Hands-On Text Coding with Large Language Models for Social Scientists

Version 2024-06-12

Instructor:	Hauke Licht, Ph.D.
Email:	hauke.licht@wiso.uni-koeln.de
Meetings:	June 7, 2024, 14:00 - 17:30 @ IBW building, Room 3.40
	June 14, 2024, 14:00 - 17:30 @ IBW building, Room 3.40

This workshop equips political science, communication science, sociology, and economics researchers with the skills to use Large Language Models (LLMs) like OpenAI's GPT 4.0 or META Research's LlaMa 3 models for automated text coding and annotation. Text coding and annotation refers to a set of tasks applied in quantitative text analysis that involve assigning whole documents, sentences, or phrases and words in them to distinct categories (e.g., positive/negative/negative sentiment or populist/non-populist statement). Due to the success of generative Artificial Intelligence, researchers can rely on LLMS to take over this task that has traditionally been performed by experts, trained coders, or crowd workers.

Through short lectures, practical exercises, and group discussions, participants will gain the practical skills and necessary theoretical understanding to apply LLMs for text coding and annotation in their research. The workshop's key learning objectives are: Understanding how LLMs can be integrated into the content-analytic text annotation process (from conceptualization to writing prompts to testing and validation). Learning key techniques and approaches for instructing an LLM to annotate or code texts, including prompt engineering and in-context learning. And acquiring the skills to automate text coding and annotation tasks with closed- and open-source LLMs using R.

Practical experience with manual text coding or annotation is advantageous but not required.

Course Description

Large Language Models (LLMs) like OpenAi's ChatGPT, Anthropic's Claude, or META Reseach's LlaMa 3 transform our societies in various domains and application areas. This transformation also concerns the social sciences, where one of LLMs' many potentials lies in opening up many new approaches to automated text analysis.

Text coding and annotation are two such application areas. Text coding and annotation refer to the task of assigning documents, sentences, or phrases and words in them to distinct categories (e.g., positive/negative/negative sentiment or populist/non-populist statement). Pioneering studies in the computer and social sciences demonstrate that researchers can use generative LLMs to obtain text-based measures from political texts, social media posts, etc., by instructing them to perform text coding and annotation tasks (e.g., Burnham, 2023; Gilardi, Alizadeh, and Kubli, 2023; Lupo, Magnusson, Hovy, Naurin, and Wängnerud, 2023; O'Hagan and Schein, 2023; Weber and Reichardt, 2023; Ziems, Held, Shaikh, Zhang, Yang, and Chen, 2023). While there are still many concerns about the reliability of this approach (cf. Palmer, Smith, and Spirling, 2024; Reiss, 2023), these studies overall underline that using LLMs for automated text analysis is a crucial skill for researchers to stay at the forefront of their disciplines.

This workshop provides applied researchers in political science, communication science, sociology, and economics with a comprehensive understanding and practical skills in using LLMs for text coding and annotation tasks. Throughout the workshop, participants will engage in interactive sessions, practical exercises, and discussions to ensure a thorough understanding of the relevant concepts and their application in real-world research scenarios. The workshop is specifically designed to bridge the gap between traditional manual text coding methodologies and the innovative potentials of LLMs. It provides researchers with the theoretical understanding and practical skills necessary to apply these transformative technologies in their work. The workshop thus demonstrates how to integrate LLMs into the traditional content analysis process.

Participants are encouraged to bring examples and concrete application ideas from their research to the workshop. Practical experience with manual text coding or annotation is advantageous but not required. Please refer to the *Prerequisites* and *Requirements* sections on the following page for details.

References

M. Burnham (2023). *Stance Detection With Supervised, Zero-Shot, and Few-Shot Applications*. DOI: 10.48550/arXiv.2305.01723

F. Gilardi, M. Alizadeh, and M. Kubli (2023). "ChatGPT outperforms crowd workers for text-annotation tasks". In: *Proceedings of the National Academy of Sciences* 120.30, e2305016120. DOI: 10.1073/pnas.2305016120

L. Lupo, O. Magnusson, D. Hovy, E. Naurin, and L. Wängnerud (2023). *How to Use Large Language Models for Text Coding: The Case of Fatherhood Roles in Public Policy Documents*. en

S. O'Hagan and A. Schein (2023). *Measurement in the Age of LLMs: An Application to Ideological Scaling*

A. Palmer, N. A. Smith, and A. Spirling (2024). “Using proprietary language models in academic research requires explicit justification”. en. In: *Nature Computational Science* 4.1, pp. 2–3. DOI: 10.1038/s43588-023-00585-1

M. V. Reiss (2023). *Testing the Reliability of ChatGPT for Text Annotation and Classification: A Cautionary Remark*. DOI: 10.48550/arXiv.2304.11085

M. Weber and M. Reichardt (2023). *Evaluation is all you need. Prompting Generative Large Language Models for Annotation Tasks in the Social Sciences. A Primer using Open Models*. DOI: 10.48550/arXiv.2401.00284

C. Ziems, W. Held, O. Shaikh, Z. Zhang, D. Yang, and J. Chen (2023). “Can Large Language Models Transform Computational Social Science?” en. Working Paper

Learning objectives

1. Participants know how to integrate LLMs for text coding and annotation into the traditional content-analytic text analysis process.
2. Participants have a basic understanding of how LLMs function and their characteristics.
3. Participants know the differences between various LLM-based text coding and annotation approaches, including prompt engineering and in-context learning.
4. Participants can automate text classification and annotation tasks using the R programming language. This includes:
   (a) writing and optimizing LLM instructions ("prompts") for typical text annotation tasks
   (b) implementing text annotation tasks using OpenAI’s GPT models and the `openai` R package or open-source alternatives with the `ollama` framework and the `rollama` R package
   (c) model- and use-case-specific cost estimation
   (d) awareness of best practices for reproducible use of OpenAI’s GPT models
5. Participants are familiar with open-source alternatives to the paid use of OpenAI’s GPT models.

Prerequisites

1. **A strong interest in text-based measurement** of socio-political or historical phenomena that can be documented in human communication (e.g., in political speeches, print media, user comments in social media, open responses in surveys, etc.). Prior practical experience with manual text coding or annotation is advantageous but *not* strictly necessary.
2. **A basic knowledge of fundamental quantitative content analysis methods**, especially manual coding and automated text classification.
3. **Basic R programming skills**, including importing and exporting tabular data such as CSV or Excel files; creating and manipulating vectors, lists, and data frames; and writing `for`-loops.

Requirements

1. **Readings:** Mandatory readings are marked with two violet exclamation marks (**!!**) in the list below. All other readings simply serve as inspiration or for technical background.
2. **Hard- and software:** Participants must bring their own laptop ($\geq$ Windows 10) or MacBook ($\geq$ macOS 11 Big Sur) with
   (a) at least 10 GB free hard drive storage space
   (b) a working R installation ($\geq$ 4.2.0)
   (c) a working RStudio installation (or another IDE like VS Code with the required extensions)
   (d) a working installation of `ollama`
   (e) working installations of the following R packages: `openai` ($\geq$ 0.4.1), `rollama` ($\geq$ 0.1.0), `metrica` ($\geq$ 2.0.3), `readr` ($\geq$ 2.1.5), `dplyr` ($\geq$ 1.1.4), `purrr` ($\geq$ 1.0.2), `ggplot2` ($\geq$ 3.5.1), and `stringr` ($\geq$ 1.5.1)
3. **An activate OpenAI account** (sign-up at `https://platform.openai.com/signup`), ideally an OpenAI *Plus* subscription (costs U.S. $ 20/month, login and go to *Pricing*), and a credit balance of U.S. $ 5-10 (see here).
4. **Data:** Participants are encouraged to bring a text data set suitable for their application/use case to the workshop. However, the instructor will also provide example data sets for everyone's use.

Course Dates and Times

The workshop will be held on two afternoons (14:00 - 17:30) in the first and second weeks of June 2024 at the University of Cologne:

1. on June 7, 2024, 14:00 - 17:30 @ IBW building, Room 3.40
2. on June 14, 2024, 14:00 - 17:30 @ Seminar building, Room S23

Course Outline

Day 1

14:00 – 14:45 Text annotation: key concepts, tasks, and best practices

This session provides an overview of the key ingredients of content-analytic text analysis. We begin with a quick history that traces the journey from manual content analysis through the development of quantitative text analysis to the advent of Large Language Models (LLMs). We will then focus on the key ingredients of content analysis: concept definition, creating a coding scheme, and developing precise coding instructions.

In exercises, participants will review and critically assess the coding instructions and schemes applied in various existing quantitative text analysis studies.

Background readings

- for an overview of quantitative text analysis methods, see

  J. Grimmer and B. M. Stewart (2013). "Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts". English. In: *Political Analysis* 21.03, pp. 267–297. DOI: 10.1093/pan/mps028

  J. W. Boumans and D. Trilling (2016). "Taking Stock of the Toolkit: An overview of relevant automated content analysis approaches and techniques for digital journalism scholars". en. In: *Digital Journalism* 4.1, pp. 8–23. DOI: 10.1080/21670811.2015.1096598

  W. van Atteveldt and T.-Q. Peng (2018). "When Communication Meets Computation: Opportunities, Challenges, and Pitfalls in Computational Communication Science". en. In: *Communication Methods and Measures* 12.2-3, pp. 81–92. DOI: 10.1080/19312458.2018.1458084

  M. Gentzkow, B. Kelly, and M. Taddy (2019). "Text as Data". English. In: *Journal of Economic Literature* 57.3, pp. 535–574. DOI: 10.1257/jel.20181020

  M. Schoonvelde, G. Schumacher, and B. N. Bakker (2019). "Friends With Text as Data Benefits: Assessing and Extending the Use of Automated Text Analysis in Political Science and Political Psychology". English. In: *Journal of Social and Political Psychology* 7.1, pp. 124–143. DOI: 10.5964/jspp.v7i1.964

  B. Bonikowski and L. K. Nelson (2022). "From Ends to Means: The Promise of Computational Text Analysis for Theoretically Driven Sociological Research". en. In: *Sociological Methods & Research* 51.4, pp. 1469–1483. DOI: 10.1177/00491241221123088

  K. L. Nielbo, F. Karsdorp, M. Wevers, A. Lassche, R. B. Baglini, M. Kestemont, and N. Tahmasebi (2024). "Quantitative text analysis". en. In: *Nature Reviews Methods Primers* 4.1, pp. 1–16. DOI: 10.1038/s43586-024-00302-w

- on text/document classification, see

  D. Hillard, S. Purpura, and J. Wilkerson (2008). "Computer-Assisted Topic Classification for Mixed-Methods Social Science Research". In: *Journal of Information Technology & Politics* 4.4, pp. 31–46. DOI: 10.1080/19331680801975367

  V. D'Orazio, S. T. Landis, G. Palmer, and P. Schrodt (2014). "Separating the Wheat from the Chaff: Applications of Automated Document Classification Using Support Vector Machines". English. In: *Political Analysis* 22.2, pp. 224–242. DOI: 10.1093/pan/mpt030

  P. Barberá, A. E. Boydstun, S. Linn, R. McMahon, and J. Nagler (2021). "Automated Text Classification of News Articles: A Practical Guide". English. In: *Political Analysis* 29.1, pp. 19–42. DOI: 10.1017/pan.2020.8

- on multi-label classification, see A. Erlich, S. G. Dantas, B. E. Bagozzi, D. Berliner, and B. Palmer-Rubin (2022). "Multi-Label Prediction for Political Text-as-Data". en. In: *Political Analysis* 30.4, pp. 463–480. DOI: 10.1017/pan.2021.15
- on token classification for entity detection, see H. Licht and R. Sczepanski (2023). *Who are they talking about? Detecting mentions of social groups in political texts with supervised learning*. en-us. DOI: 10.31219/osf.io/ufb96

14:45 – 16:00 Prompt engineering

Next, focus the workshop's main topic: using LLMs for text coding and annotation. We will begin very practically, making our first steps in instructing a GPT model for text coding. Specifically, we will focus first on *prompt engineering*. Prompt engineering means writing an instruction that tasks an LLM to generate a response for some user input without prior task-specific training. In its application to text coding or annotation, prompt engineering requires (i) translating one's coding instructions into a prompt and (ii) providing the LLM with texts it should classify or annotate one at a time.

In the exercise, participants will focus on translating the coding instruction and scheme they have reviewed in the first session into a prompt and test it interactively in the OpenAI Playground.

Background readings

- Gentle introduction:

  !! P. Törnberg (2024). *Best Practices for Text Annotation with Large Language Models*. DOI: 10.48550/arXiv.2402.05129

  !! M. Weber and M. Reichardt (2023). *Evaluation is all you need. Prompting Generative Large Language Models for Annotation Tasks in the Social Sciences. A Primer using Open Models*. DOI: 10.48550/arXiv.2401.00284

- Applied social science papers:

F. Gilardi, M. Alizadeh, and M. Kubli (2023). "ChatGPT outperforms crowd workers for text-annotation tasks". In: *Proceedings of the National Academy of Sciences* 120.30, e2305016120. DOI: 10.1073/pnas.2305016120

M. Burnham (2023). *Stance Detection With Supervised, Zero-Shot, and Few-Shot Applications*. DOI: 10.48550/arXiv.2305.01723

L. Lupo, O. Magnusson, D. Hovy, E. Naurin, and L. Wängnerud (2023). *How to Use Large Language Models for Text Coding: The Case of Fatherhood Roles in Public Policy Documents*. en

M. V. Reiss (2023). *Testing the Reliability of ChatGPT for Text Annotation and Classification: A Cautionary Remark*. DOI: 10.48550/arXiv.2304.11085

C. Ziems, W. Held, O. Shaikh, Z. Zhang, D. Yang, and J. Chen (2023). "Can Large Language Models Transform Computational Social Science?" en. Working Paper

16:15 – 17:00 Automation with R

We will follow the prompt engineering exercises from the first workshop day with guided exercises on automating prompt-based LLM text coding tasks in R. We will begin by exploring OpenAI's API to understand the structure of its in- and outputs. You will learn how to obtain and safely store your OpenAI API token. And you will learn how to send API requests and receive responses with R.

In this session's exercise, participants will replicate their interactive prompting in the OpenAI Playground with code in R.

Required readings and packages

- install the `openai` R package ($\geq$ 0.4.0), see https://irudnyts.github.io/openai
- read the documentation of the `create_chat_completions` function

17:00 – 17:30 Evaluation and reproducibility

We will follow the hands-on exercises with an overview of current best practices for evaluation and reproducible research. This will include a review of standard classification evaluation metrics like the F1 score and learning about important hyper-parameters that govern an LLM's generative behavior and its reproducibility.

This session's exercise will focus on implementing these best practices.

Day 2

14:00 – 15:00 Looking under the hood

To understand why and how LLMs can perform complex tasks such as text classification only based on prompt, we will conclude the first workshop day with a peak at the methodology underpinning Large

Language Models (LLMs). We will learn about the differences between *masked* and *causal* language modeling – exemplified by contrasting BERT (*Bidirectional Encoder Representations from Transformers*) with GPT (*Generative Pre-trained Transformer*) models. We will then cover the concept of language model pre-training and its relation to "transfer learning." This forms the foundation for understanding the motivation behind using pre-trained models in various language processing tasks.

Lastly, we will explore the evolution of generative models. A special emphasis is placed on the development from earlier GPT models to today's more advanced chat assistants. In this context, we will learn about the "Reinforcement Learning from Human Feedback" (RLHF) framework responsible for the success of LLM-based chat assistants.

Methodological background readings

A. Radford and K. Narasimhan (2018). "Improving Language Understanding by Generative Pre-Training". URL: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. DOI: 10.48550/arXiv.1810.04805

T. B. Brown et al. (2020). *Language Models are Few-Shot Learners*. DOI: 10.48550/arXiv.2005.14165

D. M. Ziegler, N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. Christiano, and G. Irving (2020). *Fine-Tuning Language Models from Human Preferences*. DOI: 10.48550/arXiv.1909.08593

E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell (2021). "On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?" In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. FAccT '21. New York, NY, USA: Association for Computing Machinery. DOI: 10.1145/3442188.3445922

15:00 – 16:00 Few-shot prompting

After this high-level methodological introduction, we will get back to coding, diving into few-shot in-context learning. Like classic prompt engineering, this approach uses instructions instead of training or fine-tuning to generate classifications or annotations with an LLM. However, instead of providing only an instruction, one also adds a few well-selected examples to the model input. Combined with the task instruction, these examples should incentivize the LLM to adapt its responses to the user's desired output and response format.

Participants will learn how to integrate few-shot examples in their model input and API requests. Further, they will learn about the potential risks and problems of selecting and using such examples.

16:00 – 17:00 Using open-source models with `ollama`

So far, we have relied on OpenAI's chat models. They are proprietary and closed-source, which limits reproducibility and has other ethical and research-pragmatic issues.

Hence, in this session, we will learn how to download and use open-source LLMs like LlaMa 3 on your local computer using `ollama` and the `rollama` R package.

In the exercises, participants will reproduce their analysis based on OpenAI's GPT models on their local machines with the (4-bit quantized) LlaMa 3 model.

Readings and computer setup

- Computer setup:
  1. install `ollama`, see https://ollama.com/download
  2. install the `rollama` R package ($\geq$ 0.4.0), see https://jbgruber.github.io/rollama/
- Papers about the trade-offs between using open- and closed-sourced models:

!! A. Palmer, N. A. Smith, and A. Spirling (2024). "Using proprietary language models in academic research requires explicit justification". en. In: *Nature Computational Science* 4.1, pp. 2–3. DOI: 10.1038/s43588-023-00585-1

M. Weber and M. Reichardt (2023). *Evaluation is all you need. Prompting Generative Large Language Models for Annotation Tasks in the Social Sciences. A Primer using Open Models*. DOI: 10.48550/arXiv.2401.00284

!! the `rollama` "Annotation" vignette (see here)

17:00 – 17:30 Open questions and directions

In this closing session, we will discuss open questions that have come up throughout the course. This discussion will also point to gaps in existing methodological research on LLM text coding and annotation in the social sciences.