Big Data and Automated Content Analysis
Part I and II (12 ECTS)

Course Manual

dr. Damian Trilling

Graduate School of Communication
University of Amsterdam

d.c.trilling@uva.nl
www.damiantrilling.net
@damian0604

Office: REC-C, 8th floor

Academic Year 2020/21
Semester 2, block 1 and 2

Coronavirus-Pandemic Online Edition

Chapter 1

About this course

This course manual contains general information, guidelines, rules and schedules for the Research Master course Big Data & Automated Content Analysis Part I and II (12 ECTS). Please make sure you read it carefully, as it contains information regarding assignments, deadlines and grading.

1.1 Course description

"Big data" is a relatively new phenomenon, and refers to data that are more voluminous, but often also more unstructured and dynamic, than traditionally the case. In Communication Science and the Social Sciences more broadly, this in particular concerns research that draws on Internet-based data sources such as social media, large digital archives, and public comments to news and products This emerging field of studies is also called *Computational Social Science* (Lazer et al., 2009) or even *Comutational Communication Science* (Shah, Cappella, & Neuman, 2015).

The course will provide insights in the concepts, challenges and opportunities associated with data so large that traditional research methods (like manual coding) cannot be applied any more and traditional inferential statistics start to loose their meaning. Participants are introduced to strategies and techniques for capturing and analyzing digital data in communication contexts. We will focus on (a) data harvesting, storage, and preprocessing and (b) computer-aided content analysis, including natural language processing (NLP) and computational social science approaches. In particular, we will use advanced machine learning approaches and models like word embeddings.

To participate in this course, students are expected to be interested in learning how to write own programs where off-the-shelf software is not available. Some basic understanding of programming languages is helpful, but not necessary to enter the course. Students without such knowledge are encouraged to follow a (free) online course (such as the one at `https://www.codecademy.com/learn/python`) to prepare.

1.2 Goals

Upon completion of this course, the following goals are reached:

A Students can explain the research designs and methods employed in existing research articles on Big Data and automated content analysis.

B Students can on their own and in own words critically discuss the pros and cons of research designs and methods employed in existing research articles on Big Data and automated content analysis; they can, based on this, give a critical evaluation of the methods and, where relevant, give advice to improve the study in question.

C Students can identify research methods from computer science and computational linguistics which can be used for research in the domain of communication science; they can explain the principles of these methods and describe the value of these methods for communication science research.

D Students can on their own formulate a research question and hypotheses for own empirical research in the domain of Big Data.

E Students can on their own chose, execute and report on advanced research methods in the domain of Big Data and automatic content analysis.

F Students know how to collect data with scrapers, crawlers and APIs; they know how to analyze these data and to this end, they have basic knowledge of the programming language Python and know how to use Python-modules for communication science research.

G Students can critically discuss strong and weak points of their own research and suggest improvements.

H Students participate actively: reading the literature carefully and on time, completing assignments carefully and on time, active participation in discussions, and giving feedback on the work of fellow students give evidence of this.

1.3 Help with practical matters

While making your first steps with programming in Python, you will probably have a lot of questions. Nevertheless, `htttp://google.com` and `http://stackoverflow.com` should be your first points of contact. After all, that's how we solve our problems as well...

Note that to compensate for difficulties arising through online teaching a course like this, there will be online office hours especially during the startup phase of the course offered. This is the place for solving technical problems.

Chapter 2

Rules, assignments, and grading

The final grade of this course will be composed of the grade of two mid-term take home exam (2 × 20%) and one individual project (60%).

2.1 Mid-term take-home exams (2 × 20%)

In two mid-term take-home exam, students will show their understanding of the literature and prove they have gained new insights during the lectures and lab sessions. They will be asked to critically assess various approaches to Big Data analysis and make own suggestions for research.

2.2 Final individual project (60%)

The final individual project typically consists of the following elements:

- introduction including references to relevant (course) literature, an overarching research question plus subquestions and/or hypotheses (1–2 pages);
- an overview of the analytic strategy, referring to relevant methods learned in this course;
- carefully collected and relevant dataset of non-trivial size;
- a set of scripts for collecting, preprocessing, and analyzing the data. The scripts should be well-documented and tailored to the specific needs of the own project;
- output files;
- a well-substantiated conclusion with an answer to the RQ and directions for future research.

2.3 Grading and 2$^{\text{nd}}$ try

Students have to get a pass (5.5 or higher) for both mid-term take-home exams and the individual project. If the grade of one of the mid-term exams is lower, a re-sit will be organized, typically within one week after the grade is communicated to the student. If the grade of the final project is lower, an improved version can be handed in within one week after the grade is communicated to the student. If the improved version still is graded lower than 5.5, the course cannot be completed. Improved versions of the final individual project cannot be graded higher than 6.0.

2.4 Presence and participation

Attendance is compulsory. Missing more than three meetings – for whatever reason – means the course cannot be completed.

Next to attending the meetings, students are also required to prepare the assigned literature and to continue working on the programming tasks after the lab sessions. To successfully finish the course, attending the lab sessions is not enough, but has to go hand-in-hand with continuous self-study.

2.5 Staying informed

It is your responsibility to check the means of communications used for this course (i.e., your email account, but – if applicable – also e-learning platforms or any other tool that the lecturer decides to use) on a regular basis, which in most cases means daily.

2.6 Plagiarism & fraud

Plagiarism is a serious academic violation. Cases in which students use material such as online sources or any other sources in their written work and present this material as their own original work without citation/referencing, and thus conduct plagiarism, will be reported to the Examencommissie of the Department of Communication without any further negotiation. If the committee comes to the conclusion that a student has indeed committed plagiarism the course cannot be completed.

General UvA regulations about fraud and plagiarism apply.

2.7 Deadlines and handing in

Per assignment, the lecturer will specify whether it has to be handed in via Canvas or via Filesender. If no specific instructions are given, or if there is any issue with submitting an assignment through Canvas, use Filesender. Please send all assignments and papers as a PDF file (and do not use formats like .docx) to ensure that it can be read and is displayed the same way on any device. Multiple files should be compressed and handed in as one .zip file or .tar.gz file. Do not email assignments directly but send them via `https://filesender.surf.nl/` to my mail-address. This way, you can also transfer huge files.

Final papers and take-home exams that are not handed in on time, will be not be graded and receive the grade 1. This rule also applies for any other assignment that might be given. The deadline is only met when the all files are submitted.

2.8 Your computer and operating system: your responsibility

Python is a language, not a program, and there are many ways of running Python code. In the past, there were quite some issues as some things work just a little bit different on Windows compared to Linux or MacOS (those two, belong to the UNIX family, are essentially identical for our purposes), which is why we advised students to install a Virtual Machine with Linux, which allows everyone to have the same environment (see the old book at `https://github.com/damian0604/bdaca/blob/master/book/bd-aca_book.pdf`). We experimented last year with letting students more freedom in their choice of an environment, and we want to continue to do so.

But because it is really hard to help with stuff related to your computer, your operating system, your keyboard etc., you need to make sure that the following things apply. Otherwise, you need to fix it – we won't have room for that during the course. So, do this **before the course starts**.

- You need to have a recent version of a Python interpreter installed, and you need to be able to run Jupyter Notebooks. Chapter 1 of van Atteveldt, Trilling, and Arcila Calderón (2021) gives instructions (you can skip the part about R, though). You can either use Python natively or use Anaconda (see Chapter 1), or you can also still opt for the Virtual Machine way as outlined in Trilling (2020). **It is important that *you* know how to use it on your machine.**
- You need to be able to install third-party packages (see Chapter 1). Test this by checking wither you can install the package `shifterator` and then load it with `import shifterator` without getting an error.
- You need to make sure that your keyboard

produces straight quotes `"` rather than typographical ones when typing. Test that by making sure that you can run `print("hello world)"` in a jupyer notebook

- Create a folder for this course at an easy to remember place, such as `/home/damian/bdaca` (Linux), `/Users/damian/bdaca` (MacOS), or `c://Users//damian//bdaca` (Windows). You could potentially also use `c://Users//damian//Desktop//bdaca` (Win) or `/Users/damian/Desktop/bdaca` (Mac) or similar. Make sure that you can locate files in that folder in Jupyter Notebook. And remember/write down it's name! Preferably, don't use spaces (can lead to confusion), and Windows users, replace \ by either a double \\ or by forward slashes (/)

If – even after trying – you do not succeed in any of these steps, ask your classmates or make an appointment with Vladislav as soon as possible.

Chapter 3

Schedule and Literature

The following schedule gives an overview of the topics covered each week, the obligatory literature that has to be studied each week, and other tasks the students have to complete in preparation of the class. In particular, the schedule shows which chapter of van Atteveldt et al. (2021) will be dealt with. Note that some basic chapters that explain how to install the software we are going to use have to be read before the course starts.

Next to the obligatory literature, the following books provide the interested student with more and deeper information. They are intended for the advanced reader and might be useful for final individual projects, but are by no means required literature. Bear in mind, though, that you may encounter slightly outdated examples (e.g., Python 2, now-defunct APIs etc.).

- McKinney, 2012: A lot of examples for data analysis in Python. A PDF of the book can be downloaded for free on `http://it-ebooks.info/book/1041/`.
- VanderPlas, 2016: A more recent book on numpy, pandas, scikit-learn and more. It can also be read online for free on `https://jakevdp.github.io/PythonDataScienceHandbook/`, and the contents are avaibale as Jupyter Notebooks as well `https://github.com/jakevdp/PythonDataScienceHandbook`.
- The pandas cookbook by Julia Evans, a collection of notebooks on github: `https://github.com/jvns/pandas-cookbook`.
- Salganik, 2017: Not a book on Python, but on research methods in the digital age. Very readable, and a lots of inspiration and background about techniques covered in our course.

Changes due to corona-related online reaching. You may have heard from students who took this course in the last years about the general setup: one lecture and one lab session per week. During the lab session, students were working through the chapters in the (old) book, additional ressources, or their own analyses. I was walking around, helping on a 1:1 basis, and when I realized that multiple students had the same problem, I was explaining it plenarily. Over the last years, student evaluations have consistently shown that the format was very much appreciated. At the same time, it is very hard to replicate this format online. This is how we will do it:

- You will need to thoroughly read through the materials **before** each meeting.
- Until Thursday evening, you can submit questions that I will spend some time answering on during the lab sessions.
- During the lab sessions, we will use breakout rooms in which you will can discuss your work and problems with your classmates. The goal here is to develop problem-solving strategies together.
- We will use Zoom's "Remote Support"/"Remote Control" feature, that allows people to take over each other's keyboard, mouse, and screen (of course you

need to approve), so that you can code together.

- A second teacher, Vladislav Petkevich, will assist during the course. Both Vladislav and I will "walk around" the breakout rooms and help out.
- Vlasislav will over online office hours to deal with specific technical problems.

Before the course starts: Prepare your computer.

✔ Chapter 1: Introduction
Make sure that you have a working Python environment installed on your computer. You cannot start the course if you have not done so.

Each week:

- Read the book chapter and/orliterature *before* the Wednesday session
- Submit questions for Friday no later than Thursday evening
- Work on writing code during Friday sessions; ask questions in breakout rooms

PART I: Basics of Python and ACA

Week 1: What is Computational Social Science, and why Python?

Wednesday, 3–2. Lecture.

We discuss what Big Data and Computational (Social—Communication) Science are. We talk about challenges and opportunities as well as the implications for the social sciences in general and communication science in particular. We also pay attention to the tools used in CSS, in particular to the use of Python.

Mandatory readings (in advance): boyd and Crawford (2012), Kitchin (2014), Hilbert et al. (2019).

Additionally, the journal *Commmunication Methods and Measures* had a special issue (volume 12, issue 2–3) about Computational Communication Science. Read at least the editorial (van Atteveldt & Peng, 2018), but preferably, also some of the articles (you can also do that later in the course).

Friday, 5–2. Lab session.

✔ Chapter 2: Fun with data

During the lab session, we will run our first code. We will showcase some possibilities, and leave the technical background for next week.

Week 2: Getting started with Python

Wednesday, 10–2. Lecture.

✔ Chapter 3: Programming concepts for data analysis
You will get a very gentle introduction to computer programming. During the lecture, you are encouraged to follow the examples on your own laptop.

Friday, 12–2. Lab session.

✔ Chapter 4: How to write code
We will do our first real steps in Python and do some exercises to get the feeling.

Week 3: Data formats

We talk about file formats such as `csv` and `json`; about encodings; about reading these formats into basic Python structures such as dictionaries and lists as opposed to reading them into dataframes; and about retrieving such data from local files, as parts of packages, and via an API.

Wedneday, 17–2. Lecture.

✔ Chapter 5: From file to dataframe and back
A conceptual overview of different file formats and data sources, and some practical guidance on how to handle such data in basic Python and in Pandas.

Friday, 19–2. Lab session.

✔ Chapter 12.1: Using web APIs: from open resources to Twitter
We will write a script to collect and handle some JSON data.

Week 4: Data wrangling, simple statistics and visualizations

Of course, you don't need Python to do statistics. Whether it's R, Stata, or SPSS – you probably already have a tool that you are comfortable with. But you also do not want to switch to a different environment just for getting a correlation. And you definitly don't want to do advanced data wrangling in SPSS. . . This week, we will discuss different ways of organizing your data (e.g., long vs wide formats) as well as how to do conventional statistical tests and simple plots in Python.

Wednesday, 24–2. Short lecture plus lab session.

✔ Chapter 6: Data wrangling
We will learn how to do data wrangling with pandas: converting between wide and long formats (melting and pivoting), aggregating data, joining datasets, and so on.

Friday, 26–3. Short lecture plus lab session.

✔ Chapter 7.1. Simple exploratory data analysis
✔ Chapter 7.2. Visualizing data

Week 5: Working with text

In this week, we will dive into how to deal with textual data. How is text represented, how can we clean it, and how can we extract useful information from it?

Wednesday, 3–3. Lecture.

✔ Chapter 9: Processing text
We discuss basic string operations and regular expressions.

Friday, 5–3. Lab session.

You will write a script to conduct a top-down automated content analysis, in which you check for the occurrence of predefined patterns or strings, and extract data from text based on regular expressions.

Take-home exam

In week 5, the first midterm take-home exam is distributed after the Friday meeting. The answer sheets and all files have to be handed in no later than the day before the next meeting, i.e. Tuesday evening (9–5, 23.59).

Week 6: (Clean) representations of text

Text as written by humans usually is pretty messy. You can use some of the techniques you learned last week to clean it up (e.g., to remove punctuation), but in this week, we will dive a bit deeper into ways to represent text in a clean(er) way. We will introduce the Bag-of-Words (BOW) representation and show multiple ways of transforming text into matrices.

Wedneday, 10–3. Lecture with exercises.

✔ Chapter 10: Text as data
This lecture will introduce you to techniques and concepts like stemming, stopword removal, n-grams, word counts and word co-occurrances, and regular expressions. We will do some exercises during the lecture.

Preparation: Mandatory reading: Boumans and Trilling (2016).

Friday, 12–3. Lab session.

You will combine the techiques discussed on Wednesday and write a full automated content analysis script using a top-down dictionary or regular-expression approach.

Week 7: Web scraping and parsing

✔ Chapter 12: Scraping online data

Wednesday, 17–3. Lecture.

We will explore techniques to download data from web pages and to extract meaningful information like the text (or a photo, or a headline, or the author) from an article on `http://nu.nl`, a review (or a price, or a link) from `http://kieskeurig.nl`, or similar.

Friday, 19–3. Lab session.

We will exercise with web scraping and parsing.

Break between block 1 and 2

PART II: Advanced analyses

Week 8: Basics of Machine Learning

In weeks 8 and 9, you will learn how to work with scikit-learn (Pedregosa et al., 2011), one of the most well-known machine learning libraries.

Wednesday, 31–3. Lecture.

✔ Chapter 7.3. Clustering and Dimensionality Reduction
✔ Chapter 8: Statistical Modeling and Supervised Machine Learning
✖ (you can skip 8.4 Deep Learning for now)

We will discuss what unsupervised and supervised machine learning are, what they can be used for, and how they can be evaluated.

Friday, 2–4. No meeting (Good Friday)

Week 9: Supervised Approaches to Text Analysis

In this week, we will combine our knowledge from weeks 4 and 5 with our knowledge from week 8 and use supervised machine learning for text classification.

Wednesday, 7–4.

✔ Chapter 11: Automatic analysis of text
✖ (you can skip 11.5. Unsupervised text analysis: topic modeling for now)

We discuss why and when to choose supervised machine learning approaches as opposed to dictionary- or rule-based approaches (weeks 4 and 5), and discuss how BOW representations can be used as an input for supervised machine learning.

Friday, 9–4. Lab session.

We exercise with supervised machine learning as a technique for automated content analysis.

Week 10: Supervised Approaches to Text Analaysis II

Wednesday, 14–4. Lecture.

We will continue with the topic in week 9, with special attention on how to find the best model using techniques such as crossvalidation and gridsearch.

Friday, 16–4. Lab session

We exercise with supervised machine learning as a technique for automated content analysis.

Week 11: Unsupervised Machine Learning for Text

Wednesday, 21–4. Lecture.

✔ Chapter 11.5. Unsupervised text analysis: Topic modeling

We will discuss Latent Dirichlet Allication (LDA) topic models. We will contrast them with other unsupervised machine learning approaches such as principal component analysis, k-means clustering, and hiearchical clustering.

Mandatory readings (in advance): Maier et al. (2018) and Tsur, Calacci, and Lazer (2015).

Friday, 23–4. Lab session.

You will apply the techniques discussed on Wednesday using gensim (Řehůřek & Sojka, 2010).

Take-home exam

In week 11, the second midterm take-home exam is distributed after the Friday meeting. The answer sheets and all files have to be handed in no later than the day before the next meeting, i.e. Tuesday evening (27–4, 23.59).

Week 12: From word embeddings to deep learning

Wednesday, 28–4

✔ Chapter 10.3.3. Word Embeddings
✔ Chapter 8.3.5. Neural Networks
✔ Chapter 8.4. Deep Learning

In this week, we will talk about a problem of standard forms of ACA: they treat words as independent from each other, and as either present or absent. For instance, if "teacher" is a feature in a specific model, and a text mentions "instructor", then this is not captured – even though it probably should matter, at least to some extend. Word embeddings are a technique to overcome this problem. But also, they can reveal hidden biases in the texts they are trained on.

Mandatory readings (in advance): Kusner, Sun, Kolkin, and Weinberger (2015) and Garg, Schiebinger, Jurafsky, and Zou (2018)

Friday, 30–4

We will apply a word2vec model and get a short introduction to keras.

Week 13: No Teaching

Suggestions for self-study of additional topics in case you want to use them for your final project or your thesis:

✔ Chapter 13 Network data
✔ Chapter 14 Multimedia data
✔ Canvas materials on time series analysis

Wednesday, 5–5: Bevrijdingsdag

Friday, 7–5: Teaching-free day UvA

Week 14: Wrapping up

Wendesday, 12–5. Open Lab

Possibility to ask last (!) questions regarding the final project.

Final project

Deadline for handing in: Friday, 28–5, 23.59.

Literature

Boumans, J. W., & Trilling, D. (2016). Taking stock of the toolkit: An overview of relevant autmated content analysis approaches and techniques for digital journalism scholars. *Digital Journalism*, *4*(1), 8–23. doi: 10.1080/21670811.2015.1096598

boyd, d., & Crawford, K. (2012). Critical questions for Big Data. *Information, Communication & Society*, *15*(5), 662-679. doi: 10.1080/1369118X.2012.678878

Garg, N., Schiebinger, L., Jurafsky, D., & Zou, J. (2018). Word Embeddings as a Lens to Quantify 100 Years of Gender and Ethnic Stereotypes. *Proceedings of the National Academy of Sciences*, *115*(16), E3635–E3644. doi: 10.1073/pnas.1720347115

Hilbert, M., Barnett, G., Blumenstock, J., Contractor, N., Diesner, J., Frey, S., … Zhu, J. J. H. (2019). Computational Communication Science : A Methodological Catalyzer for a Maturing Discipline. *International Journal of Communication*, *13*, 3912–3934.

Kitchin, R. (2014). Big Data, new epistemologies and paradigm shifts. *Big Data & Society*, *1*(1), 1–12. doi: 10.1177/2053951714528481

Kusner, M. J., Sun, Y., Kolkin, N. I., & Weinberger, K. Q. (2015). From Word Embeddings To Document Distances. *Proceedings of The 32nd International Conference on Machine Learning*, *37*, 957–966.

Lazer, D., Pentland, A., Adamic, L., Aral, S., Barabási, A.-L., Brewer, D., … van Alstyne, M. (2009). Computational social science. *Science*, *323*, 721–723. doi: 10.1126/science.1167742

Maier, D., Waldherr, A., Miltner, P., Wiedemann, G., Niekler, A., Keinert, A., … Adam, S. (2018). Applying LDA topic modeling in communication research: Toward a valid and reliable methodology. *Communication Methods and Measures*, *12*(2-3), 93–118. doi: 10.1080/19312458.2018.1430754

McKinney, W. (2012). *Python for data analysis.* Sebastopol, CA: O'Reilly.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Řehůřek, R., & Sojka, P. (2010). Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks* (pp. 45–50). Valletta, Malta: ELRA. (`http://is.muni.cz/publication/884893/en`)

Salganik, M. J. (2017). *Bit by bit: Social research in the digital age.* Princeton, NJ: Princeton University Press.

Shah, D. V., Cappella, J. N., & Neuman, W. R. (2015). Big Data, digital media, and computational social science: Possibilities and perils. *The ANNALS of the American Academy of Political and Social Science*, *659*(1), 6–13. doi: 10.1177/0002716215572084

Trilling, D. (2020). Doing computational social science with Python: An introduction. Version 1.3.2. *SSRN*. Retrieved from `https://github.com/damian0604/bdaca/blob/master/book/bd-aca_book.pdf`

Tsur, O., Calacci, D., & Lazer, D. (2015). A Frame of Mind: Using Statistical Models for Detection of Framing and Agenda Setting Campaigns. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing* (pp. 1629–1638). ACL.

van Atteveldt, W., Trilling, D., & Arcila Calderón, C. (2021). *Computational analysis of communication: A practical introduction to the analysis of texts, networks, and images with code examples in Python and R.* Hoboken, NJ: Wiley.

van Atteveldt, W., & Peng, T. Q. (2018). When Communication Meets Computation: Opportunities, Challenges, and Pitfalls in Computational Communication Science. *Communication Methods and Measures*, *12*(2-3), 81–92. doi: 10.1080/19312458.2018.1458084

VanderPlas, J. (2016). *Python data science handbook: Essential tools for working with data.* Sebastopol, CA: O'Reilly.