

# The ABC of Computational Text Analysis

## #7 WORKING WITH (YOUR OWN) DATA

Faculty of Humanities and Social Sciences  
University of Lucerne

15 April 2022

# Updated Schedule

Date	Topic
25 February 2022	Introduction + Where is the digital revolution?
04 March 2022	Text as Data
11 March 2022	Setting up your Development Environment
18 March 2022	Introduction to the Command-line
25 March 2022	Basic NLP with Command-line
01 April 2022	Learning Regular Expressions
08 April 2022	<i>no lecture (Osterpause)</i>
15 April 2022	Working with (your own) Data <del>Advanced RegEx + Data Sources</del>
22 April 2022	Ethics and the Evolution of NLP <del>Creating new Data Sets + Ethics</del>
29 April 2022	Introduction to Python
06 May 2022	NLP with Python
13 May 2022	<i>no lecture (Christi Himmelfahrt)</i>
20 May 2022	NLP with Python + Working Session
27 May 2022	Mini-Project Presentations + Discussion
03 June 2022	<i>no lecture (Fronleichnam)</i>

# Recap last Lecture

- extract + replace textual parts with RegEx

*literal:* `abc`

*meta:* `\w \s [^abc] *`

# Outline

- you are on spot!  #assignment2
- learn about available data resources
- use your own textual data
  - any text* ✓
  - “any” format*
  - from anywhere* ✓

# Assignment #2

# Feedback Assignment #2

## example solution

- make patterns more general

*date:* `DD*` `Month` `DDDD`

- keep it simple

*name of month ~ any word ~* `\w+`

- avoid false positives with positional information

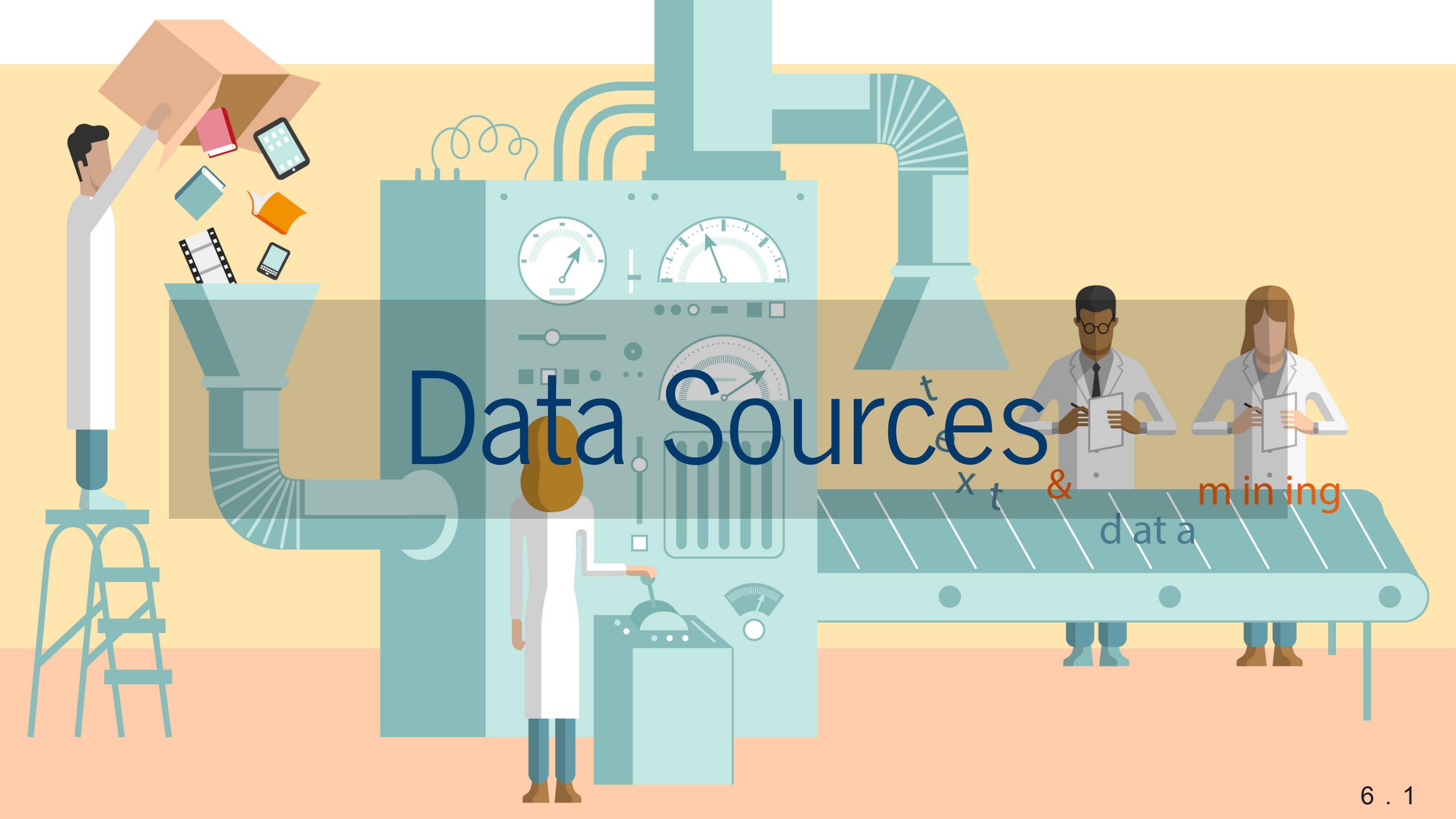
*start of line:* `^`

- names are hard to extract

*variation + inconsistency*



check the count of matches with `wc` + the cleanup with `diff`



A large industrial-style machine, resembling a factory or refinery, is the central element. It features various pipes, valves, and gauges in shades of teal and light blue. On the left side, a man in a white lab coat is shown pouring books, papers, and electronic devices (a tablet, a smartphone, a film strip) into a funnel that feeds into the machine. On the right side, two scientists in lab coats are standing behind a conveyor belt system, writing in notebooks. The conveyor belt has the words "Data Sources" and "mining" written on it. The background is a warm yellow-orange color.

# Data Sources

mining

# What Data Sources are there?

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

👉 basically, any textual document...

# Interesting Publishers

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

# 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` or `text collection`

# Search Techniques

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

- "computational social science"
- nature OR environment

# Some great historical Corpora

ready off the shelf, machine-readable

- 1 August speeches by Swiss Federal Councillors  
*provided via course repo*
- Human Rights Texts
- United Nations General Debate Corpus

# 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*

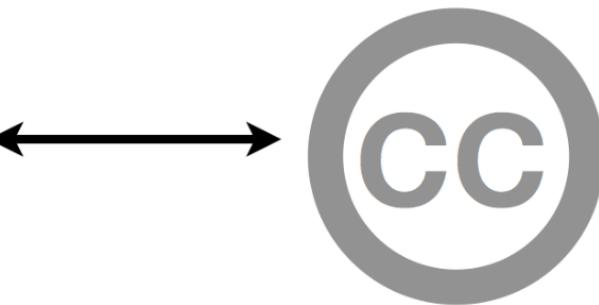
# Copyright



- 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



# 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 a feature or result.

# 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
```

# 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 ([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

```
# 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
```

# #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 -compress group4 \
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/abstimmungsbuechlein.html

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

# Example Sources

- Party Programmes across Europe
- Swiss voting booklets
- 1 August speeches by Swiss Federal Councillors
- Nestlé Annual Reports
- ... any organization of your interest 

# Foundation 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 Batch OCR from 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
```

# Preprocessing → RegEx





Questions?

# In-class: Exercises I

1. Make sure that your local copy of the Github repository KED2022 is up-to-date with `git pull`. Check out the data samples and scripts in `materials/`.
2. Install the missing tools with the commands given on the respective slides: `pandoc`, `imagemagick`, `poppler`
3. **Digest the commands. Test them. Check the resources. Ask questions. Think about your mini-project.**
4. Use `wget` to download one or all `cogito` issues (PDF files) from the [UniLu website](#).
5. Convert the `cogito` PDF files into TXT files.
6. `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 conversion

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

# References