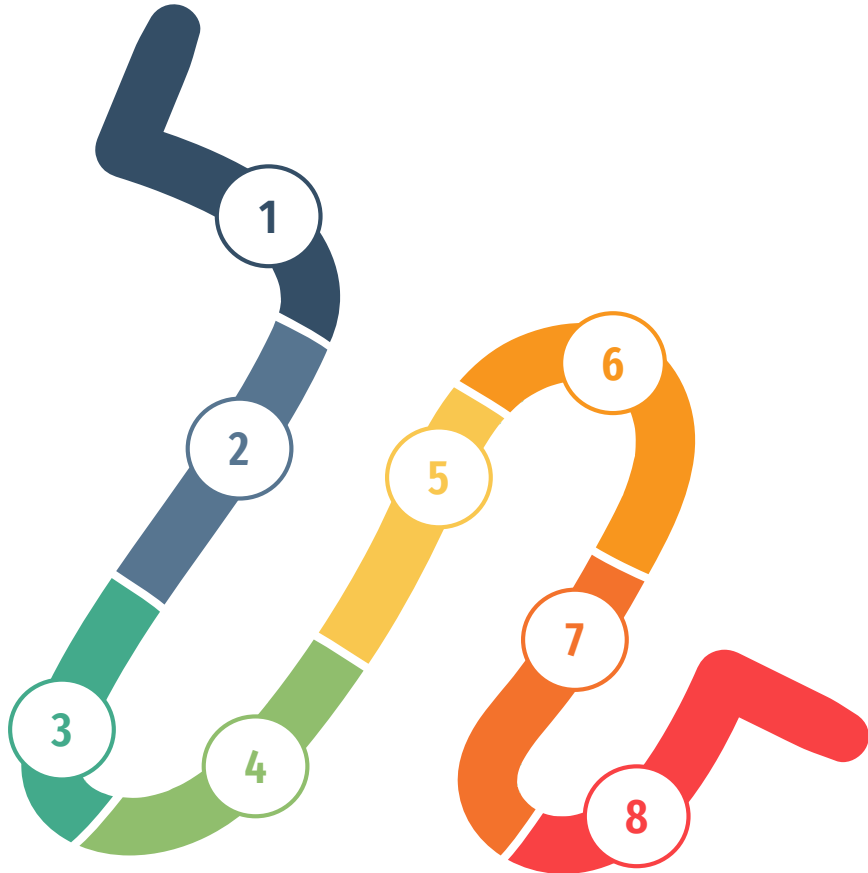


Evaluierung von annif in der TIB – ein Werkstattbericht

Susanne Arndt, Berrit Genat, Mila Runnwerth
DNB-Online-Workshop:
Erfahrungen und Perspektiven mit dem Toolkit annif
3. Dezember 2020

Oder: Der Weg ist das Ziel



- | | | | |
|---|---------------------------------|---|-------------------------|
| 1 | Vokabular transformieren | 5 | Goldstandard definieren |
| 2 | Datenkollektion zusammenstellen | 6 | Testmenge extrahieren |
| 3 | Datendump herunterladen | 7 | annif konfigurieren |
| 4 | Daten transformieren | 8 | Ergebnisse analysieren |

1

Vokabular transformieren



Dauer (A): 1 min bis 1 Tag

Dauer (B): 1 h bis 1 Tag



https://github.com/runwerth/annif_automated_indexting_at_tib/wiki/Vokabular-transformieren



	Variante A: Lookup-Table (.tsv)	Variante B: SKOS-Datei (.ttl)
Ziel	2 Spalten <ul style="list-style-type: none">• Spalte 1: 1 URI pro Klasse• Spalte 2: Bezeichner der Klasse	SKOS-Modellierung des eigenen Vokabulars
Input	HTML, Tabelle (Idealfall), XML, RDF	HTML, Tabelle (Idealfall), XML, RDF (nicht in SKOS)
Tool	z.B. OpenRefine, Protégé	z.B. OpenRefine, Protégé
Teilaufgaben	<ul style="list-style-type: none">• ggf. URI pro Vokabularterm erstellen• Infos aus einer RDF-Modellierung raussparqln	<ul style="list-style-type: none">• eigene Daten durch die SKOS-Brille sehen• ggf. URI pro Vokabularterm erstellen• ggf. Beziehungen explizit machen• Tripel anlegen• Turtle-Datei anlegen
Voraussetzungen	GREL oder Python	GREL oder Python, Basisverständnis über SKOS-Elemente und RDF-Tripel

1

Vokabular transformieren



Input

	A	B	C	D	
1	Systemstelle		Link	Beschreibung	Hi
2	mat 1		x	Allgemeines	
3	mat 1	mat 1.4	x	Didaktik der Mathematik. Programmierer und rechnergestützter Mathematikunterricht	
4	mat 1	mat 1.5	x	Beruf. Ausbildung. Nachwuchs	



1

Vokabular transformieren



Variante A - Output

```
1 URI Beschreibung
2 <http://unicorn.fly/tib_lok_sys#mat1> Allgemeines
3 <http://unicorn.fly/tib_lok_sys#mat1_4> Didaktik der Mathematik.
  Programmierer und rechnergestützter Mathematikunterricht
4 <http://unicorn.fly/tib_lok_sys#mat1_5> Beruf. Ausbildung. Nachwuchs
```

Variante B - Output

```
1 @prefix skos: <http://www.w3.org/2004/02/skos/core#> .
2 @base <http://unicorn.fly/tib_lok_sy> .
3 @prefix math: <http://unicorn.fly/tib_lok_sy#> .
4
5 math: a owl:Ontology ; owl:imports skos: .
6
7 <http://unicorn.fly/tib_lok_sys#mat1> skos:prefLabel "Allgemeines"@de .
8 <http://unicorn.fly/tib_lok_sys#mat1> a skos:Concept .
9 <http://unicorn.fly/tib_lok_sys#mat1_4> skos:prefLabel "Didaktik der Mathematik. Programmierer und
  rechnergestützter Mathematikunterricht"@de .
10 <http://unicorn.fly/tib_lok_sys#mat1_4> a skos:Concept .
11 <http://unicorn.fly/tib_lok_sys#mat1> skos:narrower <http://unicorn.fly/tib_lok_sys#mat1_4> .
12 <http://unicorn.fly/tib_lok_sys#mat1_5> skos:prefLabel "Beruf. Ausbildung. Nachwuchs"@de .
13 <http://unicorn.fly/tib_lok_sys#mat1_5> a skos:Concept .
14 <http://unicorn.fly/tib_lok_sys#mat1> skos:narrower <http://unicorn.fly/tib_lok_sys#mat1_5> .
```



2

Datenkollektion
zusammenstellen



https://github.com/runwerth/annif_automated_indexing_at_tib/wiki/Definition-des-Daten-Dumps



Dauer: ~ 10 Minuten

Portal

OAI-Schnittstelle

```
language%5D%5B1%5D=de
supplierPrefix%5D%5B0%5D=tibkat
locationCode%3A%28L%20mat%2A%20OR%20LB%20mat%2A%29
```

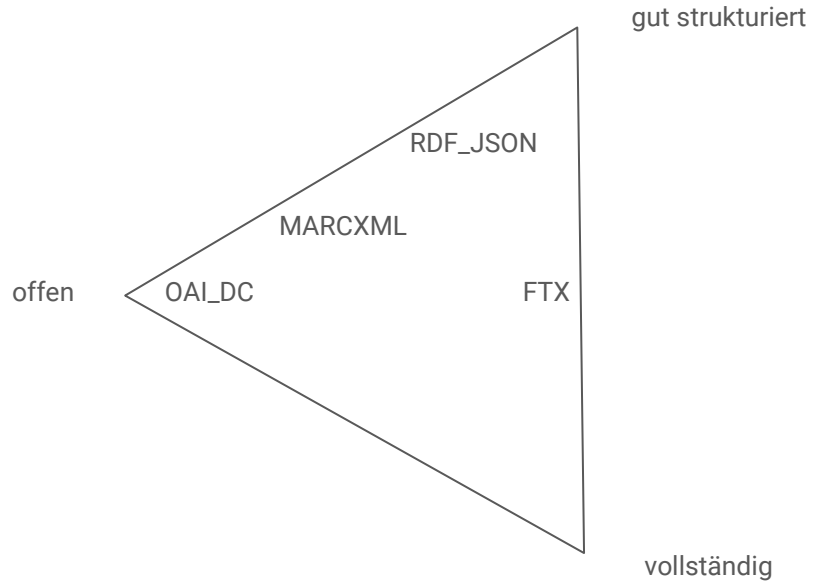
```
set=collection~tibkat
language:de
locationCode%3A%28L%20mat%2A%20OR%20LB%20mat%2A%29
```

2

Datenkollektion
zusammenstellen



Metadatenformat



https://github.com/runnwerth/annif_automated_indexing_at_tib/wiki/Definition-des-Daten-Dumps



3

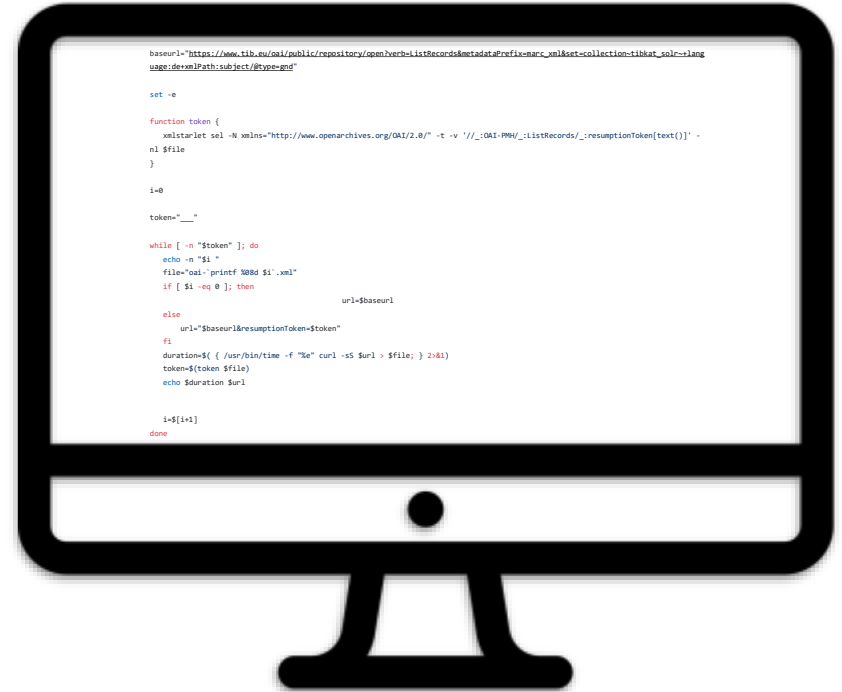
Datendump
herunterladen



- Chunks mit 100 Metadaten­sätzen
- je nach Metadatenformat zwischen 2 - 20 Sekunden pro Chunk
- größter Korpus lag bei etwa 1.270.000 Datensätzen



Dauer: 5 - 50 Stunden

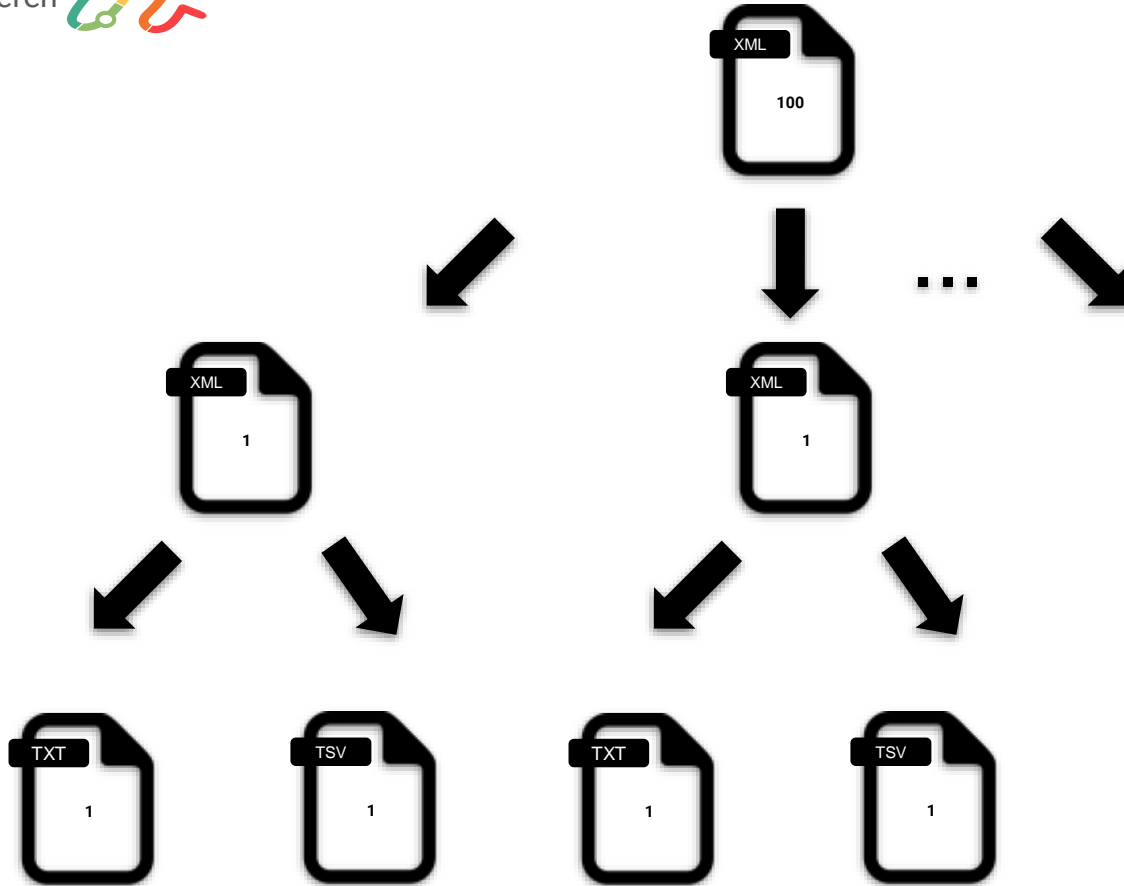


4

Daten transformieren



https://github.com/runwerth/annif_automated_indexing_at_tib/wiki/%C3%99Obersetzen-des-Dumps-in-die-annif-Syntax

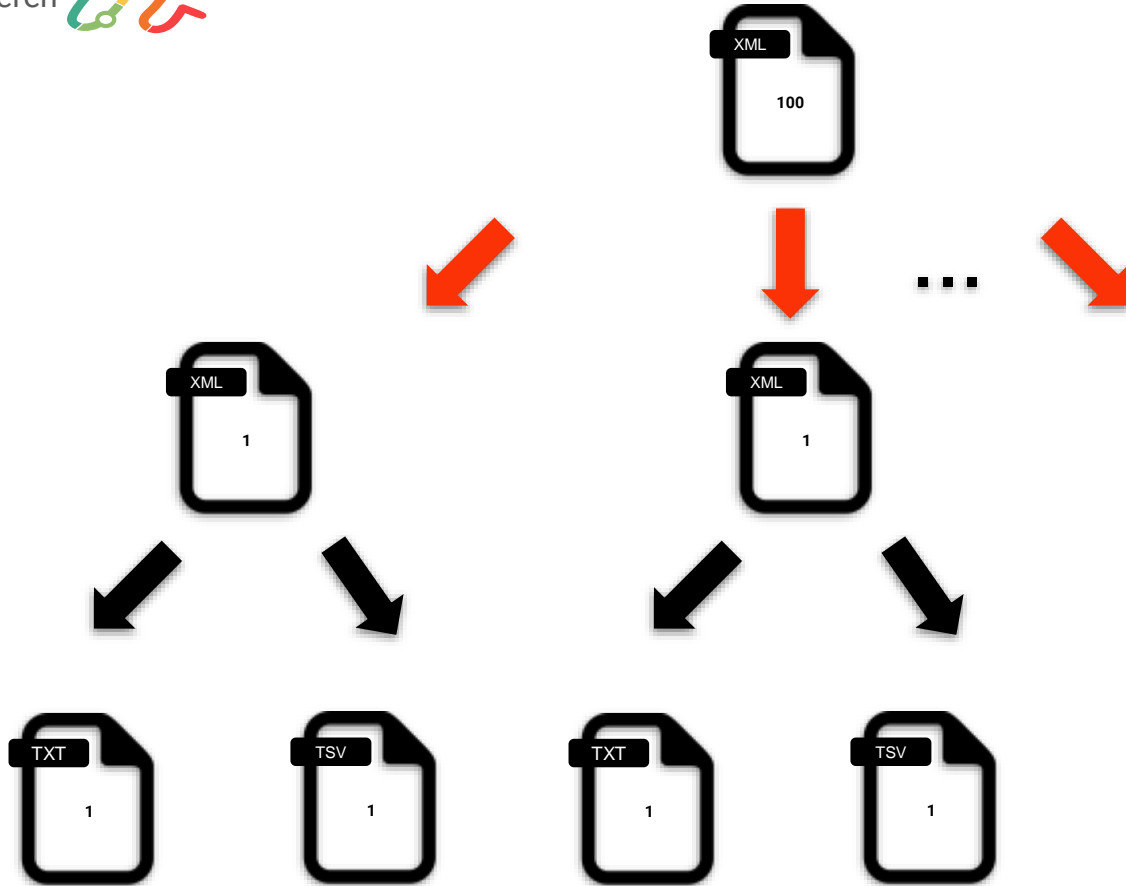


4

Daten transformieren



https://github.com/runwerth/annif_automated_indexing_at_tib/wiki/%C3%99Obersetzen-des-Dumps-in-die-annif-Syntax



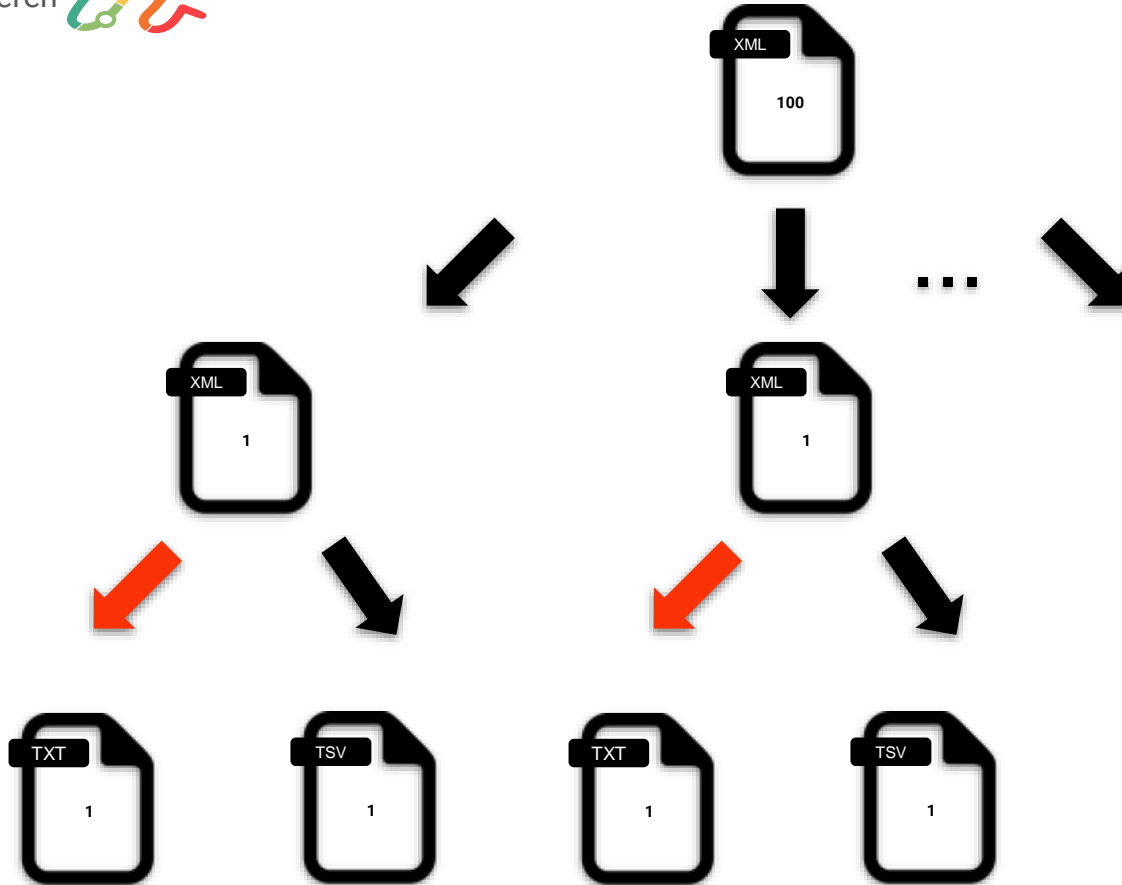
Dauer: 1 - 5 Stunden

4

Daten transformieren



https://github.com/runwertly/annif_automated_indexing_at_tib/wiki/%C3%99Obersetzen-des-Dumps-in-die-annif-Syntax



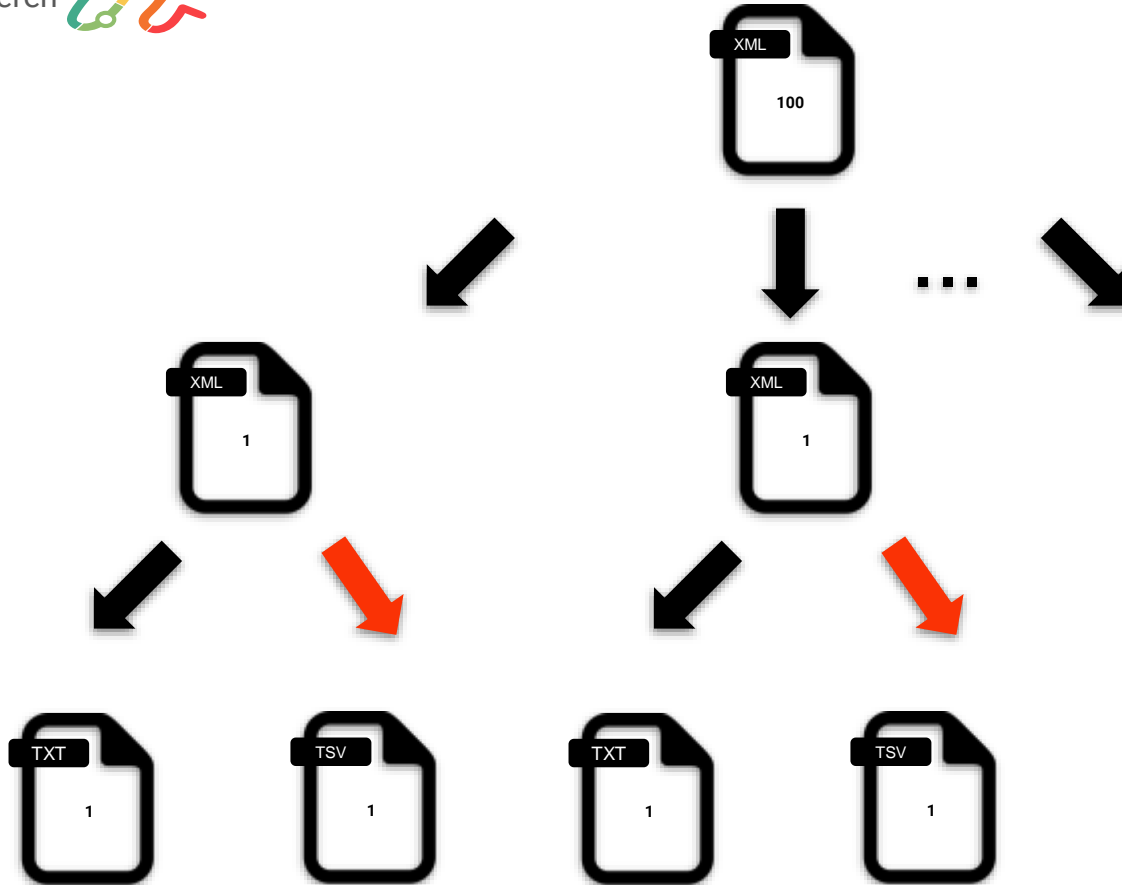
Dauer: 1 - 5 Stunden

4

Daten transformieren



https://github.com/runwerth/annif_automated_indexing_at_tib/wiki/%C3%99Obersetzen-des-Dumps-in-die-annif-Syntax



Dauer: 1 - 5 Stunden



Daten
transformieren



Der fertige Input für annif sieht dann pro Titeldatensatz so aus:

TXT: (Semantisch verwertbare) Metadaten

```
Führer auf den deutschen Schifffahrtstraßen  
Preußen, Ministerium der Öffentlichen Arbeiten  
Wasserbau  
Schifffahrt  
Deutschland
```

TSV: Ausgewählte Erschließungsdaten

```
<http://uri.gbv.de/terminology/bk/55.86> Schiffsverkehr, Schifffahrt  
<http://uri.gbv.de/terminology/bk/56.30> Wasserbau  
<http://uri.gbv.de/terminology/bk/15.40> Deutsche Geschichte: Allgemeines
```





MARCXML



```
xmlstarlet sel -N marcxml="http://www.loc.gov/MARC21/slim" -t -v  
'//marcxml:subfield[@code="2"][text()="bk"]/following-  
sibling::marcxml:subfield[@code="a"]' -nl tibkat00113689.xml
```

```
<marcxml:datafield ind1=" " ind2=" " tag="084">  
  <marcxml:subfield code="2">bk</marcxml:subfield>  
  <marcxml:subfield code="a">55.80</marcxml:subfield>  
</marcxml:datafield>  
<marcxml:datafield ind1=" " ind2=" " tag="084">  
  <marcxml:subfield code="2">bk</marcxml:subfield>  
  <marcxml:subfield code="a">50.01</marcxml:subfield>  
</marcxml:datafield>  
<marcxml:datafield ind1=" " ind2=" " tag="084">  
  <marcxml:subfield code="2">ssg</marcxml:subfield>  
  <marcxml:subfield code="a">19,2</marcxml:subfield>  
</marcxml:datafield>  
<marcxml:datafield ind1=" " ind2=" " tag="084">  
  <marcxml:subfield code="2">rvk</marcxml:subfield>  
  <marcxml:subfield code="a">ZG 8930</marcxml:subfield>  
</marcxml:datafield>  
<marcxml:datafield ind1=" " ind2=" " tag="084">  
  <marcxml:subfield code="2">loc</marcxml:subfield>  
  <marcxml:subfield code="a">T183.G32</marcxml:subfield>  
</marcxml:datafield>
```





```
xmlstarlet sel -T -t -m "//classification[@classificationName='bk']" -  
v 'code[text()]' -o $('t' -v 'entries/entry[text()]' -nl
```

```
<classificationInfo>  
  <classifications>  
    <classification classificationID="106410458"  
classificationName="bk">  
      <code>42.13</code>  
      <entries>  
        <entry>Molekularbiologie</entry>  
      </entries>  
    </classification>  
    <classification classificationID="106410148"  
classificationName="bk">  
      <code>42.01</code>  
      <entries>  
        <entry>Geschichte der Biologie</entry>  
      </entries>  
    </classification>  
  <subjects>  
    <subject id="4071722-7" type="gnd">  
      <dc:subject xml:lang="de">Gentechnologie</dc:subject>  
    </subject>  
    <subject type="mesh">  
      <dc:subject xml:lang="de">Eugenics</dc:subject>  
    </subject>  
  </subjects>  
</classificationInfo>
```



5

Goldstandard
definieren



Dauer: ~ 10 Stunden



- Intellektueller Aufwand:
 - Repräsentanz des Gesamtkorpus ermitteln,
 - Qualität von hochwertiger Erschließung definieren,
 - passende Datensätze finden.
- Empfohlen: 10 - 15% vom Gesamtkorpus
- Das bedeutete in unserem größten Korpus von etwa 1.270.000 Metadatenätzen: ~12.700 Metadatenätze für den Goldstandard
- Wir haben diesem Fall geschummelt und nur 130 Metadatenätze ausgesucht.

https://github.com/runnwerth/annif_automated_indexing_at_tib/wiki/Anlegen-eines-Goldstandards



The screenshot shows a library catalog entry for the book "Mathematische Methoden der Physik" by Hermann Goldhamer. The record includes the title, author, ISBN (978-3-7089-0844-9), and a gold seal icon indicating a gold standard. The seal is circular and gold-colored with the text "GOLD STANDARD" and "BIBLIOTHEK" around the perimeter. The catalog entry also lists the publisher (Springer), the year (2017), and the number of pages (327). The record is displayed in a clean, modern interface with a search bar at the top and navigation options on the right.

6

Testmenge
extrahieren



- Empfohlen: 10% vom Gesamtkorpus
- Auch hier haben wir gemogelt und nur eine handvoll Datensätze isoliert, um sie im WebGUI zu testen.
- Für uns erst interessant, wenn die Ergebnisse in Richtung Anwendbarkeit deuten.



Dauer: ~ 5 Minuten



7

annif konfigurieren



```
[TIBKAT_LokSysMath_de_OP]
name=TIBKAT_LokSysMath_de_OP
language=de
backend=omikuji
analyzer=snowball(german)
vocab=TIBLokSysMath
```

```
# [FIDmove_BK_de_maii]
# name=FIDmove_BK_de_maii
# language=de
# backend=maui
#
endpoint=http://mauiserver:8080/mauiserver/
# tagger=FIDmove_BK_de_maii
# vocab=bk
# limit=1000
```

Ablauf:
myAnnifProject.sh



Dauer: < 10 Minuten (inkl. annif laufen lassen)



Konfiguration:
projects.cfg

```
#!/bin/bash

annif_project="TIBKAT_LokSysMath_de_OP"

#Load local classification
echo "Start loading local classification"
annif loadvoc $annif_project vocab/TIBLokSysMath.tsv
echo "Local classification loaded"
#Train loaded vocabulary
echo "Start training"
annif train $annif_project training/TIBIndex/TIBKAT_LokSysMath_de/
echo "Local classification trained"

#Evaluate against gold standard
annif eval $annif_project goldstandard/TIBIndex/TIBKAT_LokSysMath_de/

#Run Web GUI
echo "Start Web GUI"
gunicorn --bind 0.0.0.0:5000 "annif:create_app()"
#annif run
```





Ergebnisse analysieren



Evaluierungsfeature von annif



FIDmove-Index (TIBKAT-Anteil),
Basisklassifikation, deutsch, Maui

TIB-Index, TIBKAT,
Lokalsystematik, deutsch, Omikuji

Precision (doc avg):	0.6076112412177986
Recall (doc avg):	0.6586065573770491
F1 score (doc avg):	0.5950772360608426
True positives:	147
False positives:	119
False negatives:	81
Documents evaluated:	122



Precision (doc avg):	0.08139534883720931
Recall (doc avg):	0.813953488372093
F1 score (doc avg):	0.14799154334038056
True positives:	105
False positives:	1185
False negatives:	2
Documents evaluated:	129





Ergebnisse analysieren



Stichproben mit dem WebGUI



https://github.com/runnwerth/annif_automated_indexing_at_tib/wiki/Ergebnisse



- 1 Vokabular transformieren
- 2 Datenkollektion zusammenstellen
- 3 Datendump herunterladen
- 4 Daten transformieren
- 5 Goldstandard definieren
- 6 Testmenge extrahieren
- 7 annif konfigurieren
- 8 Ergebnisse analysieren



Best Case:



~ 18h

Worst Case:



~ 85h

Wie kann es weitergehen?



Automatische Fachzuordnung von Dokumentmetadaten anhand von Erschließungsinformationen LinSearch nachbauen

Stufe 1: Zuordnung nach vorhandenen Klassifikationselementen gemäß einer Konkordanz (Mapping)



Korpus für

Stufe 2: Automatische Fachzuordnung nach linguistischer Indexierung mit

annif



https://github.com/runnwerth/annif_automated_ind_exing_at_tib/wiki/Weitere-Ideen



https://github.com/runnwerth/annif_automated_ind_exing_at_tib/wiki/Versuch-7:-TIBKAT.-LinSearch.-deutsch.-Omikuji

Wie kann es weitergehen?



Benchmark-Matrix mit einem vielversprechenden Korpus.

Lernverfahren	Parametrisierung	Ergebnisse
TF-IDF		
FastText		
Omikuji (Parabel)		
...		



Wie kann es weitergehen?



Datenaufbereitungsprozess benutzungsfreundlicher gestalten (LowFi)

Vok.

Wählen Sie Ihre Systematik

<input type="checkbox"/>	GND	<input type="checkbox"/>	BK
<input type="checkbox"/>	FIDmoveVoc	<input checked="" type="checkbox"/>	RVK
<input type="checkbox"/>	STW		



Wie kann es weitergehen?



Datenaufbereitungsprozess benutzungsfreundlicher gestalten (LowFi)

Vok. **Form.**

Wählen Sie Ihr gewünschtes Datenformat

<input type="checkbox"/>	JSON-LD	<input type="checkbox"/>	MARXML
<input type="checkbox"/>	Dublin Core	<input checked="" type="checkbox"/>	FTX



Machen Sie mit!



https://github.com/runnwerth/annif_automated_indexing_at_tib/wiki/

