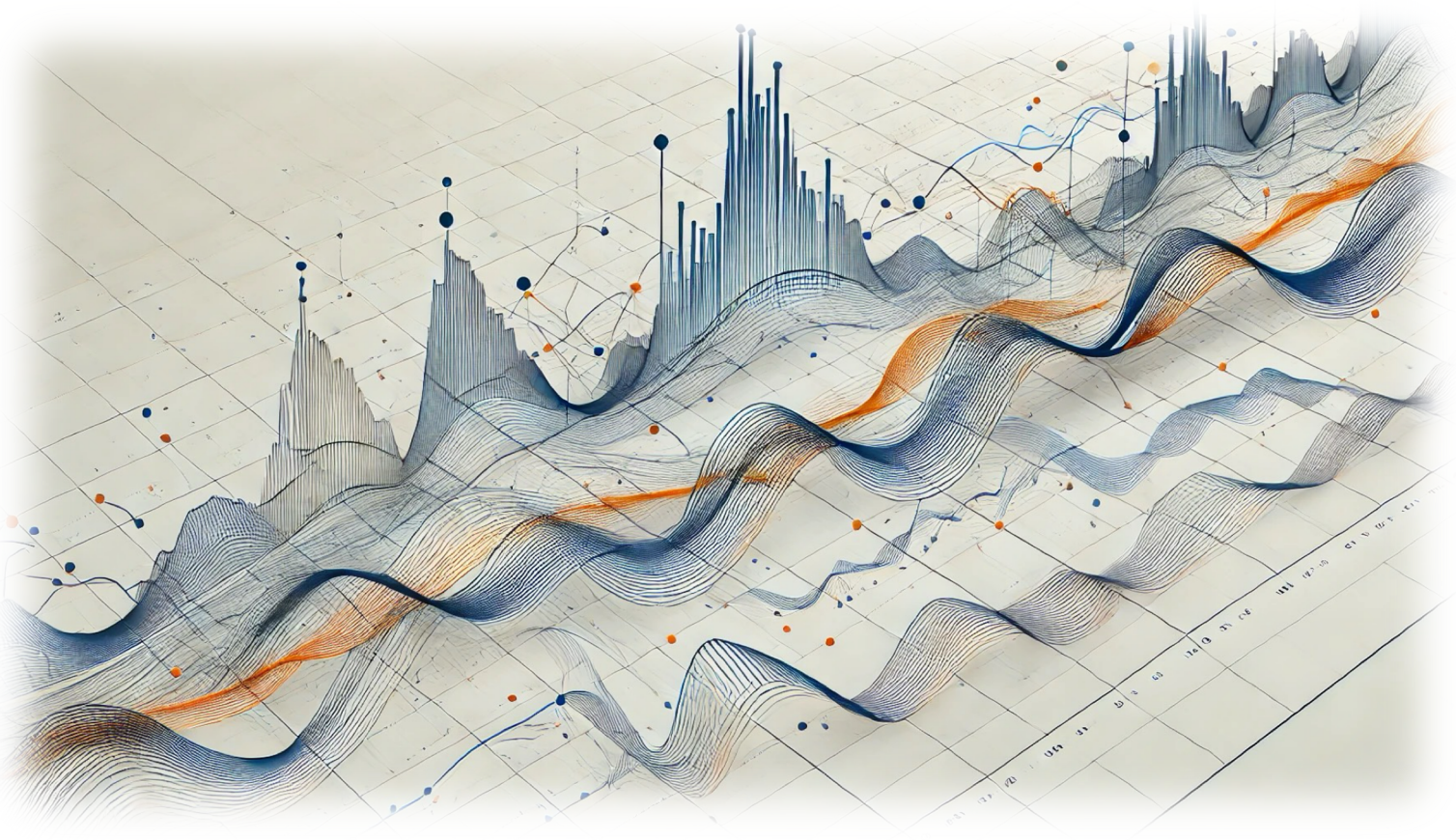
Module Ib: Digital Text Analysis

History Lab
RMA Modern and Contemporary History
GKRMV24003

2025/26 block 2



Teacher

Dr. Pim Huijnen
Drift 6, room 0.22
p.huijnen@uu.nl

Utrecht University

Schedule

Tuesday 11:15 – 15:00, Digital Humanities Workspace, UB City Centre, Drift 27, room 0.32

Description

The purpose of this module is to familiarize students of the Research MA programme Modern and Contemporary History with computational techniques to analyse textual data. More and more historians are using large sets (or corpora) of digital texts – both digitised and born-digital – as well as a growing variety of digital techniques to extract information from these texts. These range from counting tokens or words in texts to AI-based modeling. Scholars are discussing the disruptive nature of these increasingly common practices for historical scholarship under the header of Digital Humanities and Digital History. It is, therefore, timely for you to become acquainted with a number of widely used techniques and skills, and to learn how to use them for the study of history in a critical and conscientious way. A second goal of this tutorial is to provide you with sufficient information to form your own opinion of the extent to which digital resources and tools change the heuristics of historical enquiry.

In this module we will work with the Python programming language in the Jupyter Notebook environment. Off-the-shelf software for text analysis does exist (AntConc, Voyant Tools, Sketch Engine), but working with Python provides endlessly more flexibility. You will be trained in quantitative techniques in a 'hands on' setting: you will work on your own research projects in groups of 2, based on a topic of your interest. You will start by gathering a dataset that represents your topic. Next, you will be handed different techniques, in the form of Jupyter notebook programs, to extract information from this data. This will cover the larger part of the weekly meetings. The hands on-sessions will be preceded by short lectures that aim to spur reflection on the implications of digitization for the study of history.

Assessment: Portfolio

In this tutorial, you will study a topic of your choice in groups of two. Collaboration is a core-element of digital history, since 'it is rare for a single individual (be it instructor or student) to have all of the technical and interpretive skills that are needed to reach a learning outcome' (Guiliano 2022). As a group, you will describe what you did in a research portfolio (ca. 5.000 words) to be handed in at the end of the tutorial (**23 January 2026, midnight**).

All groups will present their projects to the rest of the class in one of the block weeks 4-7 (10 minute presentation + 10 minute discussion). The focus in these presentations will be less on the results than on the general progression of your project: from topic or question to finding data to iterating between close and distant reading, as well as any challenges that you may encounter in the process. This is, among other things, to balance out the fact that students present their work in different weeks and in different stages of their projects.

Use a Powerpoint/Keynote presentation for your presentation, but restrict yourself to max 4 slides, the last of which with questions for the rest of the group. These presentations, after all, are among other things meant as an opportunity to use the brainpower of the group to collect

suggestions, ideas, and best practices. To this end, every presentation should entice a discussion with the group, for which you – as presenters – are responsible.

On data

There is no restriction on the topic that you study in this tutorial, but there are a few prerequisites for the **data** that you will use. Obviously, you should be able to argue how the data represents the topic of your interest. In addition to that, some formal requirements are:

- Your data should encompass some **element of comparison**. You might compare, for example, handbooks on psychiatry in 1900 with those from 2000. You might study diplomatic papers on the downfall of the Soviet Union from the Netherlands with those from Germany. Or you might take Thomas Mann's complete oeuvre to see how his wordings and topics change over time. These (random) examples show that the comparison may be over time as well as between domains or geographical regions, and the may be between single or multiple texts.
- You should be able to **download the data** as readable text files onto your own computers – ideally legally. The above examples might sound great, but might prove unfeasible because of lack of readily available digitized data (in the second example) and/or because of copyright restrictions (in the first and third example). Getting your hands on relevant or interesting data can be difficult, because it can be hidden, literally, anywhere on the webosphere. If you have no idea, you can find some suggestions at the end of this manual. Another good starting point is I-Analyzer, a tool built by the UU's Centre for Digital Humanities that hosts a growing number of datasets for exploration and bulk download (as csv-files – see below). As a last resort, newspaper articles always work well for the goals of this module (Nexis Uni, dating back to the mid-1990s).
- This leads us to the third requirement: getting a workable **amount of data**. A general rule of thumb is that digital text analysis has an added value first and foremost when you have more data than you can reasonably read for your research. What does that mean? If your data consists of books, two text files (representing two books) might already do the trick. If you plan to study newspapers, you should rather think in terms of hundreds or thousands of articles. Generally, digital text analysis works better the more data you have. In this respect, there is no upper limit. Your laptop or pc will disagree, however. Therefore, it is important that you balance between having sufficient data to yield robust results, but still small enough to not have to wait hours or days to generate them. Since this tutorial is about learning skills and methods rather than answering research questions, I advise you to start small. What works for small amounts of data will also work for larger amounts, so you can always go for it if time allows—although a minimum of 6 documents (texts, files) would be good to have.
- Lastly, what **format** should your data be in? The notebooks that we will use in class generally work with the plain text format (.txt) or the plain spreadsheet format (.csv). Online, you will not come across the latter very often. The former all the more. This is the standard format that most platforms that allow downloading digital or digitized data offer. If so, always choose .txt, because the data format is small and, again, the notebooks work with them, anyway. Pdf's of text or other text formats (like .doc or .rtf) are also viable formats, because we can convert these to .txt or .csv. If your research project involves comparison over time, the standard format to store your data locally is in **files-per-year** – ideally, with the year as filename. Alternatively, you might (also) put more metadata in the filename (like author or country). If you are

familiar with spreadsheet formats, you are, of course, welcome to put all metadata in the column structure.

Form

You are free to choose the form of this portfolio, but it must contain the following elements (not necessarily in this order):

1) a description and justification of your topic (including some state-of-the-art) and the data that represents it;
2) the weekly assignments aimed at exploring and comparing your data and works/topics within it
3) a reflection on the results from the digital text analyses in terms of what this information adds to the existing literature, as well as a general reflection on the opportunities, but also the risks and pitfalls of quantitative text analysis for this type of historical research.

You will have noticed that this description speaks of 'research topic' rather than 'research question'. Naturally, you are allowed to answer a research question in your paper, but this formulation is to signal that this is not a prerequisite. The aim of this setup is to offer room for experiment, in a reasoned manner, with different techniques.

Assessment

The portfolios are assessed equally on the doing (the quality of the work, specifically the level of application of the methods at hand) as well as on the reflection on what you have been doing (how suitable are data and methodology to study this research topic? What perspectives do they add to the existing scholarship? What do you gain but also lose in a digital environment?).

Notebooks

Can be found on GitHub: https://github.com/PimHuijnen/QTA/tree/main.

Weekly schedule

Week	Seminar	Lab	Portfolio
18 Nov	What is digital text analysis for historians?	Python introduction	Team, topic, data
25 Nov	Working with data	Preprocessing	General statistics
2 Dec	Between statistics & hermeneutics. *Student presentation 1*	Counting words	Frequencies
9 Dec	Extracting meaning. *Student presentation 2*	Collocations, n-grams, part-of-speech, tf-idf	Words in context
16 Dec	AI for historians. *Student presentation 3*	Word embeddings	Semantic relations
13 Jan	*Student presentation 4 & 5*	Check-in, individual feedback	

20 Jan		Check-in, individual feedback	Deadline portfolio: **January 23, 2026, 11:59pm**

18 November: Week 1 – Getting started: What is digital text analysis for historians?

Seminar

In the first seminar, we will discuss what this module is about and how it connects to wider shifts currently shaping the field of history.

Lab

In the first lab session we will start with the basics of Python programming. Before we can use Python code for text analysis, you will have to become familiar with the logic and most common commands of Python programming. You will do so by doing chapter 1 of 'computational humanist' Folgert Karsdorps Python Programming for the Humanities course. In 2021, Karsdorp has overhauled his course in the open access book Humanities Data Analysis, together with Mike Kestemont and Alan Lidell. This book, however, presupposes basic knowledge of Python. Therefore, we turn to his old course in this module, because it is better suited for those altogether unfamiliar with the programming language. Once you have installed Jupyter Notebooks on your computer (see 'preparation' below) you can download chapter 1 here (click on 'download raw file' on the right top) and open it in Jupyter Notebooks.

N.B. You may skip the final assignment of chapter 1. Instead, please do the more relevant assignment at the end of this manual.

Preparations:

Please get yourself acquainted with Jupyter Notebooks before class. It is an interactive format that allows you to code and type in your browser. This is included in the Anaconda distribution. The easiest way to use Jupyter Notebooks, therefore, is to download the Anaconda app to your computer. After installation, an app called Anaconda Navigator will be on your computer. You can create and manage enviromnents and use Jupyter Notebooks from it. Open the app and click on Jupyter Notebooks, after which it will launch in your default browser. See https://programminghistorian.org/en/lessons/jupyter-notebooks if you run into any problems or if you're looking for a gentle introduction to Jupyter Notebooks.

Readings:

- Lässig, Simone. "Digital History: Challenges and Opportunities for the Profession." *Geschichte Und Gesellschaft* 47, no. 1 (June 10, 2021): 5–34. https://doi.org/10.13109/gege.2021.47.1.5.

Further reading (not mandatory):

- Arguing with Digital History working group, 'Digital History and Argument', White Paper (Roy Rosenzweig Center for History and New Media, November 13, 2017).

- Fickers, Andreas, 'Towards A New Digital Historicism? Doing History In The Age Of Abundance', *VIEW Journal of European Television History and Culture* 1:1 (2012) 19–26.
- Hitchcock, Tim, 'Confronting the Digital: Or How Academic History Writing Lost the Plot', *Cultural and Social History* 10:1 (2013) 9–23.
- Madrigal, Alexis C. "The Way We Write History Has Changed." The Atlantic, January 21, 2020. https://www.theatlantic.com/technology/archive/2020/01/smartphone-archives-history-photography/605284/.
- Nicholson, Bob 'The Digital Turn', *Media History* 19:1 (2013) 59–73.

25 November: Week 2 – Working with data

Seminar

In the second seminar we will zoom in on the particularities of quantitative text analysis. What sets computational techniques in historical research apart from conventional scholarship? (See also Lässig last week, Rockwell & Sinclair next week). As a way of introducing this, we will look at menus (http://menus.nypl.org/) as a source for both 'traditional' and digital history.

Lab

This week we will start with the notebooks for digital text analysis that I provide in this GitHub repository. 'Doing things with text 1' is all about one of the most important aspects of text mining: preprocessing. This cleans and restructures your data so that it allows you to find what you are looking for.

Preparations:

Start thinking about a research topic and, if possible, a research question. To make your life easier, pick something that you have worked on in other classes. Try and find an appropriate dataset that represents this topic/question. Finding data is something that I can help you with in class, but you must make sure yourself not to wait for too long to decide what data you would like to use. Ideally, you have a dataset by this week.

Readings:

- Nguyen, D. et.al., 'How we do things with words: analyzing text as social and cultural data', *Frontiers in Artificial Intelligence* 3 (2020), DOI: https://doi.org/10.3389/frai.2020.00062.

Further readings (not mandatory):

- Jurafsky, Dan, Victor Chahuneau, Bryan R. Routledge, and Noah A. Smith, 'Linguistic Markers of Status in Food Culture: Bourdieu's Distinction in a Menu Corpus', *Journal of Cultural Analytics*, October 18 (2016).
- Nielbo, Kristoffer L., Folgert Karsdorp, Melvin Wevers, Alie Lassche, Rebekah B. Baglini, Mike Kestemont, and Nina Tahmasebi. "Quantitative Text Analysis." *Nature*

Reviews Methods Primers 4, no. 1 (April 11, 2024): 25. https://doi.org/10.1038/s43586-024-00302-w.

2 December: Week 3 – Between statistics and hermeneutics

Seminar

In this seminar we will look at some of the underpinnings of computational techniques for digital history on a methodological level (next week we will zoom in on tools & techniques). Central to today's class will be the debate between those who think that sound use of quantitative techniques means adhering to the scientific method (as in hypothesis testing on the basis of statistical significance) and those who believe that quantitative techniques are yet another way of opening up our source material, regardless how we come to our conclusions. In 2019, this debate reached a high point when Nan Z. Da criticized her own field (English literature) for using computational tools way too naively (see the first item in the Further Reading list). The pamphlets by Da and Underwood in the Chronicle of Higher Education can be read as summaries of this debate. Literary scholars dominate the digital humanities in the United States, but it is not difficult to see the implications of this debate for the field of history, as Rockwell and Sinclair (both historians – see further readings) demonstrate.

This week, we will start with the student presentations.

Lab

The notebook 'Doing things with text 2' offers several approaches to getting to know your data: counting the total number of words in your texts and identifying and counting the most frequent words in your texts (in total and by word length). This notebook also contains some basic visualizations of these calculations, which you can use for your portfolio.

N.B.1 If you have not yet managed to find relevant data by today, please use the example text files '1977.txt' and '2015.txt' (the screenplays of Star Wars IV and VII, respectively) from the GitHub repository to practice with the notebooks.

N.B.2 The notebook 'Doing things with text 2' works for a single text file. Notebook 3c allow you to do the same with multiple files (text or csv files, which can be cleaned in notebooks 3a and 3b, respectively). Do note that most of the code in these notebooks will treat all files that you load simultaneously as one single corpus, so make sure not to load any files that you would like to compare with one another.

Readings:

- Da, Nan Z., 'The Digital Humanities Debacle. Computational Methods Repeatedly Come up Short', *Chronicle of Higher Education*, March 27 (2019).
- Gavin, Michael. "Is There a Text in My Data? (Part 1): On Counting Words." *Journal of Cultural Analytics*, January 25, 2020. https://doi.org/10.22148/001c.11830.
- Underwood, Ted, 'Dear Humanists – Fear Not the Digital Revolution', *Chronicle of Higher Education*, March 27 (2019)

Further readings (not mandatory):

- Da, Nan Z., 'The Computational Case against Computational Literary Studies', *Critical Inquiry* 45:3 (2019) 601–39, https://doi.org/10.1086/702594.
- Rieder, Bernhard and Theo Röhle, 'Digital Methods: Five Challenges', in David M. Berry (ed), *Understanding Digital Humanities* (Basingstoke & New York: Palgrave Macmillan, 2012) 86–103.
- Rockwell, Geoffrey and Stéfan Sinclair, *Hermeneutica. Computer-Assisted Interpretation in the Humanities* (Cambridge and London: The MIT Press, 2016), chapter 6: 'A Model Theory: Thinking Through Hermeneutical Things'.
- Online discussion forum on Da's article on the *Critical Inquiry* website (with contributions from, among others, Ted Underwood and Stanley Fish)
- Debates section of the *Journal of Cultural Analytics*, with responses to Da's article.
- Special issue on digital history of *BMGN-LCHR* from 2013, including a forum section with responses to Rens Bod's thesis of the humanities as a pattern-searching domain of knowledge.

Portfolio assignment:

You cannot publish anything based on digital text analysis without being absolutely clear on your data, so this is what we will start with. The question what your data is about should be answered both in a qualitative way—how representative is it for your topic of interest or, in other words: what's in there, what's not?—and in a quantitative one. With regard to the latter, think about the following questions:

- What does your data look like in number of documents (and what do they represent: books? Articles?)
- How coherent is your data: does it cover an entire period or archive, is it a sample?
- How coherent is your data in terms of words-per-document (and, possibly, *unique* words per document, specific types of words per document, etc.)?
- What is the OCR quality (and how can you say something about this)?

You can answer some of these questions based on looking at your data, others on the calculations in the notebooks 'Doing things with text 2' and/or 'Doing things with text 3c'.

Commonly this information is provided in the form of tables, so make sure you add these to your portfolio. If you wish, you can also add some visualizations with the basic statistics to your portfolio: number of documents, number of words/tokens per document, type token ratio, etc. etc.

9 December: Week 4 – Extracting meaning

Seminar

Today, we will talk about the different approaches towards digital text analysis. Broadly speaking, there are two different philosophies: some try and turn the unstructured data that texts usually are into structured data formats that can be counted. Others aim to find ways to study text as language. We will explore both and look at the ways in which basic computational operations like word or n-gram counts and collocations on the one hand (see Moretti & Pestre), and part-of-speech tagging on the other (see Baker et.al.), can provide historians with new and meaningful insights.

The long list of further readings are mostly examples of empirical studies that use some of the techniques – or at least approaches – that are central to today's class. These are primarily meant for inspiration.

Lab – Approaches to digital text analysis

We will continue our exploration of our data with the notebooks Doing things with words 4, 5 & 6. These add some proven functionalities for text analysis, both on the keyword and on the corpus level:

- **Keyword searches** allow you to plot the relative frequency of one or more keywords in a graph (scatter plot). If your data is diachronic, you can see trends in word usage in this way.
- **N-grams** print the top n of phrases (word combinations) of 2, 3, or more words in each of your texts along with their frequency. For some questions this might be more relevant than single words.
- **Collocations** find statistical relations between words within a particular window of words. The meaning of a word can often be deduced from the words that appear strikingly often in its vicinity in a text. There are different measures to compute collocations and our notebook works with three of them (log likelihood, pointwise mutual information and raw frequency). The code prints the top n-collocations for one or more keywords within one or more windows based on each of the algorithms.
- Term frequency – inversed document frequency (**tf-idf**) is a basic statistic that sets texts apart from one another. The tf-idf function in notebook 5 highlights the top n-words that are most unique, specific or distinguishable for a single text (document, file) in your corpus compared to all texts (documents, files) in bar charts. The **cosine similarity** function yields a heat map that shows the similarity between each of the texts in your corpus (the greener, the more alike).
- The **part-of-speech** (POS) tags that the Spacy package adds to each word in the data you load allows you to search for specific types of words (verbs, nouns, proper nouns, adjectives – more are possible). The notebook prints the top n of each of these word types along with their frequency
- The **Named Entity Recognition** (NER) function creates lists of words that are recognized as particular entities: geographical locations, persons, product names, etc. Do realize that the quality of NER is heavily dependent on the language of your data (English always works best).

Preparations:

Think of keywords that might be of interest in the light of your research question. Words that are key to your topic, words that might denote a shift in focus or mentality, and/or words that have strong associations in your dataset that you are interested in. Particularly interesting are words that you can base the comparison within your data (see above, 'On Data') on. You will use these words to interpret the analyses that you will do in this and next weeks' lab sessions.

Readings:

- Baker, P., C. Gabrielatos, and T. McEnery, 'Sketching Muslims: A Corpus Driven Analysis of Representations Around the Word "Muslim" in the British Press, 1998-2009', *Applied Linguistics* 34 (2013) 255-278.

- Moretti, Franco and Dominique Pestre, 'Bankspeak. The language of World Bank Reports', *New Left Review* 92 (2015) 75-99.

Further readings (not mandatory):

- Blaxill, Luke, 'Quantifying the Language of British Politics, 1880-1910', *Historical Research* 86:232 (May 2013) 313–41, https://doi.org/10.1111/1468-2281.12011.
- De Bolla, Peter, *The Architecture of Concepts: The Historical Formation of Human Rights* (New York: Fordham University Press, 2013).
- van Eijnatten, Joris and Pim Huijnen (2021), 'Something Happened to the Future. Reconstructing Temporalities in Dutch Parliamentary Debate, 1814-2018', *Contributions to the History of Concepts* 16, pp. 52-82, DOI: 10.3167/choc.2021.160204.
- Guldi, Joanna, 'The History of Walking and the Digital Turn: Stride and Lounge in London, 1808–1851', *The Journal of Modern History* 84:1 (March 2012) 116–44, https://doi.org/10.1086/663350.
- Kestemont, Mike, Folgert Karsdorp, and Marten Düring, 'Mining the Twentieth Century's History from the Time Magazine Corpus', *Proceedings of the 8th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH)* (2014) 62.
- Klingenstein, S., T. Hitchcock, and S. DeDeo. "The Civilizing Process in London's Old Bailey." *Proceedings of the National Academy of Sciences*, no. Published ahead of print (June 16, 2014). https://doi.org/10.1073/pnas.1405984111.
- McGillivray, Barbara, 'Computational Methods for Semantic Analysis of Historical Texts', in Kristen Schuster and Stuart Dunn (eds.), *Routledge International Handbook of Research Methods in Digital Humanities* (London: Routledge, 2020), pp. 261-274.
- Michel, Jean-Baptiste et al., 'Quantitative Analysis of Culture Using Millions of Digitized Books', *Science* 331 (2011) 176 -182
- Murdock, Jaimie, Colin Allen, and Simon DeDeo, 'Quantitative and Qualitative Approaches to the Development of Darwin's Origin of Species', *Current Research in Digital History* 1 (2018), https://doi.org/10.31835/crdh.2018.14
- Scherrer, Marten et.al., 'The rise and fall of rationality in language', *Proceedings of the National Academy of Sciences* 118, 51 (2021). DOI: https://pnas.org/doi/full/10.1073/pnas.2107848118.
- Twenge, Jean M., W. Keith Campbell, and Brittany Gentile, 'Changes in Pronoun Use in American Books and the Rise of Individualism, 1960-2008', *Journal of Cross-Cultural Psychology* 44: 3 (2013) 406–15.

Portfolio assignment:

Experiment with each of the functionalities in Doing things with words 4, 5 and 6. Decide which of these make most sense for your topic of interest. Zoom in on these functionalities:

- Reflect on your choice for particular functionalities: how can these help studying your topic of interest? How do they relate to each other?
- Make reflected choices for your input: what keywords do you use? What texts do your contrast or combine? Make sure to experiment with different combinations and not necessarily stick to your first choice.
- Fiddle around with different settings: algorithms, window sizes, POS- or NER-categories, top n, etc. Demonstrate that your choices have not been informed by what happened to be default or easiest, but by your interest and expertise.

16 December: Week 5 – AI for historians

Seminar

In this seminar, we will go a step further when it comes to computational techniques and explore the particularities of AI and machine learning. This approach to data science has huge potential (see Gavin et.al.), but implies a major shift in thinking about how science is done that has important consequences (see Kissinger in the Further Reading). Machine learning, therefore, underlines the need for critical reflection in the use of computational techniques – particularly for whom the use of these techniques are not native to their disciplines, like historians. This is what we will do in this seminar (see Koolen et.al.).

Lab – Word embeddings

The final notebook Doing things with words 7 introduces you to machine learning by way of word embeddings (or vector semantics – depending on your field). The notebook lets you model your corpus in your own word embedding space. This allows you to find words with similar embeddings (vectors) to one or more keywords, which indicates semantic similarity. In other words: word embeddings let you find word meanings by showing you synonyms and other similarly used words within a given corpus of texts. Keep adhering to the principle of starting with a small data set (because the model can take long to train), but know that word embeddings work better the more data you put in—so you might want to try that at some point.

Preparations:

Listen to the item 'First Contact' in episode 803 of the famous podcast series This American Life. In the item, TAL editor David Kestenbaum talks to Microsoft scientists about the question whether AI in times of ChatGPT has got to the point that it is starting to have something like human intelligence? Form your own opinion on the conclusions the item offers.

Readings:

- Gavin, Michael et.al., 'Spaces of Meaning: Conceptual History, Vector Semantics, and Close Reading', Matthew K. Gold en Lauren F. Klein (eds.), *Debates in the Digital Humanities* (Minneapolis 2019), 243-267.
- Koolen, Marijn, Jasmijn van Gorp and Jacco van Ossenbruggen, 'Toward a model for digital tool criticism: Reflection as integrative practice', *Digital Scholarship in the Humanities* 34 (2019) 368-385.

Further readings (not mandatory):

- Garg, Nikhil, Londa Schiebinger, Dan Jurafsky, and James Zou. "Word Embeddings Quantify 100 Years of Gender and Ethnic Stereotypes." *Proceedings of the National Academy of Sciences* 115, no. 16 (April 17, 2018): E3635–44. https://doi.org/10.1073/pnas.1720347115.

- Goldberg, Yoav, and Omer Levy. "Word2vec Explained: Deriving Mikolov et al.'s Negative-Sampling Word-Embedding Method." *arXiv:1402.3722 [Cs, Stat]*, February 15, 2014. http://arxiv.org/abs/1402.3722.
- Hamilton, William L., Jure Leskovec, and Dan Jurafsky. "Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change." *Proceedings of ACL*, 2016.
- Kenter, Tom, Melvin Wevers, Pim Huijnen, and Maarten de Rijke. "Ad Hoc Monitoring of Vocabulary Shifts over Time." In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management*, 1191–1200. New York: ACM, 2015.
- Kissinger, Henry A. 'How the Enlightenment Ends', *The Atlantic*, June (2018).
- Tahmasebi, Nina, Lars Borin, and Adam Jatowt. "Survey of Computational Approaches to Lexical Semantic Change." *arXiv:1811.06278 [Cs]*, March 13, 2019. http://arxiv.org/abs/1811.06278.
- Wevers, Melvin and Marijn Koolen, 'Digital Begriffsgeschichte: Tracing semantic change using word embeddings', *Historical Methods: A Journal of Quantitative and Interdisciplinary History* 53 (2020) 226-243, DOI: https://doi.org/10.1080/01615440.2020.1760157.

Portfolio assignment:

Experiment with the functionalities in Doing things with words 7:

- Make reflected choices for your input: what data did you build your model on? What keywords do you use? Make sure to experiment with different combinations and not necessarily stick to your first choice.
- Reflect on how the clustering visualization works for you: what insights do you get from it? How can you make sure you see what you think you see?

13 January: Week 6 – Check-in

After our well-deserved Winter break, we will see each other again this week to check-in, hear the final presentations and get back to work.

20 January: Week 7 – Check-in

No class this week, but do come in if you need a place to work (together) and/or want some feedback from your teacher.

Friday 23 January, midnight: portfolio deadline

Data Repositories

English & other languages

- Hathi Trust: https://www.hathitrust.org/ ("The largest digitized collection of knowledge allowable by copyright law", 18 miljoen items)
- Europeana: https://www.europeana.eu/en
- Project Gutenberg: www.gutenberg.org
- UU Library: https://cdh.uu.nl/2021/09/historical-data-available-via-yoda/
    - Eighteenth Century Collections Online
    - Guardian (1791-1909) & Observer (1910-2003)
    - Nineteenth Century U.K. Periodicals, Module 1
    - Times Digital Archives (1785-2011)
    - Times Literary Supplement (1902-2014)
    - (N.B. most of these datasets are very large and very badly OCR'd)
- Princeton's Curated List of Humanities Datasets: https://cdh.princeton.edu/research/resources/humanities-datasets/
- Historical magazines: https://centurypast.org/magazine-directory/
- Distributed Email of Secrets: https://ddosecrets.substack.com/ (very *very* large datasets from document leaks by transparency advocates)
- Movie scripts: https://imsdb.com/, https://www.simplyscripts.com/, https://gointothestory.blcklst.com/script-download-links-9313356d361c, http://www.script-o-rama.com/snazzy/table.html, http://www.awesomefilm.com/, https://sfy.ru/, https://www.dailyscript.com/movie.html, http://www.moviescriptsandscreenplays.com/
- Pop music lyrics: https://genius.com/

Dutch

- Architectuurtijdschriften: https://tresor.tudelft.nl/tijdschrift/architectuurtijdschriften/
- Delpher Tijdschriften: https://www.delpher.nl/nl/tijdschriften (o.a. Libelle, Margriet, Vrij Nederland, e.v.a.)
- Delpher Boeken: https://www.delpher.nl/nl/boeken (o.a. veel (academische) leerboeken)
- Digitale Bibliotheek der Nederlandse Letteren: www.dbnl.nl
- (Digitale) bronnencollecties Huygens Instituut: https://resources.huygens.knaw.nl/
- Natuurtijdschriften: www.natuurtijdschriften.nl
- Nederlands Tijdschrift voor Geneeskunde: www.ntvg.nl (via UU-proxy, historisch archief raadpleegbaar via zoekfunctie)
- Your teacher has digital archives for you to use of, among other things:
    - Allerhande (1954-1973, 1979-2022)
    - BMGN (ca. 1970-2020)
    - jaarverslagen BVD/AIVD (1946-2019)
    - DSM I-V
    - Economisch Statistische Berichten (1916-2016)
    - De Groene Amsterdammer (1877-1940)
    - Militaire Spectator (1832-2007)
    - Staten Generaal Digitaal (1814-2018)

Final assignment Python tutorial (week 1) – Automatically downloading multiple pdfs from a website

Read the text carefully and follow the assignments in italic.

Getting useful data is a common need for digital historians. Even if data (textual or otherwise) is available in online repositories, it can be a pain to get the entire dataset on your local pc, especially if you have to download each file individually. A possible way of bypassing this is contacting the owners of the data/website and ask for the entire dataset. An alternative is to write a program that does the work for you. This is what you will practice with in this final assignment.

Automatic downloading usually works in two steps: First, you create a list of URLs pointing towards all the files you want to download (which, in our case, will be stored in a dictionary rather than in an actual Python list). Second, you ask your program to pull the file from each URL and save it to your pc. The code for the second step is provided below, the assignment focuses on the first. You can complete this step with the knowledge of Python you've gained in completing Chapter 1 of Folgert Karsdorp's Python tutorial. Specifically, it will involve strings, lists, and dictionaries.

For this assignment, we want to download the complete historical archive of the Dutch environmental magazine *De Kleine Aarde*, which can be found on www.dekleineaarde/archief. The assignment involves two steps: 1) making a list of all the pdfs we want to download, and 2) do the actual downloading. The first step is mandatory for this assignment, the second is voluntary.

Step 1 – Making a list of pdfs to download

The first thing you want to do is to check out the URLs of the actual files (in this case, pdfs). The challenge is to find the regularity in the URLs of all files, because this will enable you to get the full list of URLs with the help of your coding skills. (If the URLs of the files were completely random, you would have to copy-paste them by hand, which would not be faster than to downloading the actual files by hand in the first place.)

Open some of the issues on the De Kleine Aarde *website. What pattern do you recognize in the URLs of these files?*

You will notice that the URLs are very similar. Only the issue names are different. But even within the issue names you can discover a pattern, right?

Create a variable `url` *for the first part of the URL that is the same for each of the issues in the online archive.*

Create a list `seasons` *that contains the variants for all four issues within a single year*

Create an empty dictionary `pdf_dict` *that we will later fill with pairs of issue names (as keys) and URLs to those issues (as values).*

Having done this, we can start writing our program that stores a list of all the URLs in the archive in the dictionary we just created. We opt for a dictionary rather than a list, because it allows us to store additional information with the URLs – in our case, the name of the corresponding issue that we will later use as a file name.

Ok, let's get to work. Our program starts with a loop over all available years in the following manner:

```
For i in range(xxxx,xxxx):
```

You get this first line of your program almost for free, because you need the `range` command that you are unfamiliar with. `Range` does what it says it does: it takes the range between the two numbers provided between the brackets. See for which years the *De Kleine Aarde* archive is available on the website and fill in the xxxxs accordingly (remember to use your knowledge here of how Python handles ranges of numbers).

Start with creating a second for-loop that is nested within the first and that loops over the `seasons` *list and that holds the following commands:*

> *Create a string* `issue` *as a combination of the year (= i) and the item in the* `seasons` *list (N.B. i is an integer. To manipulate the* `issue` *string, you need to change it into a string with the* `str()` *command)*
>
> Now create a string `pdf_url` as a combination of the variables `url` and `issue` that you created and parts of the actual URL that are the same for all the pdfs in the online archive (do not forget the .pdf extension itself!).
>
> Finally, appoint the `pdf_url` string as a value for the `issue` key in the dictionary `pdf_dict` in the manner that you have learned in Karsdorp's tutorial.

Congratulations, you have now successfully created a dictionary with URLs for all issues of *De Kleine Aarde*. You can check for correctness by printing all items one by one in the following manner:

```
For issue, url in pdf_dict.items():
      print(url) # add or change 'url' for 'issue' to check pdf_dict's keys
```

Step 2 – Automatically download all pdfs in pdf_dict

Now that we have a dictionary with all URLs, we can add the second part of our program. This is fully provided for you here, so it should be no more than a matter of copy-pasting. It starts with importing the necessary libraries:

```
import os
import requests
import platform
```

In the next field, copy-paste the following lines of code. This creates a 'headers' variable that we can pass onto the `requests.get()` command in the following lines, which we will use to automatically open online pdfs. The 'headers' variable tricks the website that holds the pdfs

(kleineaarde.nl in our case) into thinking the request to open the pdf comes from a normal browser rather than a script, which the website would block:

```
# Build a dynamic User-Agent string based on the system
system = platform.system()          # e.g. 'Darwin', 'Windows', 'Linux'
release = platform.release()        # e.g. '22.6.0'
user_agent = (
    f"Mozilla/5.0 ({system}; {system} {release}) "
    "AppleWebKit/537.36 (KHTML, like Gecko) "
    "Chrome/120.0.0.0 Safari/537.36"
)
headers = {"User-Agent": user_agent}
```

Next, we create a new dictionary based on `pdf_dict` called `pdf_dict_small`. We do this because we are friendly and do not want to overload kleineaarde.nl.

```
pdf_dict_small = dict(list(pdf_dict.items())[:5])
```

In the final field, we do the actual scraping:

```
Path = 'your/path/here/' # change your/path/here/ for the actual path on your pc
you want to store the pdfs in

for issue, pdf_url in pdf_dict_small.items():
    print('Processing \'' + issue + '\'...')
    r = requests.get(pdf_url, headers=headers) # opens pdf
    with open(os.path.join(Path + str(issue) + '.pdf'), 'wb') as f:
        f.write(r.content) # writes pdf named [issue].pdf in path folder
```

This part of the program starts with a variable 'path' that should be a folder on your pc you would like to store the pdfs of *De Kleine Aarde* in. N.B. It is wise to create a dedicated folder for this. If you want the real thing and use `pdf_dict` instead of `pdf_dict_small`, this program will download 151 files with a total volume of almost 12GB on your computer. It is wise to think about where you want to store these files before running the program!

As you will have recognized, the program continues with a for-loop, looping over the keys and values of our 'pdf_dict' dictionary. For each of the items in the dictionary, the program does the following:

- It starts with printing which item (issue) it is processing
- It automatically opens the pdf behind the URL 'pdf_url' and stores its information in a variable 'r'
- It creates a pdf file with the current issue as its name in the folder you pointed 'path' towards
- It writes the content of the variable 'r' in the file you just created

After having run the entire program, you will end with 151 pdf files on you pc. Next, you can do things like running text recognition over these pdfs, extracting, preprocessing and merging the text, and start analysing the texts. This pipeline has the advantage that it stores the entire pdf-archive on your local pc, but it is not the most efficient (and, potentially, a legal grey zone). A more advanced alternative is to not save each pdf to your pc, but to do the text recognition and extraction of the pdfs within the last for-loop and to store only the text as .txt or .csv files, straining the storage of your pc a lot less. The trade-off is that you will not have access to the original pdfs if the *Kleine Aarde* website ever goes offline.