

The ABC of Computational Text Analysis

#7 WORKING WITH (YOUR OWN) DATA

Alex Flückiger
Faculty of Humanities and Social Sciences
University of Lucerne

April 14, 2023

Recap last Lecture

- describe text as pattern using RegEx
- extract + replace textual parts

literal: abc

meta: \w \s [^abc] *

power of .*

Outline

- learn about available data resources
- use your own textual data

any text ✓

“any” format

from anywhere ✓



Data Sources



What Data Sources are there?

- broadly social
 - newspapers + magazines
 - websites + social media
 - reports by NGOs/GOs
- scientific articles
- economic
 - business plans/reports
 - contracts
 - patents

👉 basically, any textual documents...

Interesting Publishers

- **Nexis Uni**
newspaper, business + legal reports (international)
licensed by the university
- **Constellate**
scientific articles of JSTOR across disciplines
provides an easy dataset builder
- **HathiTrust and Project Gutenberg**
massive collection of books (international)
open, HathiTrust requires agreement

 check out other resources licensed by **ZHB**

Dataset Search

- **Harvard Dataverse**
open scientific data repository
- **Google Dataset Search**
Google for datasets basically
- corpora by the Department of Computational Linguistics @ UZH

👉 search for a topic followed by corpus, text collection or text as data

Some great historical Corpora

ready off-the-shelf, machine-readable

- 1 August speeches by Swiss Federal Councilors
provided via course repo
- Human Rights Reports by various NGOs
- United Nations General Debate Corpus



There are still not many.

Online Computational Text Analysis

- **Impresso**
 - many historical newspapers + magazines (CH, LU)
 - free, requires account
- **bookworm HathiTrust**
 - great filtering by metadata
 - credible scientific source
- **Google Ngram Viewer**
 - no filtering option
 - useful for quick analysis

Search Techniques

Make your web search more efficient by using dedicated tags. Examples:

- "computational social science"
- nature OR environment
- site:nytimes.com

Data is property

... and has rights too

- copyright may further limit access to high quality data
- check the rights before processing the data



Copyright
All Rights Reserved



Creative Commons
Some Rights Reserved

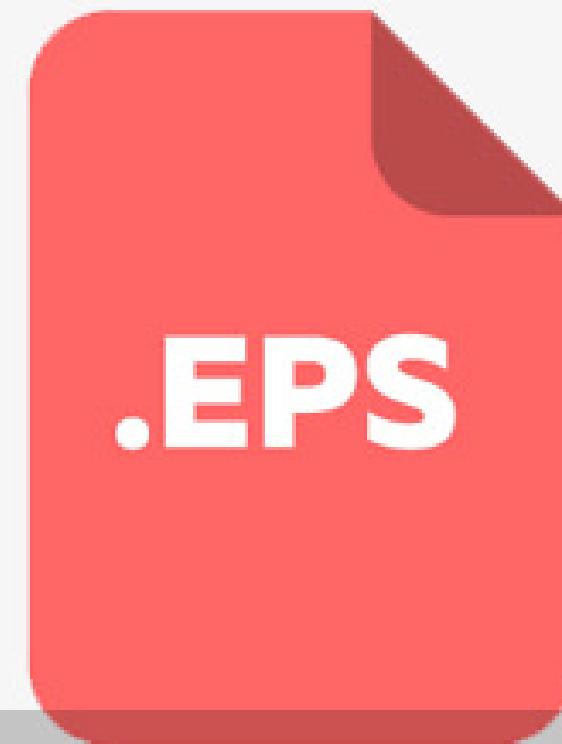


Public Domain
No Rights Reserved

Copyrights may restrict some data use ([src](#))

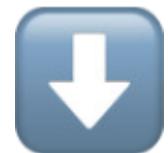
Preparing your own Data

A world for humans ...
and a jungle of file
formats.



Common Conversions

news, press releases, reports from organizations



digital native documents
.pdf, .docx, .html

scans of (old) documents
.pdf, .jpg, .png

convert to .txt

Optical Character Recognition (OCR)

machine-readable A green square icon containing a white checkmark, indicating that the conversion is machine-readable.

Conversion of DOCX

use case: news articles from Nexis

- `pandoc` to convert many file formats
- download as single articles in `.docx` on Nexis

```
1 # convert docx to txt
2 pandoc infile.docx -o outfile.txt
3
4 ### Install first with
5 brew install pandoc      # macOS
6 sudo apt install pandoc # Ubuntu
```

Conversion of native PDF

use case: Swiss party programmes

- `pdftotext` extracts text from non-scanned PDF

```
1 # convert native pdf to txt
2 pdftotext -nopgbrk -eol unix infile.pdf
3
4 ### Install first with
5 brew install poppler          # macOS
6 sudo apt install poppler-utils # Ubuntu
```

Optical Character Recognition (OCR)

- OCR ~ convert images into text
extract text from scans/images
- **tesseract** performs OCR
language-specific models
supports handwriting + Fraktur texts
- image quality is crucial

Wir gehen schnell, um die Küh
wohl, daß wir an der hellen Sc
hellen Sonne ...

Wir gehen schnell, um die Küh
wohl, daß wir an der hellen Sc
hellen Sonne ...

Wir gehen schrigJL um die Küh
wohl, daß wir an der hellen Son
hellen Sonne ...

steps when performing OCR ([Wikipedia](#))

Conversion of digitalized PDF

use-case: historical party programmes

1. extract image from PDF + improve contrast
2. run optical character recognition (OCR) on the image

```
1 # convert scanned pdf to tiff, control quality with parameters
2 convert -density 300 -depth 8 -strip -background white -alpha off \
3 infile.pdf temp.tiff
4
5 # run OCR for German ("eng" for English, "fra" for French etc.)
6 tesseract -l deu temp.tiff file_out
7
8 ### Install first with
9 brew install imagemagick          # macOS
10 sudo apt-get install imagemagick   # Ubuntu
```

Configure ImageMagick properly

Windows Ubuntu users need to execute the following

```
1 # disable security policy for Windows
2 sudo sed -i '/<policy domain="coder" rights="none" pattern="PDF"/d'
3
4 # increase memory limits
5 sudo sed -i -E 's/name="memory" value=".+$/name="memory" value="8G:/g'
6 sudo sed -i -E 's/name="map" value=".+$/name="map" value="8GiB"/g'
7 sudo sed -i -E 's/name="area" value=".+$/name="area" value="8GiB"/g'
8 sudo sed -i -E 's/name="disk" value=".+$/name="disk" value="8GiB"/g'
```

#LifeHack: Make a PDF searchable

use case: scanned book chapters

```
1 # output searchable pdf instead of txt
2 convert -density 300 -depth 8 -strip -background white -alpha off
3 file_in.pdf temp.tiff
4
5 tesseract -l deu temp.tiff file_out pdf
```

Scraping PDF from Websites

use case: Swiss voting booklet

- `wget` to download any files from the internet

```
1 # get a single file
2 wget EXACT_URL
3
4 # get all linked pdf from a single webpage
5 wget --recursive --accept pdf -nH --cut-dirs=5 \
6 --ignore-case --wait 1 --level 1 --directory-prefix=data \
7 https://www.bk.admin.ch/bk/de/home/dokumentation/abstimmungsbuechle
8
9 # --accept FORMAT_OF_YOUR_INTEREST
10 # --directory-prefix YOUR_OUTPUT_DIRECTORY
```

Interesting Resources

- **Party Programmes across Europe**
covers over 1000 parties from 1920 until today in over 50 countries
- **Swiss voting booklets**
from 1978 until today
- **1 August speeches by Swiss Federal Councillors**
- **Nestlé Annual Reports**
- ... any organization of your interest 

Basics of Batch Processing

perform the same operation on many files

```
1 # loop over all txt files
2 for file in *.txt; do
3
4     # indent all commands in loop with a tab
5
6     # rename each file
7     # e.g. a.txt -> new_a.txt
8     mv $file new_${file}
9
10 done
```

Perform OCR for many PDF

```
1 for FILEPATH in *.pdf; do
2     # convert pdf to image
3     convert -density 300 $FILEPATH -depth 8 -strip \
4         -background white -alpha off temp.tiff
5
6     # define output name (remove .pdf from input)
7     OUTFILE=${FILEPATH%.pdf}
8
9     # perform OCR on the tiff image
10    tesseract -l deu temp.tiff $OUTFILE
11
12    # remove the intermediate tiff image
13    rm temp.tiff
14
15 done
```

Preprocessing → RegEx





Questions?

In-class: Exercises I

1. Make sure that your local copy of the Github repository KED2023 is up-to-date with `git pull`. Check out the data samples in `materials/data` and the scripts to extract their text in `materials/code`.
2. Install the missing tools with the commands given on the respective slides: `pandoc`, `imagemagick`, `poppler`.
3. **Apply the commands to reproduce on the given data. Test them on your own data. Check the resources. Ask questions. Think about your mini-project.**

In-class: Exercises II

1. Use `wget` to download *cogito* and its predecessor *uniluAKTUELL* issues (PDF files) from the [UniLu website](#). Start with downloading one issue first and then try to automatize the process to download all the listed issued using arguments for the `wget` command.
2. Convert the *cogito* and *uniluAKTUELL* PDF files into TXT files using `tesseract`. Try with a single issue first and then write a loop to batch process all of them.
3. What is the University of Lucerne talking about in its issues? Use the commands of the previous lectures to count the vocabulary.
4. Do the same as in 3.), yet analyze the vocabulary of *cogito* and *uniluAKTUELL* issues separately. Does the language and topics differ between the two magazines?

In-class: Exercises III

1. Use `wget` to download a book from Project Gutenberg and count some things (e.g., good/bad, joy/sad).
2. `wget` is a powerful tool. Have a look at its arguments and search for more examples in tutorials on the web.

Resources

Make a more sophisticated script for PDF-to-TXT conversion

- Erick Peirson. 2015. Tutorial: Text Extraction and OCR with Tesseract and ImageMagick - Methods in Digital and Computational Humanities - DigInG Confluence. [online](#)



Have a nice
Easter break!