

# The ABC of Computational Text Analysis

## #7 WORKING WITH (YOUR OWN) DATA

Alex Flückiger

Faculty of Humanities and Social Sciences  
University of Lucerne

06 April 2023

# Recap last Lecture

- describe text as pattern using RegEx
- extract + replace textual parts 
  - literal: `text a b c`
  - meta: `\w \s [^abc] *`
  - power of `.`  $*$
- questions regarding assignment 2

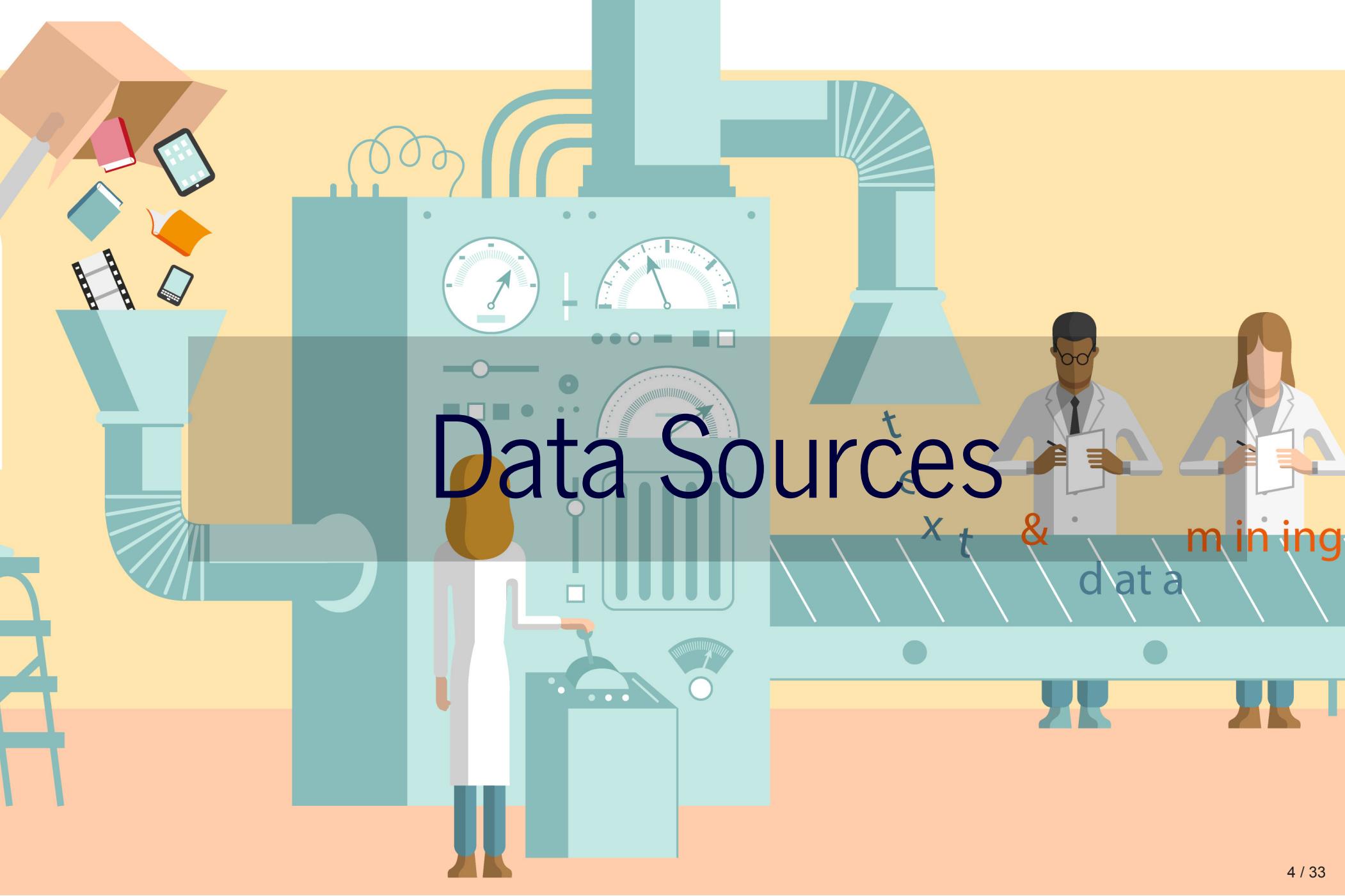
# Outline

- learn about available data sources
- Build and justify (!) your own dataset

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...

# 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.

# Swissdox: A game changer

## Assemble corpus and export as .csv

- over 250 Swiss newspapers
- historical and updated daily
- needs registration

The screenshot shows the 'Corpus query' section of the Swissdox interface. At the top, there's a navigation bar with 'Swissdox@LiRI' and links for 'Start', 'Projects', 'Corpus query' (which is highlighted in blue), 'Retrieved datasets', and 'Manual'. The main area is titled 'Corpus query' and contains three input fields: 'Languages \*' (with a dropdown menu labeled 'Select languages'), 'Source \*' (with a dropdown menu labeled 'Select sources'), and 'Date ranges' (with a date range set to '2023-04-01 ~ 2023-04-30' and a calendar icon). Below these fields is a note: '\* no filtering is applied if no option is selected'. At the bottom are two buttons: 'Reset filters' and 'Next'.

# More Interesting Publishers

- **Nexis Uni**

- international newspaper, business + legal reports
  - licensed by the university

- **Constellate**

- scientific articles of JSTOR across disciplines
  - provides an easy dataset builder

- **Project Gutenberg and HathiTrust**

- massive collection of books
  - open, HathiTrust requires agreement



check out other resources licensed by **ZHB**

# Search existing Datasets

- Harvard Dataverse  
open scientific data repository
- Google Dataset Search  
Google for datasets basically

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

# Online Computational Text Analysis

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

# Search Techniques



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

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

# Data is Property

... and has rights too

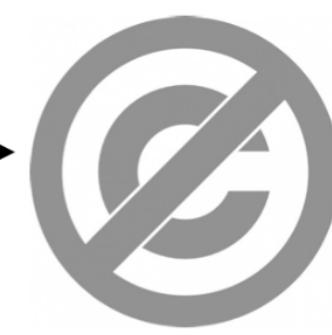
- copyright may further limit access to high quality data
- check the rights before processing data



**Copyright**  
**All Rights Reserved**



**Creative Commons**  
**Some Rights Reserved**



**Public Domain**  
**No Rights Reserved**

# Preparing your own Data

**DOC**

**.JPG**

**.PNG**

**.PSD**

**.EPS**

**.CDR**

**TXT**

**.GIF**

**.PPT**

**.MP3**

**.WAV**

**.AI**

**MOV**

**.EXE**

**.DMG**

**.RAR**

**.ZIP**

**.PDF**

**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 rounded square icon containing a white checkmark symbol.

# Conversion of DOCX

## Use case: news articles from **Nexis**

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

```
# convert docx to txt
pandoc infile.docx -o outfile.txt

### Install first with
brew install pandoc      # macOS
sudo apt install pandoc # Ubuntu
```

# PDF: Digitalized or Digital?

## Two flavours of PDF documents

### Politische Richtlinien 1951

Vom Parteitag der Schweizerischen Konservativen Volkspartei am 9. September 1951 in Schwyz einstimmig gutgeheissen.

Die Schweizerische Konservative Volkspartei bekennt sich zu den Grundsätzen der christlichen Weltanschauung. Die christliche Auffassung von der menschlichen Persönlichkeit und der Gesellschaft bildet die Grundlage ihrer Politik zur Förderung der allgemeinen Wohlfahrt.

Sie bekennt sich zur Solidarität des ganzen Volkes und lehnt sowohl den im Marxismus begründeten Klassenkampf als auch die auf dem Wirtschaftsliberalismus beruhende Vorherrschaft des Kapitals ab.

*Digitalized PDF made from a scanned page*

### EINLEITUNG

Die Schweiz braucht mehr grüne Politik. Grüne Politik für die Umwelt, für das Klima, für eine nachhaltige Wirtschaft und für soziale Gerechtigkeit in der Schweiz und in der Welt. Die nationalen Wahlen vom 20. Oktober 2019 sind dafür eine zentrale Weichenstellung. Wir GRÜNE wollen darum im National- und Ständerat mindestens vier Sitze hinzugewinnen und unseren Einfluss ausbauen.

Die GRÜNEN sind die fünftstärkste Partei in der Schweiz und haben in ihrer 36-jährigen Geschichte viel bewegt. Unsere Themen sind mitten in der Gesellschaft angekommen. Die GRÜNEN haben den Atomausstieg und die Energiewende mehrheitsfähig gemacht. 2019 wird der erste Atommeiler im bernischen Mühleberg abgestellt. Gentechfreie Landwirtschaft oder die Vereinbarkeit von Beruf und Familie sind grüne Errungenschaften, genauso wie eingetragene Partnerschaften, Verkehrsverlagerung und Tempo 30 in Wohnquartieren. Ohne GRÜNE wäre die Schweiz mit bewaffneten Missionen in Afghanistan unterwegs und hätte 22 überflüssige Kampfflugzeuge beschafft.

*Native PDF converted from digital document (e.g., docx)*

# Conversion of native PDF

## Use case: Swiss party programmes

- `pdftotext` extracts text from non-scanned PDF

```
# convert native pdf to txt
pdftotext -nopgbrk -eol unix infile.pdf

### Install first with
brew install poppler          # macOS
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*

# 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

```
# convert scanned pdf to tiff, control quality with parameters
convert -density 300 -depth 8 -strip -background white -alpha off \
infile.pdf temp.tiff

# run OCR for German ("eng" for English, "fra" for French etc.)
tesseract -l deu temp.tiff file_out

### Install first with
brew install imagemagick          # macOS
sudo apt-get install imagemagick # Ubuntu
```

# Configure ImageMagick

Only Windows Ubuntu: Paste the following in your command-line

```
# disable security policy for Windows
sudo sed -i '/<policy domain="coder" rights="none" pattern="PDF"/d'

# increase memory limits
sudo sed -i -E 's/name="memory" value=".+$/name="memory" value="8GiB/g'
sudo sed -i -E 's/name="map" value=".+$/name="map" value="8GiB"/g'
sudo sed -i -E 's/name="area" value=".+$/name="area" value="8GiB"/g'
sudo sed -i -E 's/name="disk" value=".+$/name="disk" value="8GiB"/g'
```

# #LifeHack: Make a PDF searchable

## Use case: scanned book chapters

```
# output searchable pdf instead of txt  
convert -density 300 -depth 8 -strip -background white -alpha off -  
file_in.pdf temp.tiff  
  
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

```
# get a single file
wget EXACT_URL

# get all linked pdf from a single webpage
wget --recursive --accept pdf -nH --cut-dirs=5 \
--ignore-case --wait 1 --level 1 --directory-prefix=data \
https://www.bk.admin.ch/bk/de/home/dokumentation/abstimmungsbuechle

# --accept FORMAT_OF_YOUR_INTEREST
# --directory-prefix YOUR_OUTPUT_DIRECTORY
```

# Interesting Resources

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

# What data are you interested in?

Think about the topic of your mini-project



*Illustration of text analysis generated by [Image Creator from Microsoft Bing](#)*

# Basics of Batch Processing

perform the same operation on many files

```
# loop over all txt files
for file in *.txt; do

    # indent all commands in loop with a tab

    # rename each file
    # e.g. a.txt -> new_a.txt
    mv $file new_${file}

done
```

# Perform OCR for many PDF

```
for FILEPATH in *.pdf; do
    # convert pdf to image
    convert -density 300 $FILEPATH -depth 8 -strip \
    -background white -alpha off temp.tiff

    # define output name (remove .pdf from input)
    OUTFILE=${FILEPATH%.pdf}

    # perform OCR on the tiff image
    tesseract -l deu temp.tiff $OUTFILE

    # remove the intermediate tiff image
    rm temp.tiff

done
```



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. Decide on one use-case that interests you most. Install the missing tool with the commands given on the respective slides (e.g., `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 `pdftotext`. 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!