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

Version 2024-01-17

Instructor: Hauke Licht, Ph.D.
Email: hauke.licht@wiso.uni-koeln.de
Place: Paulstraße 3, D-50678 Colgone
Times: February 27, 2024, 9:30–17:00 and
March 26, 2024, 9:30–13:30

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 model for text coding and annotation. 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, in-context (few-shot) learning, and cost calculations. And acquiring the skills to automate text coding and annotation tasks using Python, the `openai` package, and OpenAi's GPT models. Practical experience with manual text coding or annotation is advantageous but not required.

Course Description

Large Language Models (LLMs) like OpenAi's ChatGPT or Google Research's BART are transforming our societies in a wide range of domains and applications. This also applies to the social sciences, where one of LLMs' many potentials lies in the new approaches to automated text analysis they enable. 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 (ranging from sentiment analysis to the detection of hate speech). These studies 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 – from conceptualization to coding scheme and instruction development to testing and validation.

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.

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 approaches for LLM-based text coding and annotation, including prompt engineering, in-context learning, and instruction tuning.
4. Participants can automate text classification and annotation tasks using Python, the `openai` Python package, the OpenAI API, and OpenAI's GPT 3.5-turbo or 4.0 models. This includes:
   (a) writing and optimizing LLM instructions ("prompts") for typical text annotation tasks
   (b) implementing text annotation tasks using OpenAI's GPT models
   (c) model- and use-case-specific cost calculations
   (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** (e.g., manual coding and automated text classification). The instructor can provide participants who do not already have this knowledge with references to introductory literature, which they can review through self-study in approximately 6 hours. A background in qualitative content analysis should make it easier for the relevant persons to access the materials.
3. **An activate OpenAI account** (sign-up at `https://platform.openai.com/signup`) and ideally an OpenAI *Plus* subscription (costs USD $20/month) so you can use GPT 4.0 in addition to GPT 3.5-turbo (login and go to `https://chat.openai.com/#pricing`).
4. **Basic Python programming skills**, including creating and manipulating lists and dictionaries, writing `for`-loops, importing and exporting tabular data such as CSV or Excel files. Due to time constraints, an introduction or recap of these basic programming skills cannot be provided during the workshop. The instructor can provide participants who do not already have these programming skills with preparatory materials, which they should be able to work through in approximately 6 hours. A background in R programming should make it easier to access these materials.

Requirements

1. **Readings:** None of the applied social science and computer science/NLP papers listed in the syllabus are mandatory readings. They are simply listed to inspire and inform.
2. **Hard- and software:** Participants must bring a laptop with a working Python $\geq$3.9 installation to the workshop. In addition, the instructor will provide a list of required Python packages two weeks before the first workshop day. Participants need to install these packages before the first Workshop day.
3. **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 February 27 and March `tbd`, 2024. We will meet at the *Max Planck Institute for the Study of Societies*, (Paulstraße 3, D-50678 Colgone) in room `tbd`. On the first day, we will meet from 9:30 am to 5:30 pm, including a 90-minute lunch break. On the second day, we will meet from 9:30 am to 1:00 pm.

Course Outline

Day 1

09:30 – 10:00 Introduction

Everyone will introduce themselves and give examples of their project ideas. The Instructor will briefly explain the course outline and the goals of the first workshop day.

10:00 – 12:30 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.

A significant focus of this session will be understanding different coding tasks and levels/units of coding. We will discuss document or text classification (including binary, multi-class classification, and multi-label classification), token or word-level classification for detecting and extracting phrases and multi-word expressions (as in entity recognition), and the pairwise comparison approach, which servers to scaling documents on a latent dimension based on pairwise codings.

In exercises, participants (i) define their target concept(s) and (ii) discuss their definitions in small groups. They then (iii) decide on the appropriate coding unit and a suitable coding task and (iv) develop a coding scheme and instructions.

Related political science methods paper

- 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

14:00 – 15:00 Large Language Models

Next, we will delve into the theory underpinning the development of Large Language Models (LLMs). We will begin by exploring the concept of LLM 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.

A key part of our discussion focuses on the differences between *masked* and *causal* language modeling, exemplified by contrasting BERT (Bidirectional Encoder Representations from Transformers) with GPT (Generative Pre-trained Transformer) models. This comparison provides insights into the distinct methodologies and capabilities of each model type.

We will then examine 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 talk about "scaling laws" that characterize how models' size and complexity impact their performance and capabilities and the "Reinforcement Learning from Human Feedback" (RLHF) framework responsible for the success of OpenAI's GPT 3.5 and 4.0 models.

Finally, we will redirect our attention to the workshop's main topic: Using LLMs for text annotation. We will begin with an overview of three key techniques for instructing LLMs to perform text coding and related tasks: prompt engineering, Few-shot in-context learning, and instruction tuning.

Related methods paper

W. X. Zhao et al. (2023). *A Survey of Large Language Models*. DOI: 10.48550/arXiv.2303.18223

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

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

J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei (2020). *Scaling Laws for Neural Language Models*. DOI: 10.48550/arXiv.2001.08361

J. Hoffmann et al. (2022). *Training Compute-Optimal Large Language Models*. DOI: 10.48550/arXiv.2203.15556

15:00 – 16:30 Prompt engineering

We will make our first practical steps in instructing a GPT model for text coding. We will focus first on the *prompt engineering* approach. Prompt engineering means writing an instruction that tasks an LLM to generate a response for some user inputs without prior task-specific training. In its application to text coding or annotation, the prompt engineering approach requires (i) to translate one's coding instructions into a prompt and (ii) to provide the LLM with texts that it should classify or annotate.

The exercises will thus focus on learning and implementing best practices for writing prompts for ChatGPT. Further, participants will practice using ChatGPT interactively to generate text-based measurements for their target concept.

Related methods paper

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

Applied social-science papers for inspiration

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

16:30 – 17:30 Automation

We will follow the prompt engineering exercises with guided exercises on automating prompt-based LLM text coding tasks with Python, the `openai` package, and the OpenAI API. We will begin by exploring OpenAI's API sandbox to understand the structure of its API in- and outputs. Next, we will cover how to set up Python and obtain an OpenAI API token to send requests and receive responses from participants' local computers.

Participants will learn how to use the OpenAI model API to send instructions to one of their GPT models and parse their responses. Further, we will cover how to instruct the system to return responses in a machine-readable format.

Day 2

09:30 – 10:00 Introduction

At the beginning of this second workshop day, participants will report their experiences with automating text coding and annotation tasks with OpenAI's GPT models.

10:00 – 11:00 In-context learning

We will then follow up on the sessions on prompt engineering and automation with a deep dive into few-shot in-context learning approaches. 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.

11:00 – 12:00 Evaluation and replicability

In this session, we will discuss evaluation and applicability. This session is designed to engage students in critical thinking and rigorous application of the concepts learned throughout the course. Exercises will focus on implementing the best practices we discuss.

We will begin by focusing on evaluating LLMs' text classifications and annotations. We will consider various perspectives and methodologies, including quantitative evaluation through prediction on a held-out test set and qualitative evaluation through a review of concrete coding decisions and the model's apparent "rationales." This will include a review of standard performance metrics like the F1 score and the AUC.

We will then address the topic of replicability and its limits in applications of closed-source LLMs like OpenAI's. Participants will learn how OpenAI's GPT models' generative behavior can be controlled through its hyperparameters.

12:00 – 13:00 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 that might spur interesting research collaborations.

The instructor will further provide participants with pointers to resources for using open-source LLMs in their research. While using open-source LLMs can requires more advanced computer setup steps and access to GPU computing resources, their use can alleviate some of the replicability concerns raised.