Seminar: The ABC of computational Text Analysis

Alex Flückiger

2020

Contents

1 Outline

In this practice-oriented seminar, bachelor students of social and cultural sciences learn relevant technical skills and programming that they can incorporate into their everyday studies. Moreover, they develop an understanding of current developments in the field of information technology. This course aims to foster general digital literacy and to build a solid foundation for computational analysis, using Python and the command-line.

In this seminar, we focus on the computational processing of digital and digitized texts. A key point of scientific work is the systematic preparation and aggregation of data as well as the swift retrieval of relevant information. This task requires the handling of a wide variety of data forms, including data that is not yet structured in a tabular form. The seminar covers the full workflow from gathering textual data to analyze the content of an entire text collection to producing interactive visualizations. Sounds cool? It certainly is.

Along the road, we deal with questions like these:

- How can texts be quantitatively exploited to complement the qualitative content analysis?
- What are regular expressions, and why are they so powerful in the context of text analysis?
- How to download data automatically from a website and process *en masse*?
- How can historical texts be extracted from PDFs using Optical Character Recognition (OCR)?

Link to lecture on UniLu website

2 Schedule

This is a new course. The material emphasized, as well as the schedule, will be adapted to the needs and interests of the students. Thus, the schedule below is provisional. Likely, we will change some topics and orderings as we go.

We have 12 seminar sessions together.

Date	Topic
27 February 2020	Introduction + Where is the digital revolution?
05 March 2020	Text as Data
12 March 2020	Setting up your Development Environment
19 March 2020	Introduction to the Command-line
26 March 2020	Basic NLP with Command-line
02 April 2020	Learning Regular Expressions
09 April 2020	RegEx + Data Sources
16 April 2020	*no lecture (Osterpause)*
23 April 2020	Creating new Data Sets + Ethics
30 April 2020	Introduction to Python
07 May 2020	NLP with Python
14 May 2020	NLP with Python + Working Session
21 May 2020	*no lecture (Christi Himmelfahrt)*
28 May 2020	Mini-Project Presentations + Discussion(#week-12-mini-project-presentations–discussion)

3 Sessions

3.1 Week 1: Introduction + Where is the digital revolution?

On the one hand, I present the goals and organization of the seminar. On the other hand, we look at some recent applications that give an impression of the fascinating prospects of computers in the area of artificial intelligence (AI) and digital humanities (DH).

Slides: HTML | PDF

3.1.1 Required Reading

- Lazer, David, Alex Pentland, Lada Adamic, Sinan Aral, Albert-László Barabási, Devon Brewer, Nicholas Christakis, Noshir Contractor, James Fowler, Myron Gutmann, Tony Jebara, Gary King, Michael Macy, Deb Roy, and Marshall Van Alstyne. 2009. "Computational Social Science." *Science* 323(5915):721–23.

3.1.2 Optional Reading

- Graham, Shawn, Ian Milligan, and Scott Weingart. 2015. *Exploring Big Historical Data: The Historian's Macroscope*. Open Draft Version. Under contract with Imperial College Press. online

3.2 Week 2: Text as Data

Computational text analysis comes with many challenges that are unique due to the fuzziness of natural language. In this session, we learn about its methodological foundation, and we conduct our first computational text analysis to understand how this translates into practice.

Slides: HTML | PDF

3.3 Week 3: Setting up your Development Environment

The title says it all. We are getting ready for the practical part of the course: Programming. As the installation of Python and non-standard command-line tools may be tricky, we do this in class rather than doing it as homework. Moreover, I will also introduce some principles to organize research and jargon that help you to find your way in the programmer's brave new word.

Slides: HTML | PDF | Installation Guide

3.3.1 Optional: pimp your workflow

- Healy, Kieran. 2019. "The Plain Person's Guide to Plain Text Social Science." online.

3.4 Week 4: Introduction to the Command-line

The command-line is a powerful tool at your disposal. It is the working horse for many data wrangling tasks. In this session, you learn the basics of shells and perform many operations by effectively substituting clicks on the screen with commands. Admittedly, it is not overly exciting at this stage, yet it is essential for more sophisticated automation later on.

Slides: HTML | PDF

3.4.1 Recommended Resources

- The Programming Historian
- DigitalOcean

3.5 Week 5: Basic NLP with Command-line

Counting words is the most basic method to look at texts from a computational perspective. The command-line provides tools to quickly sift through a massive text collection to describe the use of words quantitatively. In no time, you can also take a systematic look at the word usage in context. Sounds like a Swiss knife for computational text analysis in social science? It certainly is.

Slides: HTML | PDF

3.6 Week 6: Learning Regular Expressions

When working with text data, you spend a lot of time to clean your documents and extract some pieces of information. Doing this by hand is not only a pain but simply impossible when facing many documents. Fortunately, there is a formal language named Regular Expressions that allows writing expressive and generalizable patterns. Using these patterns, you can extract and remove any textual parts systematically without missing a single instance.

Slides: HTML | PDF

3.6.1 Required Reading

- Ben Schmidt. 2019. Regular Expressions. online

3.6.2 Recommended Resource

Everything we have touched about text processing in greater detail.

- Nikolaj Lindberg. egrep for Linguists. online

3.6.3 Online Regular Expression Editor

A visual editor to check your regular expressions.

- Rubular

3.7 Week 7: RegEx + Data Sources

To this point, you have acquired the skills to cut a document into pieces and, subsequently, to extract, replace, and count any textual elements. Unless you have interesting data, these tools are neat but of no greater use. Besides some further practicing with RegEx, we turn to relevant data resources in social science. Given you have plain text at hand, your tools cut through this data like butter. For other formats, we learn about some remedies in the next session.

Slides: HTML | PDF

3.8 Week 8: Creating new Data Sets + Ethics

The world we live in is not made for machines but people – for better or for worse. While perfectly readable, documents often require a subsequent conversion to allow machine processing. Firstly, digital documents are shipped in various formats and need a conversion to plain text. Secondly, historical documents require an additional step called optical character recognition (OCR) to extract the text from the scanned original. Converting thousands of documents is easy when using the shell.

Slides: HTML | PDF

3.9 Week 9: Introduction to Python

It may come as a surprise that we start with Python in the ninth session only. As the folks say, Python is among the coolest programming languages, relatively easy to learn, and provides excellent NLP packages so that you don't have to implement everything yourself. All true as long as you have your data ready. In this session, we begin with an introduction to the basic syntax of Python. Starting with this dry subject is necessary as it allows you to modify the more sophisticated NLP analyses to your needs.

Slides: HTML | PDF

3.10 Week 10: NLP with Python

Python is the language of choice when it comes to serious NLP. Have you ever wondered how the frequency of terms evolves over the years? Or how the language differs between two groups of documents whereby the groups may be formed by any metadata (person, organization, gender etc.)? Exploring is most effective in an interactive and visual mode - so be it. Among some basic statistics, this is the serious stuff where we finally arrive in our journey. Moreover, you will learn the jargon of NLP to don't get lost in the forest of yet unknown terms.

Slides: HTML | PDF

3.10.1 Code

Go to the static code: Python NLP

To run the code in your browser without any installation:

launch binder

(path `scripts/KED2020_10.ipynb`)

3.11 Week 11: NLP with Python II + Working Session

In today's session, we continue our deep dive into NLP with Python. It is the last piece in our puzzle. During this course, you have learned about the entire workflow, from assembling datasets of documents to analyze their content and visualize your findings. As soon as you have a structured text collection along with basic meta data (e.g., publication date), you can take numerous perspectives to look at your data. At this stage, it is time for the kick-off of the mini-projects allowing you to work with your data of interest.

Slides: HTML | PDF

3.11.1 Explore interactively: 1 August Speeches by Swiss Federal Councilors

As a matter of tradition, Swiss Federal Councilors give an official speech on Swiss National Day. Simon Schmid (journalist Republik), with the collaboration of Prof. Andreas Kley (Faculty of Law, UZH), collected many of these speeches and kindly shared the resulting dataset with me. The collection comprises 166 speeches, which is a multiple of the publicly available here.

The interactive visualization linked below shows how the language differs between speakers of *Social Democratic Party of Switzerland* (SP) and speakers of other parties. The top right corner shows terms that have been frequently used by all parties, while the top left and the lower right corner reveal words that have been used primarily by the members of the SP and correspondingly by the center-right parties.

You can search for the terms of your interest. Moreover, you may click on the points in the plot to show the context of the corresponding words within speeches. These functions allow for a quick investigation of the corpus along the dimensions of Swiss parties.

Explore in Browser *(it takes a few seconds to load)*

3.12 Week 12: Mini-Project Presentations + Discussion

In this session, it is entirely your turn. Going beyond mere toy examples, you present what you have worked on in groups and showing off your first harvest of computational text analysis.

The seminar is coming to an end, yet it doesn't have to be a dead-end. You may have gotten more proficient in cursing your computer but also making your way through the jungle of technology. Continue the journey, cheers!

Slides: HTML | PDF

4 Exercises

You need to submit three exercises to complete the seminar successfully. The point of the exercises is not to make it hard to pass but rather to foster the engagement with the covered material of this class. As you like, you may prefer to work in teams to discuss different approaches. Nonetheless, each student has to submit his own solution.

#	Topic	Published	Deadline
1	Data Wrangling	19 March 2020	26 March 2020 (by midnight)
2	Regex NLP	02 April 2020	16 April 2020 (by midnight)
3	Python NLP	07 May 2020	14 May 2020 (by midnight)

4.1 Formal Instructions

Your submission is a single script, meaning that is readily executable, and is named as follows:

- Bash scripts: `SURNAME_KED2020_ex_NR.sh`
- Python scripts: `SURNAME_KED2020_ex_NR.py`

The script follows the order of the tasks in the exercise. In addition to the commands you have used to come up with a solution, you also provide a short, yet concise explanation of the actual solution as a comment (lines starting with `#`). Please use the following examples as template:

4.1.1 Bash

```
#!/bin/bash

##################################################
### Exercise 1
### Seminar: The ABC of computational Text Analysis
### University of Lucerne
##################################################

### task 1a)
echo "this is a test"
# solution: echo prints out the provided text in the commandline

### task 1b)
echo -n "test" | wc
# solution: wc counts the lines, words and characters.
# The argument -n is necessary to omit the trailing new-line symbol.
# "test" has 4 characters.

...
```

4.1.2 Python

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

##################################################
### Exercise 1
### Seminar: The ABC of computational Text Analysis
### University of Lucerne
##################################################

### task 1a)
print("Hello, World")
# outputs the provided string to the prompt

...
```

5 Mini-Project

In the final session, you present a short analysis. You are free to choose your research question as well as the used computational methods and data. It is certainly more fun when you work with data from your area of interest.

Again, the aim of this project is not to overwhelm students with too ambitious requirements. It should be the other way around. You will have as much freedom as you need to engage with your data creatively. I will be glad when you realize that your knowledge is already good enough to perform powerful analyses.

The only requirement is that you have to complement your claims with some quantitative facts about the data, which you can freely choose. You may work in teams of two.

6 Optional: Writing a Seminar Paper

You are welcome to write a seminar paper (Hauptseminararbeit) for which you get additional credit points. As I am in the position of a guest lecturer, I will accept seminar papers in cooperation with Prof. Sophie Mützel.

Due to the practical foundation of this seminar, you are well-prepared to apply computational text analysis in a personal project subsequently. Although this is not a requirement, you may want to turn your mini-project into a seminar paper by deepening your empirical inquiry.

Students planning to write a seminar paper should send me an email with their research idea until **10 May 2020** at the latest. When you would like to discuss your idea in person, feel free to do so any time after the seminar.

Requirements for the seminar paper (Hauptseminararbeit):

- Write your thesis in German or English
- Use any computational methods to analyze your data
- Your paper has a theoretical question guiding your methodical approach. In other words, methods are a means, not an end in themselves.
- Formal: 15 pages (A4), 12 pt Times New Roman, 3cm margin, 1.5 line spacing
- Deadline for submitting the final paper: **31 August 2020**

7 Kursbeschreibung (German)

In diesem praxisorientierten Seminar erlernen die Studierenden aller Fächer der KSF zentrale technische Fertigkeiten, die sie in ihren unmittelbaren Studienalltag einbauen können, und erhalten darüber hinaus auch einen Eindruck über aktuelle technische Entwicklungen. Das Ziel dieser Veranstaltung ist das technische Sensorium zu schärfen und eine solide Basis für weiterführende computergestützte Analysen zu schaffen.

Zentral für alle Arten des wissenschaftlichen Arbeitens ist das systematische Aufbereiten und Aggregieren von Daten sowie das selektive Auffinden von Informationen. Diese Arbeit erfordert ein Umgang mit vielfältigen Datenformen, die insbesondere auch nicht tabellarisch strukturiertes Datenmaterial umfassen. Der Seminarfokus liegt hierbei auf der computergestützten Prozessierung von digitalen und digitalisierten Texten. Das Seminar bearbeitet Fragen wie diese:

- Wie lassen sich Texte quantitativ erschliessen, um die qualitative Inhaltsanalyse zu komplementieren?
- Was sind reguläre Ausdrücke und wieso sind diese für textanalytische Fragestellungen ungemein nützlich?
- Wie können Daten automatisiert aus dem Internet geladen und massenhaft verarbeitet werden?
- Wie können historische Texte mithilfe von Optical Character Recogniton (OCR) aus PDFs extrahiert werden?

Inputs von den Studierenden für inhaltliche Schwerpunkte sind willkommen.