TEXT ANALYSIS FOR SOCIAL SCIENCES IN PYTHON

(MSc in Economics, WiSo Universität Hamburg)

Winter Semester 2021 – 2022

Instructor : Huyen Nguyen

Website: http://huyenttnguyen.com

E-mail: huyen.nguyen-1@uni-hamburg.de

COURSE SUMMARY

We are living in a rapidly digitized world, with an ever-increasing availability of large-scale textual corpora in law, politics and economics. This massive data development scene poses exciting challenges for social scientists, to understand the fabrics and functioning of our societies, beyond just numbers. Coupling the proliferation of legal and political corpora with the speedy growth of data science toolkits, we have at hand a powerful infrastructure to extract hidden novel insights about relevant institutional and human patterns in texts.

This course gives comprehensive introduction to the basic theory and hands-on applications of text analysis and machine learning for social science in Python. The course begins with quick introduction to Python languages and moves on to the challenge of representing texts as data. Next, it gives an overview of key techniques to clean texts, extract relevant information, and represent documents as vectors. These techniques include, but not limited to, for instance, measuring document similarity, clustering documents based on topics, as well as visualization methods such as word clouds and spatial relation plots between documents. Students are also provided with various sources to different text corpora, tips and techniques to query a programming issue online and self-study materials to deepen their understanding beyond the scope of the course.

Finally, we consider text-based prediction problems. For instance, given the evidence of a particular case, how will a judge decide on sentences? Given recorded speeches and transcripts of politicians, how ideological is a politician? Such predictions are then incorporated into social science analysis, Students will investigate and implement the relevant machine learning tools for making these types of predictions, including regression, classification, and deep learning models. If time permits, we will also touch upon causal inference methods using texts, either as treatment or outcome in a given data context.

Acknowledgement: The structure and materials of this course are drawn on a combination of pioneering text analysis courses of Elliot Ash (ETH Zurich), Brandon Stewart (Princeton), William Lowe (Hertie School) and Caroline Le Pennec-Caldichoury (HEC Montreal).

COURSE OBJECTIVES

- Comprehensive overview of contemporary approaches in using Python to analyze text as data, and how it is applied in social science policy questions.
- Familiarity with statistical and practical issues around textual data.
- Ability to fit, interpret and apply the basic classes of text cleaning and analysis techniques to independent research projects.

PRE-REQUISITES

This course requires a basic understanding of Python, statistics, probability theories and applied econometric techniques used in social sciences. The first three sessions will quicky introduce you to Python basics and essential packages for text analysis, then we will quickly move onto text as data and relevant methods.

ALL students are required to check their math and stats background. you don't have a solid background in calculus, linear algebra, and probability, read part 1 from this online book.

GETTING STARTED

1) **Python**

The examples in the course will mainly use Python. Additionally, when we move to regression exercises, Stata might be used in economic examples. You are strongly recommended to install and set up Python before the course, using the instructions in the following links:

Python Setup Instructions
Codecademy Online Python Course

2) **Jupyter Notebook**

Read the following step-by-step blog to get familiar with Jupyter Notebook.

https://medium.com/codingthesmartway-com-blog/getting-started-with-jupyter-notebook-for-python-4e7082bd5d46

For installation and trying out Jupyter Notebook, read this:

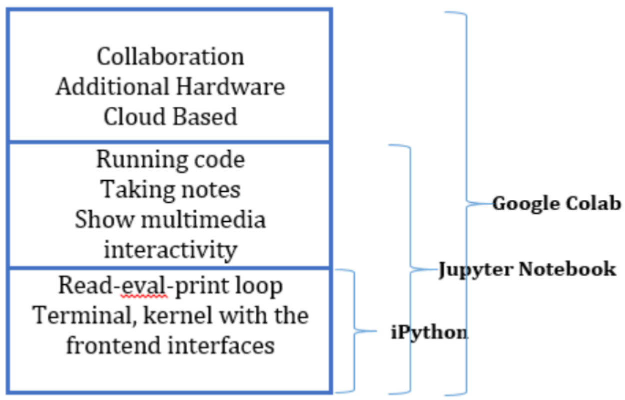
https://jupyter.readthedocs.io/en/latest/install/notebook-classic.html

https://satyaborg.com/posts/accessing-virtualenv-in-jupyter-notebooks

3) **Google Colab**

Google Colab is a specialized version of Jupyter Notebook, which runs on the cloud and offers free computing resources. It does not require a setup, plus the notebooks that you will create can be simultaneously edited by your team members – in a similar manner you edit documents in Google Docs. The greatest advantage is that Colab supports most popular machine learning libraries which can be easily loaded in your notebook. In order to use Google Colab, you will need a Google Account. Your files are by default linked to your Google Drive. For more information and instructions, please check the following link: https://colab.research.google.com/?utm_source=scs-index

The following diagram summarizes the relationship between iPython, Google Colab and Jupyter Notebook:



4) **StackOverflow** https://stackoverflow.com/
Your go-to website (in addition to Google) for ANY programming bug, installation, package issues. Good programming and data wrangling skills = good Googling/StackOverflow skills.

NOTE! I will IGNORE any questions and e-mails related to programming/installation/package issues. The answers are highly likely available on StackOverflow or from your classmates (use Slack forum to ask, or ask them directly in class).

COURSE LOGISTICS & HOUSE RULES

1) **Lectures** are held in the computer room **WiWi 1005** on campus **every Wednesday** morning, from 13.10.2021 to 26.01.2022.

The *only exception* is Week 8 (i.e. Wed 08/12 session). During this week, there will be individual group consultation appointments via Zoom to discuss your mid-term project proposal assignment (more details below).

Every lecture consists of three parts: 45' of lecture → 10' break → 45' code exercise (self) → 5' break → 45' code exercise (team). Classes start at 8.00 AM sharp.

Students are expected to read the required readings and do assignments before class, as well as actively participating in all classes. Attendance on at least 11/14 sessions are required.

2) **Course materials** (slides, exercise links, data sources, tips) are uploaded **on GitHub** https://github.com/httn21uhh/Text-Analysis-for-Social-Sciences-in-Python
   - ***Lecture slide*** will be added to GitHub → Lecture Slides folder **every Thursday morning** of the same week, unless otherwise noticed. For the list of required and readings
   - ***Exercises*** (on Jupyter notebook/Google Colab) that are used in class (and accessible after class for your own practice) will be updated here and/or in Stine/Open Olat.
   - ***Data sources and tips*** are updated regularly as the course proceeds.

3) **Course forum** (introduction, teammate mix-and-match, exchange of ideas, code bugs) are **on Slack:** https://join.slack.com/t/textanalysisf-eia6533/shared_invite/zt-wkqywq13-3OQMM~S~gShngJ~VcTP2~Q

   How to join in this Slack channel? Use your @uhh account to create an account, register and log into the channel (Desktop apps are also downloadable) using the above link.

   NOTE (!) Do NOT share Slack channels to any non-course participants (!) Any requests must go through me first. Otherwise, I will personally kick them out and you will receive a minus point for doing so.

   There are 4 channels, as follows:

   - *#introduce-yourself*: [COMPULSORY, during 1st session] Introduce yourself, your research interest and your goals with the course
   - *#group-matching* [COMPULSORY, deadline Wednesday 20.10]: For the project mid-term proposal and final match yourself to groups found on the Google Sheet in the top of the link (or use the link here). (!) DO NOT CHANGE any details of other classmates.
   - *#qna-code-bugs*: The Q&A space for code issues, solutions, Stackoverflow materials and any useful exercises you found.
   - *#research-projects*: The Q&A space across groups to discuss research project ideas, proposals, planning, and recording resources.

NOTE: Announcement and notices will be done mainly via STINE, as well as the #announcement channel in Slack forum.

COURSE GRADING POLICY

The course will be graded on the usual grading scale with passing grades from 1.0 (very good) to 4.0 (sufficient), and with a failing grade 5.0 (insufficient). The grades are determined as follows:

30%: Midterm coding research pitch (5') + 1-page research proposal
(Deadline: 23.59 Wednesday 24.11)

70%: Final group research project oral presentation (25') (exact date to be scheduled)

Additionally, all students are required to submit a confidential written contribution of him/her/themselves and what other group members of his/her/their own team have contributed. In general, free-riding and foul play are never an option in joint projects.

Submission rule: For every hour of late submission of the midterm pitch & proposal, everyone in the team will receive a 5% reduction of the score. Should you have verifiable extenuating circumstances (e.g. illness, personal loss, hardship, or caring duties), an extension can be granted. In such cases, please contact the course instructor as soon as possible before the deadline.

You can find more detailed instructions on the pitch, proposals, the consultation sessions and final oral presentations in a separate file in Github/Open Olat.

COURSE SYLLABUS & READING MATERIALS

This detailed syllabus is subject to changes as the course proceeds. I will update it regularly on Open Olat and Github repository, please check it on a weekly basis.

Required readings are to be read and analysed thoroughly. Optional readings [OPT] are intended to broaden your knowledge in the respective area and it is recommended to at least get a gist of these materials. We will go through the step-by-step exercises together in class, and you are required to go through any remaining parts on your own before the next lecture.

Week	Date & location	Topic & Content	Readings	Exercises
1	Wed 13.10	Welcome & Installations Python basics – variables, expressions, statements, conditions, functions	A beginner's guide to Python https://wiki.python.org/moin/BeginnersGuide	session 1 – 3, Python Open Lab colab_tutorial.ipynb (part 1.1) (available after class) [OPT] Check this free course on Practical Python Programming
2	Wed 20.10 DEADLINE @ 23.59: group matching	Python basics –, loops, logical operators, strings, list vs. dict, file handling	(same as above)	session 4- 9 and 11, Python Open Lab colab_tutorial.ipynb (part 1.2) (available after class)
3	Wed 27.10	Python basics – numpy, panda, nltk, scikit-learn	TBD (mostly examples illustrating the usage of these different packages)	session 12 Python Open Lab + intro to nltk.ipynb

				+ colab_tutorial.ipynb (part 2) intro to scikit-learn overview numpy tutorial and pandas tutorial.
4	Wed 03.11	Text as data – state-of-the-art technique overview	Gentzkow, Kelly, and Taddy, "Text as Data." Grimmer and Stewart, "Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts." Denny and Spirling (2018), "Text Preprocessing for Unsupervised Learning: Why It Matters, When It Misleads, and What to Do about It"	Check Chapter 4 to 6 of the NLTK book
5	Wed 10.11 Note: Consultation session sign-up in Week 8 available	Text as data - obtaining and preprocessing corpora	Webscraping Twitter data https://towardsdatascience.com/hands-on-web-scraping-building-your-own-twitter-dataset-with-python-and-scrapy-8823fb7d0598 An example to webscrape with Selenium package https://towardsdatascience.com/how-to-use-selenium-to-web-scrape-with-example-80f9b23a843a Text formatting with Regex https://medium.com/factory-mind/regex-tutorial-a-simple-cheatsheet-by-examples-649dc1c3f285	Step-by-step preprocessing example https://medium.com/hackerdawn/the-simple-nlp-text-preprocessing-tutorial-you-were-waiting-for-2e6b082962b2

			https://medium.com/factory-mind/regex-cookbook-most-wanted-regex-aa721558c3c1	
6	Wed 17.11	Supervised method – tf-idf & Bag-of-word approach (e.g. Sentiment analysis)	Econometrics and sentiment analysis https://core.ac.uk/display/323232776?source=2 **(Econometrics meets sentiment : an overview of methodology and applications 2020)** Sebastian Raschka, "Turn your Twitter Timeline into a Word Cloud	TBD
7	Wed 26.11 DEADLINE @ 23.59: 5' pitch recording + 1-page research proposal	Supervised method – Dictionary-Based approach	TBD	TBD
8	Wed 01.12 (consultation week)	(further self-practice on supervised method examples)	TBD	TBD (self-practice exercises)
9	Wed 08.12	Unsupervised Method – Space, tf-idf and cosine similarity	Document distance Leon Derczynski, "Collocations," http://www.derczynski.com/sheffield/teaching/inno/7c.pdf Yoav Goldberg and Omer Levy, "Word2Vec explained: Deriving Mikolov et al's Negative Sampling Word Embedding Method". Piero Molino, "Word embeddings: Past, present, and future".	Andy Thomas, A Word2Vec Keras Tutorial

			Matt Kusner, Yu Sun, Nicholas Kolkin, and Killian Weinberger, "From word embeddings to document distances". [OPT] Garg et al 2018, Word embeddings quantify 100 years of gender and ethnic stereotypes	
10	Wed 15.12	Unsupervised Method - Topic Models	Brandon Rose, "Document clustering in python. Blei, Ng and Jordan, 2003. "Latent Dirichlet Allocation," *Journal of Machine Learning Research*.	Shivam Bansal, "Beginner's guide to topic modeling in Python."
Christmas holiday (19.12.2021 – 02.01.2022)				
11	Wed 05.01.22	Machine Learning: Classification Understand the document classification task, evaluate classification models, evaluate, and deal with errors	Spiess and Mullainathan, Machine Learning: An Applied Econometrics Approach, *Journal of Economic Perspectives*. [OPT] Geron's book Chapter 4	TBD
12	Wed 12.01.22	Machine Learning: Dimensionality Reduction & Feature Selection	Maitra and Yan, PCA and PLS for Regression * Feature Selection in Scikit-Learn	TBD
13	Wed 19.01.22	Causal inference – Text as Treatment/Confounders	Text and Causal Inference: A Review of Using Text to Remove Confounding from Causal Estimates Katherine A. Keith, David Jensen, and Brendan O'Connor How to Make Causal Inferences Using Texts Naoki Egami, Christian J. Fong, Justin Grimmer,	TBD

			Margaret E. Roberts, and Brandon M. Stewart [OPT] A comprehensive collection of research papers on causal inferences and language https://github.com/causaltext/causal-text-papers	
14	Wed 26.01.22	Causal inference – Text as Mediators/Outcome	Causal Mediation Analysis for Interpreting Neural NLP: The Case of Gender Bias Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer and Stuart Shieber A Survey of Online Hate Speech through the Causal Lens Antigoni M. Founta, Lucia Specia	TBD
Exam	TBA	Group project oral presentation (25')		

SUPPLEMENTARY MATERIALS

The course will be mainly based on the weekly lecture slides and in-class exercises, but the following books and code exercises can be used as reference along with the slide content.

· Natural Language Processing in Python, Third Edition, available at nltk.org/book.

· Aurelien Geron, Hands-On Machine Learning with Scikit-Learn & TensorFlow, O'Reilly 2017 (link)

. Jupyter notebooks Github for Geron's book.

· Google Developers Text Classification Guide (This guide contains some practical tips and code examples for using text data)

. For Python syntax programming, this book Fluent Python (O'Reilly 2015) serves as a good guideline https://www.oreilly.com/library/view/fluent-python/9781491946237/