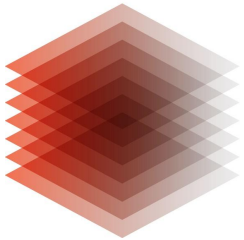


TIB
LEIBNIZ-INFORMATIONSZENTRUM
TECHNIK UND NATURWISSENSCHAFTEN
UNIVERSITÄTSBIBLIOTHEK

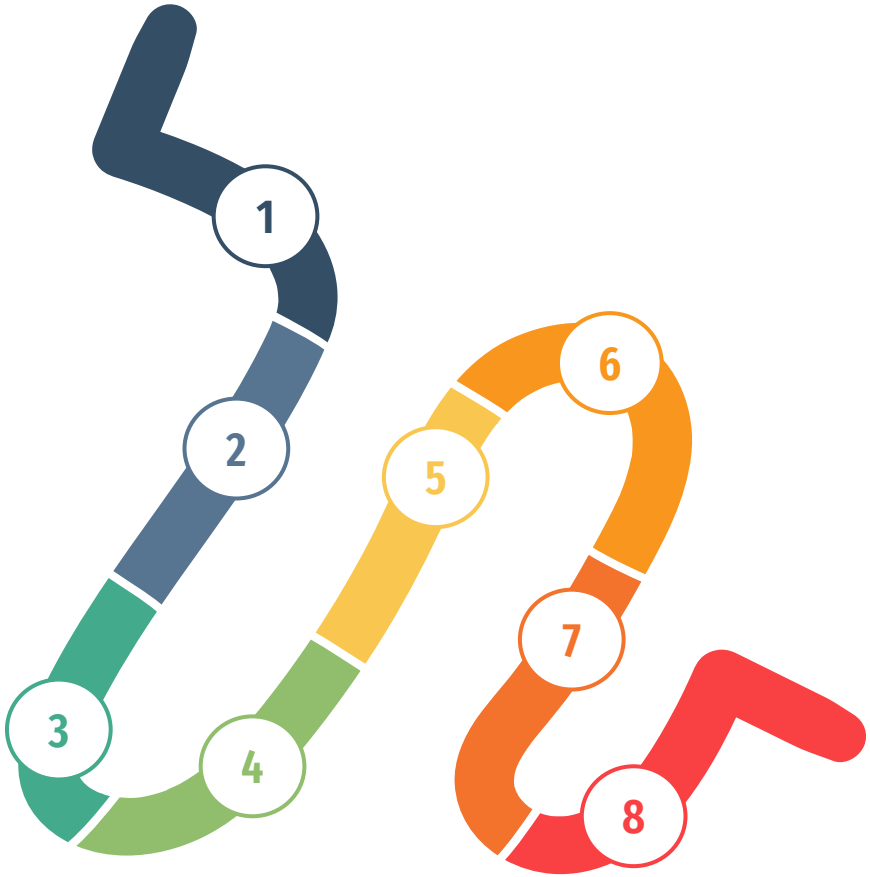
Evaluierung von annif in der TIB – ein Werkstattbericht

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

1 Vokabular transformieren

Dauer (A): 1 min bis 1 Tag
Dauer (B): 1 h bis 1 Tag

https://github.com/tunnwerth/annif_automated_indexing_at_tib/wiki/Vokabular-transformieren

	Variante A: Lookup-Table (.tsv)	Variante B: SKOS-Datei (.ttl)
Ziel	2 Spalten • 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	• ggf. URI pro Vokabularterm erstellen • Infos aus einer RDF-Modellierung raussparqln	• 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

https://github.com/tunnwerth/annif_automated_indexing_at_tib/wiki/Vokabular-transformieren

Input

D5 | Unterhaltungsmathematik (Rätsel, Spiele, u. a.)

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

1 Vokabular transformieren

https://github.com/tunnwerth/annif_automated_indexing_at_tib/wiki/Vokabular-transformieren

Variante A - Output

```
URI Beschreibung
<http://uricorn.fly/tib_lok_sys#mat1>   Allgemeines
<http://uricorn.fly/tib_lok_sys#mat1_4> Didaktik der Mathematik. Programmierter und rechnergestützter Mathematikunterricht
<http://uricorn.fly/tib_lok_sys#mat1_5> Beruf. Ausbildung. Nachwuchs
```

Variante B - Output

```
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@base <http://uricorn.fly/tib_lok_sy> .
@prefix math: <http://uricorn.fly/tib_lok_sy#> .

math: a owl:Ontology ; owl:imports skos: .

<http://uricorn.fly/tib_lok_sys#mat1> skos:prefLabel "Allgemeines"@de .
<http://uricorn.fly/tib_lok_sys#mat1> a skos:Concept .
<http://uricorn.fly/tib_lok_sys#mat1_4> skos:prefLabel "Didaktik der Mathematik. Programmierter und rechnergestützter Mathematikunterricht"@de .
<http://uricorn.fly/tib_lok_sys#mat1_4> a skos:Concept .
<http://uricorn.fly/tib_lok_sys#mat1> skos:narrower <http://uricorn.fly/tib_lok_sys#mat1_4> .
<http://uricorn.fly/tib_lok_sys#mat1_5> skos:prefLabel "Beruf. Ausbildung. Nachwuchs"@de .
<http://uricorn.fly/tib_lok_sys#mat1_5> a skos:Concept .
<http://uricorn.fly/tib_lok_sys#mat1> skos:narrower <http://uricorn.fly/tib_lok_sys#mat1_5> .
```

2 Datenkollektion zusammenstellen

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

Homepage

Die Metadaten zum Abruf über OAI-PMH

Über die OAI-Schnittstelle der TIB können Metadaten in den Formaten OAI Dublin Core, MARC XML oder RDF XML heruntergeladen werden, eine tägliche Aktualisierung ist möglich. Details siehe bitte in der Schnittstellen-Dokumentation.

Es gibt verschiedene Tools, die das Herunterladen unserer Daten über OAI-PMH ermöglichen. Ein Tool mit graphischer Oberfläche ist bespielsweise MarcEdit.

TIBKAT-Daten

Beispiel-Links (Komplettabzug, tagesaktuell)

Dublin Core:	https://www.tib.eu/oai/public/repository/open?verb=ListRecords&metadataPrefix=oai_dc&set=tibkat
MARC XML:	https://www.tib.eu/oai/public/repository/open?verb=ListRecords&metadataPrefix=marc_xml&set=tibkat
RDF XML:	https://www.tib.eu/oai/public/repository/open?verb=ListRecords&metadataPrefix=rdf_xml&set=tibkat
RDF JSON:	https://www.tib.eu/oai/public/repository/open?verb=ListRecords&metadataPrefix=rdf_json&set=tibkat

Metadaten der frei verfügbaren elektronischen Bestände

Beispiel-Links (komplette Daten, tagesaktuell)

Dublin Core:	https://www.tib.eu/oai/public/repository/open?verb=ListRecords&metadataPrefix=oai_dc
MARC XML:	https://www.tib.eu/oai/public/repository/open?verb=ListRecords&metadataPrefix=marc_xml
RDF XML:	https://www.tib.eu/oai/public/repository/open?verb=ListRecords&metadataPrefix=rdf_xml
RDF JSON:	https://www.tib.eu/oai/public/repository/open?verb=ListRecords&metadataPrefix=rdf_json

Metadaten des TIB AV-Portals

Beispiel-Links (tagesaktuell)

Dublin Core:	https://www.tib.eu/oai/public/repository/av-portal?verb=ListRecords&metadataPrefix=oai_dc
MARC XML:	https://www.tib.eu/oai/public/repository/av-portal?verb=ListRecords&metadataPrefix=marc_xml
RDF XML:	https://www.tib.eu/oai/public/repository/av-portal?verb=ListRecords&metadataPrefix=rdf_xml

Die Metadaten als Gesamtabzug (Dump)

TIBKAT-Daten und Metadaten der frei verfügbaren elektronischen Bestände

Die Katalogdaten der TIB und die Metadaten der frei verfügbaren elektronischen Bestände sind komprimiert als Gesamtabzug in den Formaten JSON-LD, N3 und Turtle abrufbar:

JSON-LD:	https://tib.eu/data/rdf/open_jsonld.dump.gz
N3:	https://tib.eu/data/rdf/open_n3.dump.gz
Turtle:	https://tib.eu/data/rdf/open_turtle.dump.gz

RDF-Export von Metadaten des TIB AV-Portals

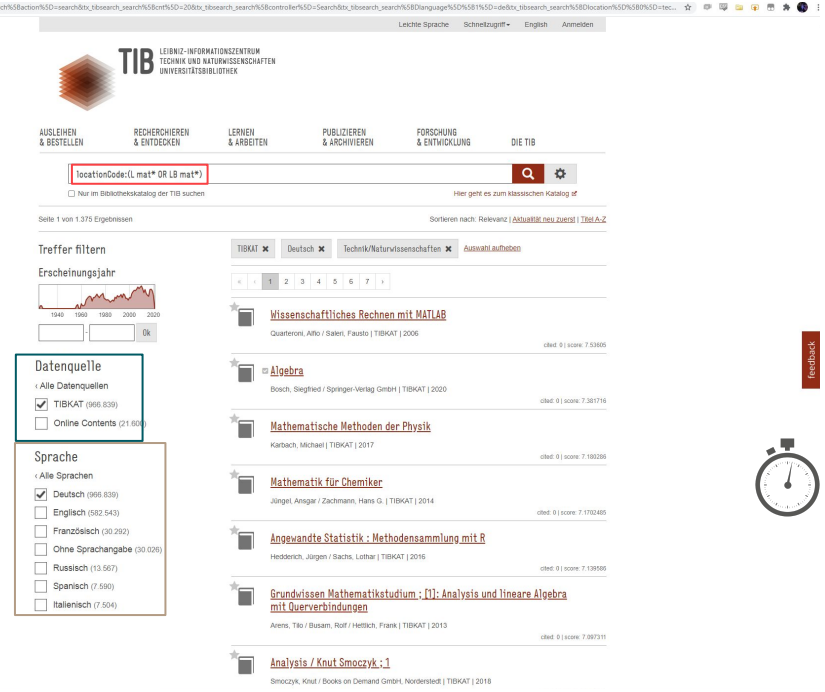
Die die autoritativen sowie zeitbasierten, automatisch generierten Metadaten von Filmen des AV-Portals der TIB sind ebenfalls komprimiert als Gesamtabzug in den Formaten JSON-LD, N3 und Turtle abrufbar. Diese Datensätze finden Sie auf der Download-Seite des AV-Portals der TIB.

https://www.tib.eu/de/die-tib/open-data

OAI-Schnittstelle

https://www.tib.eu/oai/public/repository/open?verb=ListRecords&metadataPrefix=marc_xml&set=collection~tibkat_solr~+language:de+xmlPath:subject/@type=gnd

2 Datenkollektion zusammenstellen



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

Dauer: ~ 10 Minuten

Portal

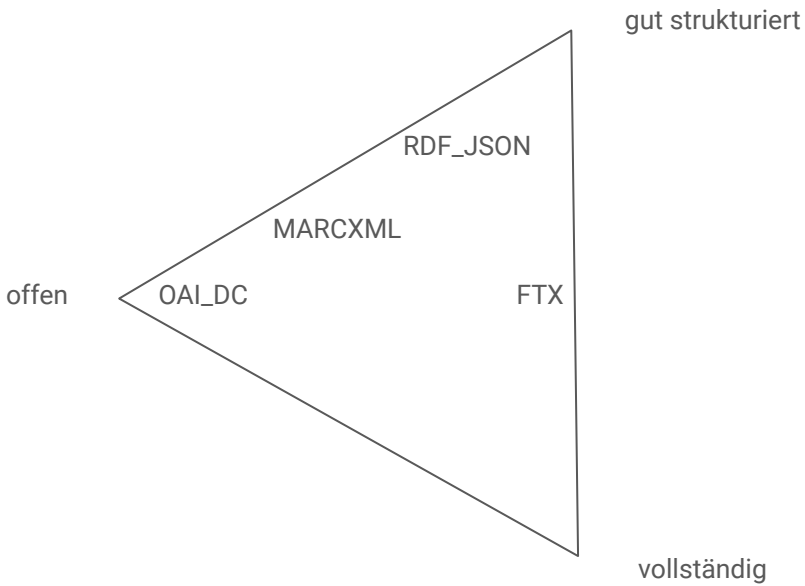
language%5D%5B1%5D=de
supplierPrefix%5D%5B0%5D=tibkat
locationCode%3A%28L%20mat%2A%20OR%20LB%20mat%2A%29

OAI-Schnittstelle

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

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

Metadatenformat

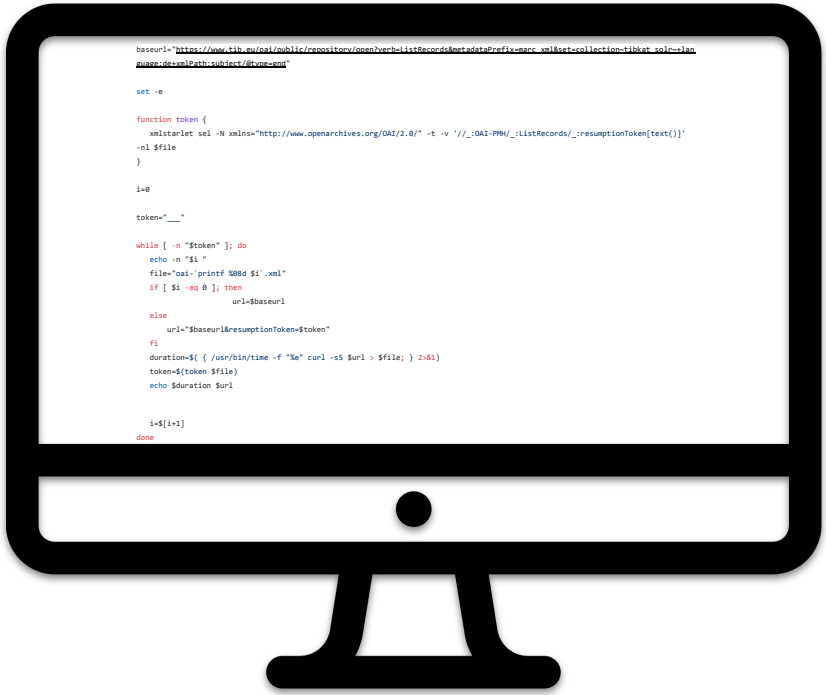


3 Datendump herunterladen

https://github.com/runnwerth/annif_automated_indexing_at_tib/wiki/Daten-Dump-herunterladen

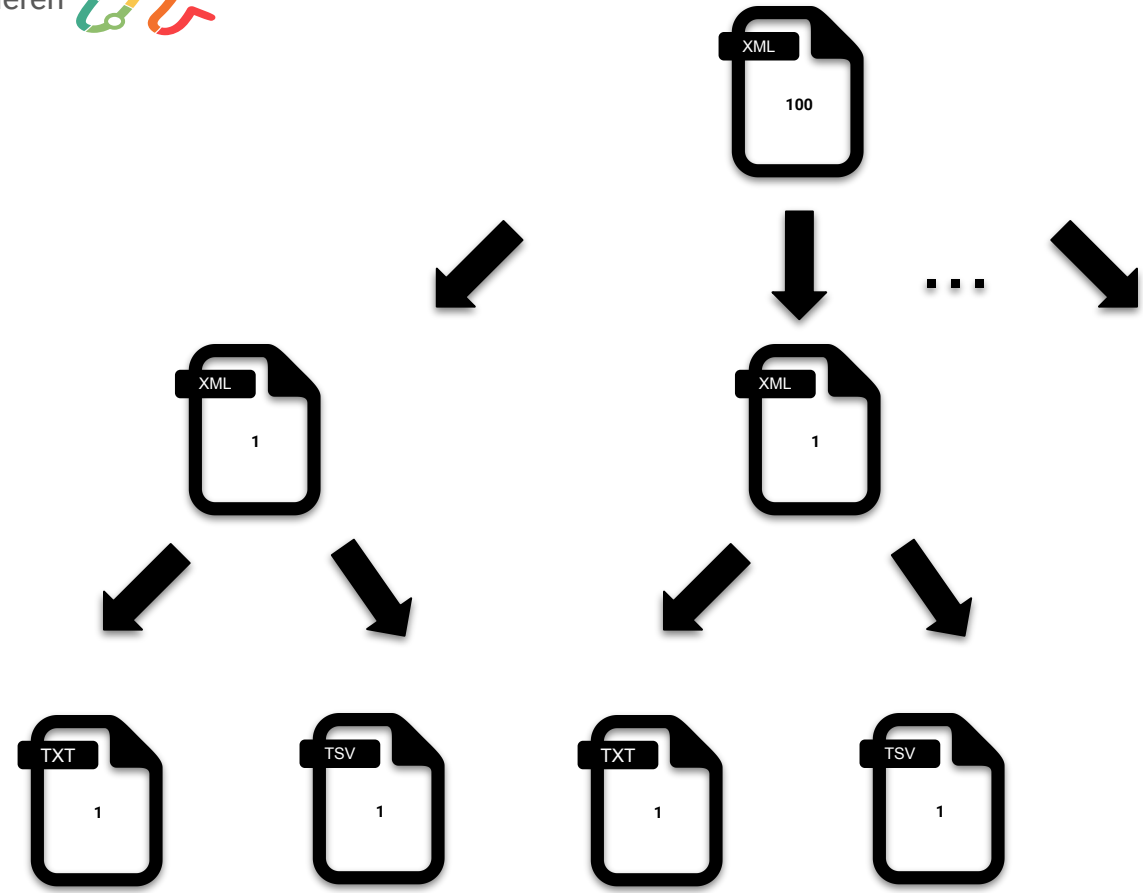
- Chunks mit 100 Metadatensätzen
- je nach Metadatenformat zwischen 2 - 20 Sekunden pro Chunk
- größter Korpus lag bei etwa 1.270.000 Datensätzen

Dauer: 5 - 50 Stunden



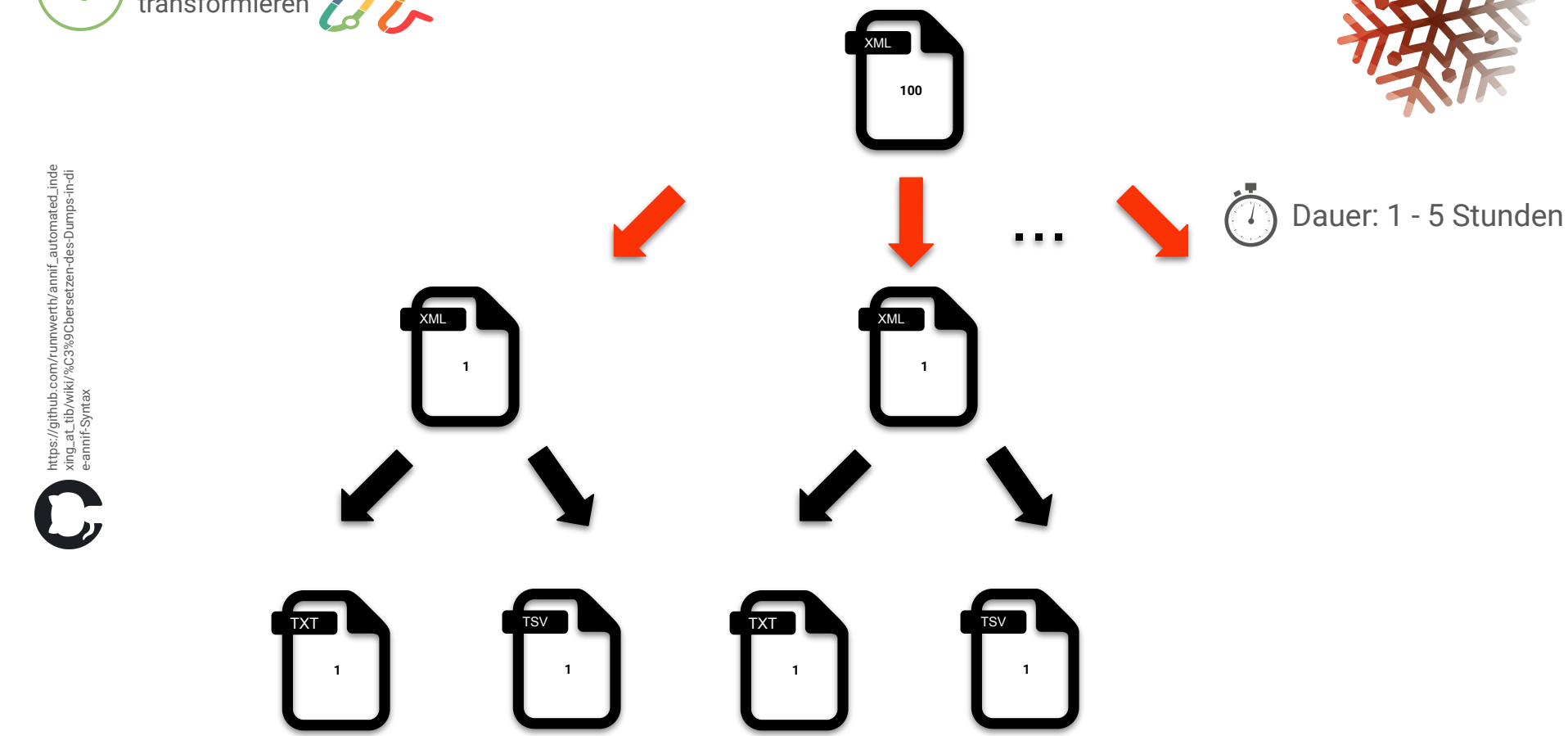
```
baseurl="https://www.tib.eu/oai/public/repository/oai?verb=ListRecords&metadataPrefix=marc_xml&set=collection-tibkat_solr--language:de+xmlPath:subject/@type=gnd"

set -e

function token {
   xmlstarlet sel -N xmlns="http://www.openarchives.org/OAI/2.0/" -t -v '//_:OAI-PMH/_:ListRecords/_:resumptionToken[text()]'
-nl $file
}

i=0

token="___"

while [ -n "$token" ]; do
   echo -n "$i "
   file="oai-`printf %08d $i`.xml"
   if [ $i -eq 0 ]; then
                  url=$baseurl
   else
       url="$baseurl&resumptionToken=$token"
   fi
   duration=$( { /usr/bin/time -f "%e" curl -sS $url > $file; } 2>&1)
   token=$(token $file)
   echo $duration $url


   i=$[i+1]
done
```

4 Daten transformieren



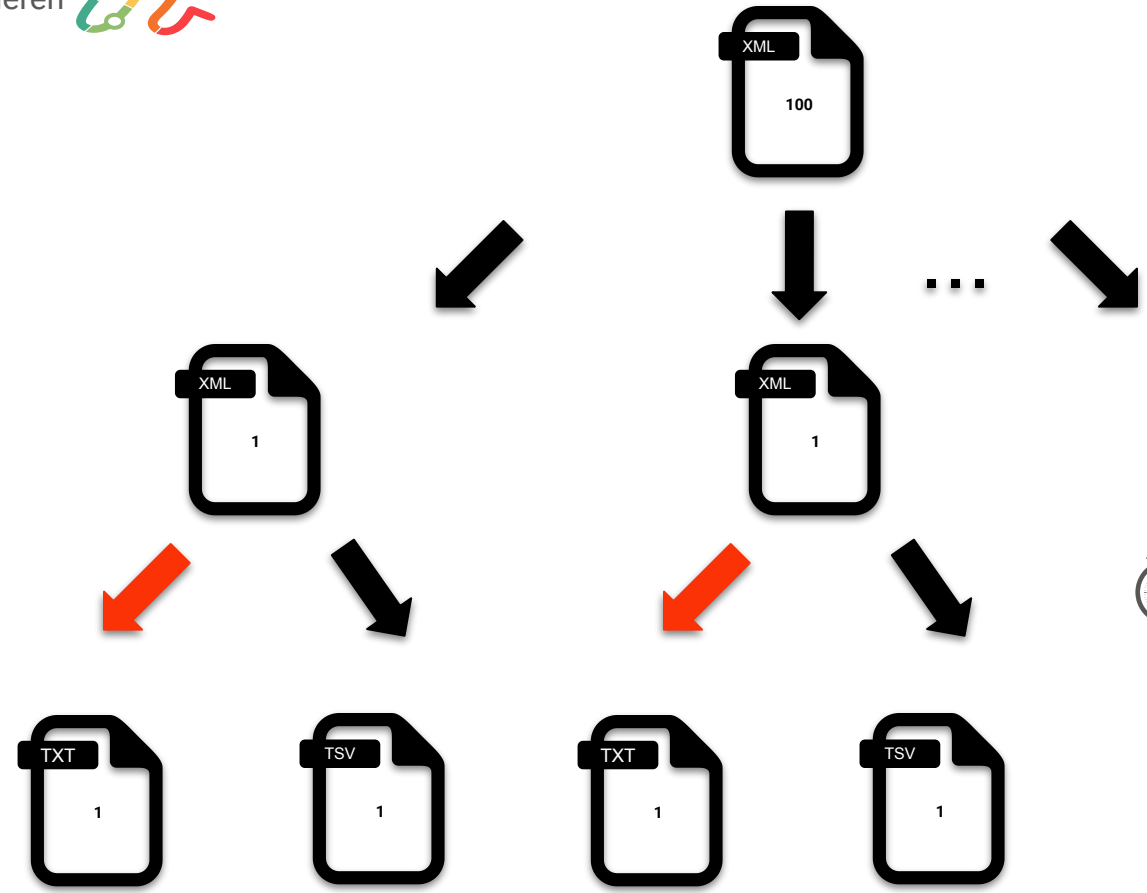
https://github.com/runnwerth/annif_automated_indexing_at_tib/wiki/%C3%9Cbersetzen-des-Dumps-in-die-annif-Syntax

4 Daten transformieren



https://github.com/runnwerth/annif_automated_indexing_at_tib/wiki/%C3%9Cbersetzen-des-Dumps-in-die-annif-Syntax

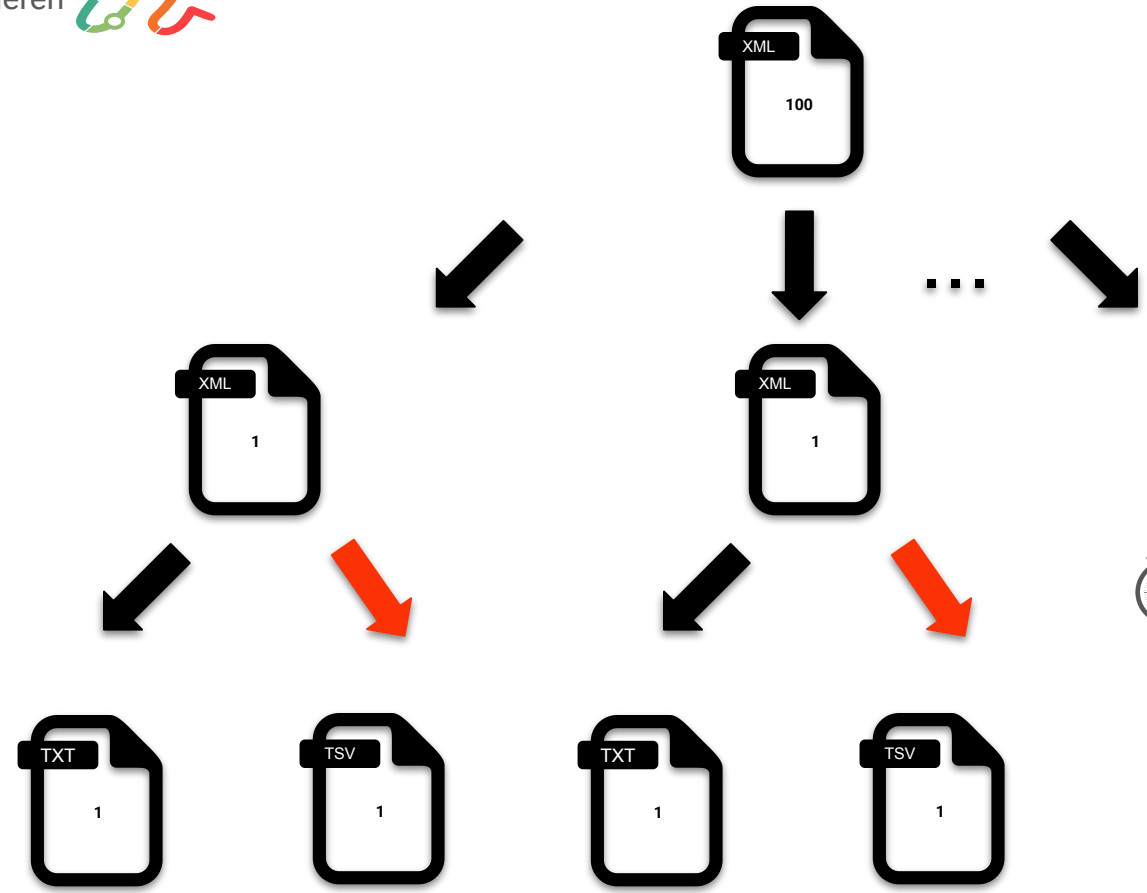
4 Daten transformieren



Dauer: 1 - 5 Stunden

https://github.com/runnwerth/annif_automated_indexing_at_tib/wiki/%C3%9Cbersetzen-des-Dumps-in-die-annif-Syntax

4 Daten transformieren



Dauer: 1 - 5 Stunden

https://github.com/runnwerth/annif_automated_indexing_at_tib/wiki/%C3%9Cbersetzen-des-Dumps-in-die-annif-Syntax

4 Daten transformieren

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

https://github.com/runnwerth/annif_automated_indexing_at_tib/wiki/%C3%9Cbersetzen-des-Dumps-in-die-annif-Syntax

TXT: (Semantisch verwertbare) Metadaten

```
Führer auf den deutschen Schiffahrtstraßen
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>
```

https://github.com/runnwerth/annif_automated_indexing_at_tib/wiki/%C3%9Cbersetzen-des-Dumps-in-die-annif-Syntax

FTX

```
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>
```

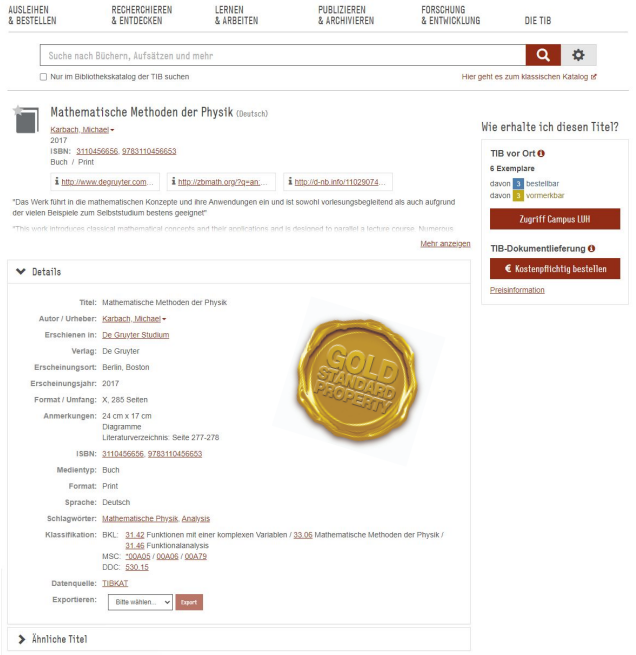
https://github.com/runnwerth/annif_automated_indexing_at_tib/wiki/%C3%9Cbersetzen-des-Dumps-in-die-annif-Syntax

5 Goldstandard definieren

Dauer: ~ 10 Stunden

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

- 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 Metadatensätzen: ~12.700 Metadatensätze für den Goldstandard
- Wir haben diesem Fall geschummelt und nur 130 Metadatensätze ausgesucht.



6 Testmenge extrahieren

https://github.com/runnwerth/annif_automated_indexing_at_tib/wiki/Extraktion-einer-Testmenge

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

Konfiguration: projects.cfg

https://github.com/runnwerth/annif_automated_indexing_at_tib/wiki/Aufsetzen-eines-annif-Projektes

Ablauf: myAnnifProject.sh

```
#!/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
```

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

8 Ergebnisse analysieren

Evaluierungsfeature von annif

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

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

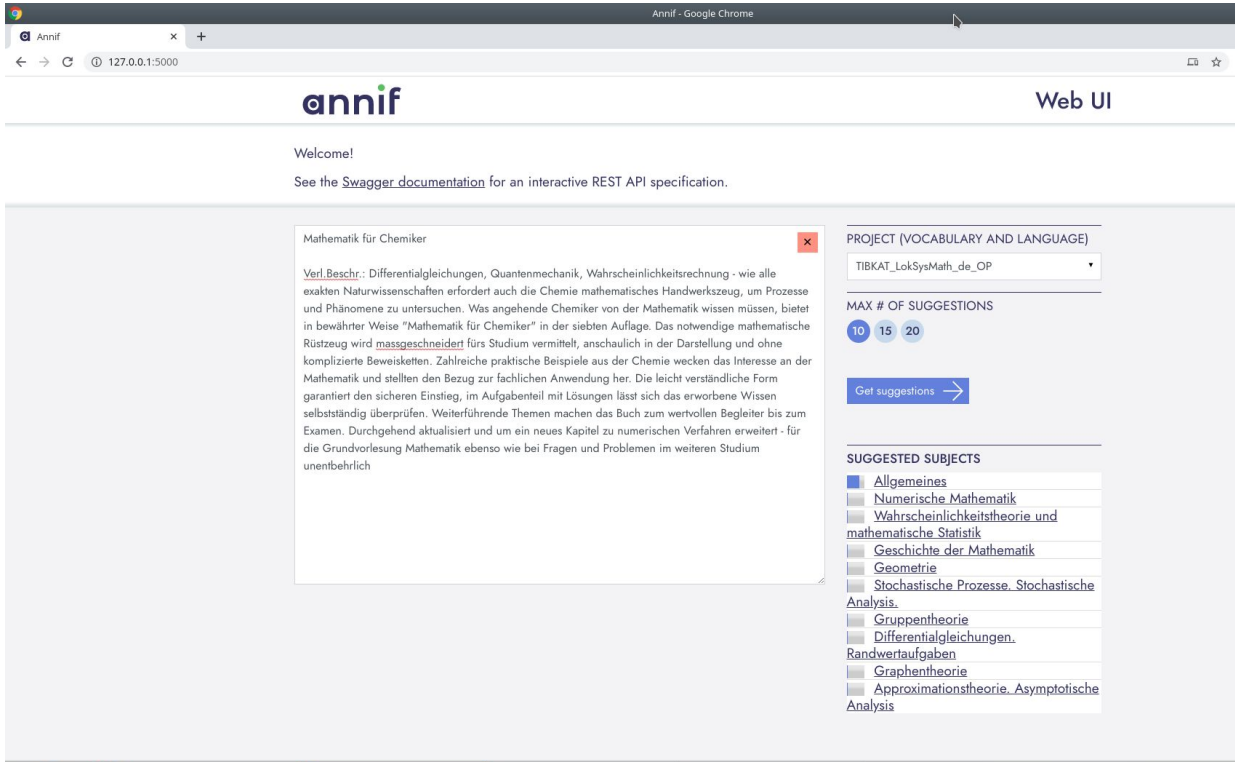
Precision (doc avg):	0.6076112412177986
Recall (doc avg):	0.6586065573770491
F1 score (doc avg):	0.5950772360608426
True positives:	147
False positives:	119
False negatives:	81
Documents evaluated:	122

TIB-Index, TIBKAT, Lokalsystematik, deutsch, Omikuji

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

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_indexing_at_tib/wiki/Weitere-Ideen

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

Wie kann es weitergehen?

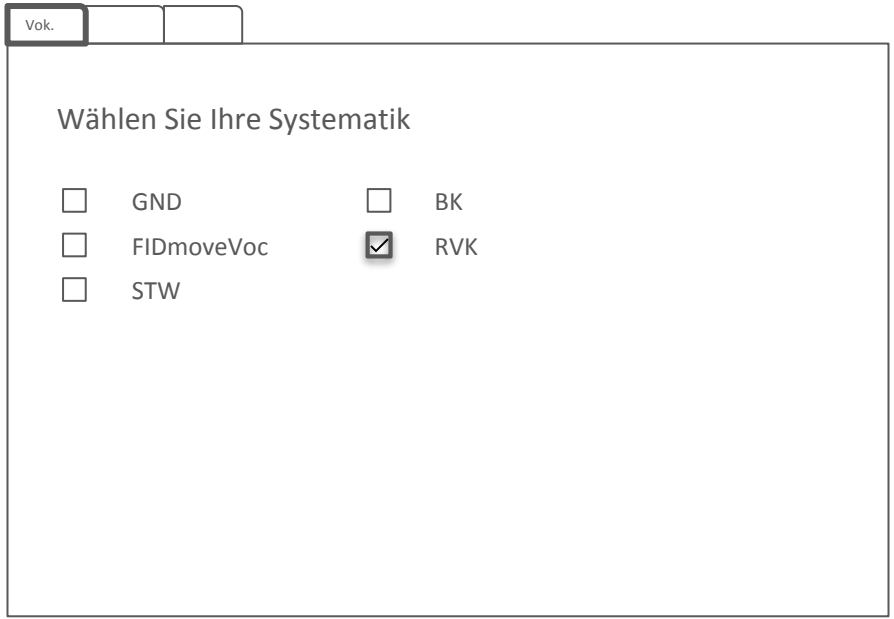
Benchmark-Matrix mit einem vielversprechenden Korpus.

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

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

Wie kann es weitergehen?

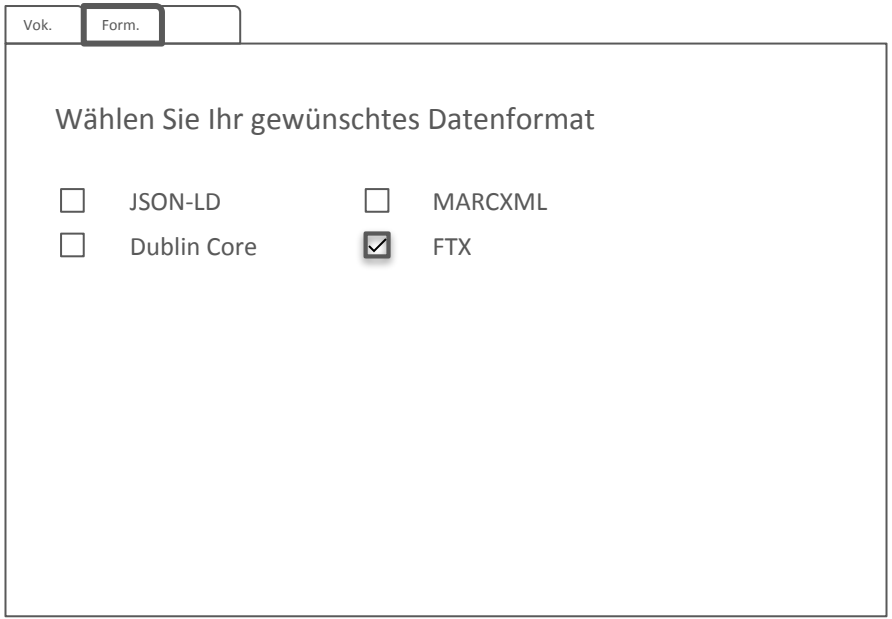
Datenaufbereitungsprozess benutzungsfreundlicher gestalten (LowFi)



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

Wie kann es weitergehen?

Datenaufbereitungsprozess benutzungsfreundlicher gestalten (LowFi)



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

Machen Sie mit!

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