Sociologica. V.18 N.2 (2024)
ISSN 1971-8853
https://sociologica.unibo.it/

Best Practices for Text Annotation with Large Language Models

Petter Törnberg*

Institute for Language, Logic and Computation, University of Amsterdam (The Netherlands)

Submitted: May 1, 2024 – Revised version: July 28, 2024
Accepted: September 23, 2024 – Published: October 30, 2024

Abstract

Large Language Models (LLMs) have ushered in a new era of text annotation, as their ease-of-use, high accuracy, and relatively low costs have meant that their use has exploded in recent months. However, the rapid growth of the field has meant that LLM-based annotation has become something of an academic Wild West: the lack of established practices and standards has led to concerns about the quality and validity of research. Researchers have warned that the ostensible simplicity of LLMs can be misleading, as they are prone to bias, misunderstandings, and unreliable results. Recognizing the transformative potential of LLMs, this essay proposes a comprehensive set of standards and best practices for their reliable, reproducible, and ethical use. These guidelines span critical areas such as model selection, prompt engineering, structured prompting, prompt stability analysis, rigorous model validation, and the consideration of ethical and legal implications. The essay emphasizes the need for a structured, directed, and formalized approach to using LLMs, aiming to ensure the integrity and robustness of text annotation practices, and advocates for a nuanced and critical engagement with LLMs in social scientific research.

Keywords: Text labeling; classification; data annotation; large language models; text-as-data.

Acknowledgements

The supporting agency for this study is Dutch Research Council (NWO) VENI VI.Veni.201S.006

* p.tornberg@uva.nl

1 Introduction

The recent year has seen instruction-tuned Large Language Models (LLM) emerge as a powerful new method for text analysis. These models are capable of annotation based on instructions written in natural language — so called *prompts* — thus obviating the need to train models on large sets of manually classified training data (Wei et al., 2022). The models are highly versatile and can be applied to a wide array of text-as-data tasks, ranging from common procedures like sentiment analysis or topic modeling, to project-specific annotation challenges. Unlike previous methods, LLMs appear to draw not merely on syntactic properties of the text, but to leverage contextual knowledge and inferences to achieve high levels of performance across languages — even rivaling human experts in performance on some annotation tasks (Törnberg, 2024b). The ease-of-use, high accuracy, and relatively low costs of LLMs have meant that their use has exploded in recent months, appearing to represent a paradigm shift in text-as-data by enabling even researchers with limited knowledge in computational methods to engage in sophisticated large-scale analyses (Gilardi et al., 2023; Rathje et al., 2024; Törnberg, 2024b).

While LLMs bring important advantages over previous approaches to text-as-data and enable exciting new research directions, the rapid growth of the field is not without problems. LLM-based text annotation has become something of an academic Wild West, as the lack of established standards has meant that both researchers and reviewers lack benchmarks for evaluating LLM-based research, leading to risks of low-quality research and invalid results. LLMs fit poorly into our existing epistemic frameworks: many of the lessons from machine learning are obsolete, and while using LLMs at times appear eerily similar to working with human coders, such similarities can be equally misleading. While easy to use, the models are black boxes, and prone to bias, misunderstandings, and unreliable results — leading some researchers to warn against using the models for annotation altogether (Kristensen-McLachlan et al., 2023; Ollion et al., 2024). The models raise important questions about bias, calibration, and validation, and the field is thus in need of common standards for what constitutes acceptable and recommended research practices.

While critics are not inaccurate in describing LLMs as subjective, flawed, black-boxed, potentially biased, and prone to misunderstanding — these descriptions often apply similarly to human coders. In conventional coding procedures, such issues are managed by organizing coding in rigorous processes that identify disagreements, validate the reliability, and make transparent the management of subjectivity. Rather than neither using LLMs uncritically or rejecting them altogether, such an approach implies the possibility to instead structure, direct and formalize their use in ways that harnesses their capacities, while remaining conscious of their inherent weaknesses and risks.

As LLMs enter into our research processes, they will inevitably shape our epistemologies and findings: research tools are not merely passive instruments, but active participants in research procedures (Latour & Woolgar, 2013). By disrupting our established research procedures, LLMs bring to the surface challenging questions of meaning, nuance, and ambiguity that quantitative scholars too often seek to avoid. Such disruptions can be made productive, encouraging reflexivity and to consider the role of our methodologies in knowledge production. As scholars have argued, all research involves elements of interpretation, and interpretation is inherently subjective and contested (Byrne, 2002). The challenge is to acknowledge and manage this subjectivity through transparency and rigorous procedures.

This brief paper seeks to contribute to addressing the need for common standards by suggesting a set of best practices for how LLMs can be reliably, reproducibly, and ethically em-

ployed for text annotation. The paper targets both researchers seeking advice on how to use LLMs in a rigorous and reliable way, and reviewers seeking standards for evaluating research. The paper argues that, while LLMs can indeed be prone to display bias and unreliable results, we should not reject their use altogether — instead, we should manage their potential weaknesses by bringing them into a rigorous annotation process. The paper draws on previous published research published using LLMs, the authors own extensive work in the field, and discussions with scholars working in the field. The author's work using LLMs includes tracing the discursive shifts on migration over 40 years of Swedish parliamentary debates, measuring populism in political speech, and teaching a course in which students use LLMs to pursue their own innovative research projects. To illustrate the argument, we will throughout this essay draw on the example of a project in which LLMs are used to examine how affective polarization shapes the communication of political elites.

We will cover the following nine points: (1) choose an appropriate model, (2) follow a systematic coding procedure, (3) develop a prompt codebook (4) validate your model, (5) engineer your prompts, (6) specify your LLM parameters, (7) discuss ethical and legal implications, (8) examine model stochasticity, (9) consider that your data may be in the training data.

2 Choose an Appropriate Model

The choice of which LLM to use is one of the most central decisions in LLM-based text analysis (Yu et al., 2023). There are now a large and diverse set of models to choose from, ranging from small open-source local models that can be run on a phone to large platformed models accessible through a web interface or API — so called AIAAS (Artificial Intelligence As A Service). At the moment of writing, most studies using LLMs for text annotation have employed platform-based proprietary models, in particular OpenAI's models, and few offer explicit motivations for their model choice (e.g., Heseltine & Clemm Von Hohenberg, 2024; Tan et al., 2024). The popularity of platform-based models is likely due to their sophisticated capabilities, relatively low price, and ease-of-use — but such models also come with several important problems. First, proprietary models such as ChatGPT have been shown to change over time without notice, giving different results to the same instructions as a result of changes in the backend (Chen et al., 2023). While the API provides access to stable models, these tend to be deprecated after a relatively short time, making reproducibility nearly impossible. Second, as it is not known what data these models are trained on, the OpenAI models do not pass even a low bar of transparency (Liesenfeld et al., 2023). Third, using a model through an API can be problematic in terms of ethics and legal consideration for certain data, and the current advice is that OpenAI models should not be used with proprietary, secret, or confidential data (Ollion et al., 2024; Spirling, 2023).

While different models come with advantages and disadvantages, it is thus important to consider the implications of using a specific model. The choice of model should be explicitly argued for, and drawing on issues that are considered central to academic research, we can point to six general factors that should be considered when selecting which LLM for annotation:

1. Reproducibility: The results can be replicated by others using the same data and methodology, ensuring the results are consistent and reliable. To ensure reproducibility, use a fixed version of the LLM throughout the project, document the version, and ensure that the model will be available for future use.

2. Ethics and legality: The model should respect ethical and legal standards, including considerations of privacy, not storing research data, and compliance with relevant data privacy regulations.
3. Transparency: The methodologies, data sources, assumptions, and limitations of the model should be clearly documented and accessible for scrutiny.
4. Culture and language: The LLM should adequately support the language(s) and cultures of your textual data. Some models are more proficient in certain languages than others, which can influence the quality of the annotations — and even bias your findings if your corpus includes several languages. Specifically, many models are English and US centric, which can result in lower performance on other languages and cultures (Ollion et al., 2024).
5. Scalability: Ensure that the model can handle the size of your relevant data material in terms of costs and time. The speed of offline models depends largely on the available hardware, whereas for API-based models it depends on their rate limits and costs. (If you need to classify large amounts of data, it may be worth considering using a semi-supervised model trained on data annotated by the LLM. While this adds an additional step, such models tend to be faster and are possible to run on an average laptop, thus allowing processing large quantities of data).
6. Complexity: Ensure that the model has the capacity to handle the complexity of the task, for instance relating to advanced reasoning or parsing subtle latent meaning. Challenging analysis tasks and long prompt instructions may require larger and more sophisticated models, such as GPT4.0, that are capable of higher levels of reasoning and performance on benchmark tasks.

In general, best practice is to use an open-source model for which the training data is publicly known. It should be noted that not all downloadable models can be considered open-source models, as models vary significantly in terms of their openness of code, training data, model weights, licensing, and documentation — and it is therefore important to compare the models based on existing benchmarks for openness (Liesenfeld et al., 2023). The models also vary significantly in their capacity for text annotation. Some open source models have been found to yield results comparable to those of ChatGPT for certain tasks (Alizadeh et al., 2023; Weber & Reichardt, 2023). To compare and select an appropriate model, there are several benchmarks and leaderboards that provide an overview of the capacities of the quickly changing landscape of available models (Bommasani et al., 2023; Chia et al., 2024; HuggingFace, 2024).

Models that have been tuned to avoid controversial subjects — so-called "guardrails" (Fernandes et al., 2023; Ziegler et al., 2019) — can be problematic for certain annotation tasks, as the models may refuse to annotate particular issues that may be understood as controversial (Törnberg, 2024b). For instance, if the model is used to annotate messages with potentially controversial content (such as messages with radical political content) or the task itself can be seen as controversial (such as identifying the gender of an author), the models may provide low-quality responses, or refuse to respond altogether.

If possible, the model should be hosted on your own infrastructure instead of relying on cloud-based APIs. Hosting the model yourself gives you complete control over the model version and updates, as well as over how the model handles any sensitive or confidential information, and makes your work replicable. While self-hosting is not available for all models, it

can be surprisingly easy, cheap, and significantly faster than API-based models, depending on your available hardware and the annotation task at hand. Ideal practice also involves assessing whether your results can be reproduced using several models, thereby showing that the prompt and results are robust to details of implementation. Using LLMs for annotation through their web interface should in general be avoided, as these interfaces do not allow setting parameters, version control, and do not provide sufficient privacy or copyright provision — the data you provide is often kept and used for training models.

However, the best model ultimately depends on the task at hand, and it should be acknowledged that there are often trade-offs. It may, for instance, not be possible to use a smaller open-source model for complex tasks, and the researcher may thus be forced to use a model such as GPT-4. In choosing the model, it is useful to look at what instructions the model was tuned on, and how the model scores on benchmarks that are relevant for your domain of application (Chang et al., 2024). While it likely that we will soon see the development and standardization of academic-led open source academic LLMs specifically developed for data annotation, which will help resolve these tradeoffs (Spirling 2023), the bottom-line is thus that *the choice of model must always be motivated and argued for on the basis of explicit quality standards*.

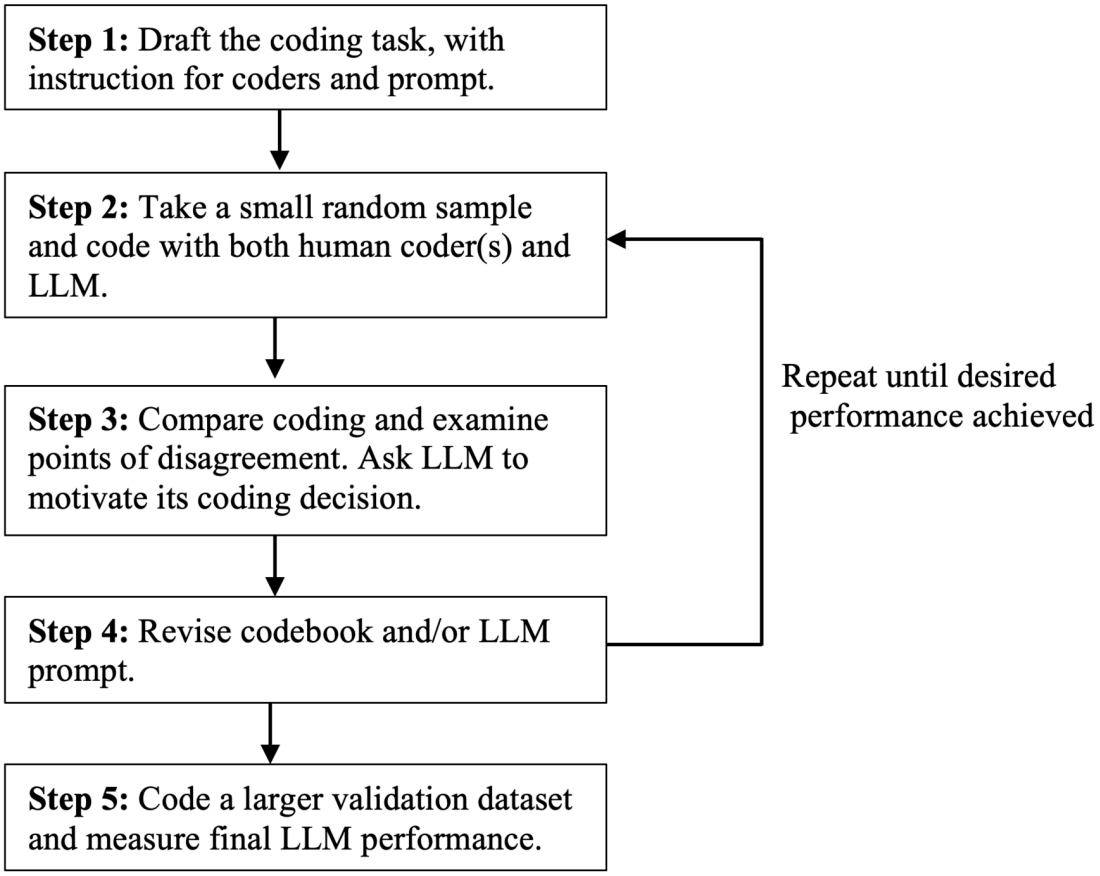

Figure 1: Example of a systematic coding procedure.

3 Follow a Systematic Coding Procedure

Text annotation is rarely merely a straight-forward technical task but tends to involve the challenging work of defining and operationalizing the meaning of social scientific concepts (Neuendorf, 2017). There are almost always boundary cases that become obvious only when engaging with the data — and some level of subjectivity is hence inevitable in coding. As scholars have long argued, it is more productive to openly acknowledge and face such issues, rather than to conceal them under a veneer of false objectivity. This recognition does not undermine the validity of the research; rather, it enriches the analysis by exposing the multifaceted layers of meaning that exist within the data, and enabling scholars to critically examine their own biases and assumptions.

Since LLMs can be fallible, unreliable, biased, and prone to misunderstand instructions (Ollion et al., 2024), it is important that the LLM is integrated into a systematic coding procedure that handles these issues and aligns their coding with the intended task. Such procedures are already well-established when it comes to organizing human coding efforts, and the LLM can successfully be brought into such a process.

An important difference between human coders and LLMs is that while LLMs code one text at the time, humans will tend to remember previous codings, and often learn and adapt over time. This in fact represents a common challenge when using human coders, as it means that definitions will tend to shift slightly over time, leading to inconsistencies in the data. At the same time, it means that researchers can draw important insights through qualitative engagement with the data involved in coding. While employing LLMs can supercharge coding procedures, it is important that it does not offset the advantages gained from in-depth engagement with the data.

Annotation work is generally organized as an iterative coding process (Yan et al., 2019; Glaser & Strauss, 2009): coders start with a set of texts, discuss discrepancies, refine the guidelines, and then proceed with the next set of texts. Such calibration sessions, where coders align their understanding and application of the guidelines, are crucial for maintaining consistency. When coding with an LLM, the development of the prompt is simply brought into this loop — simultaneously developing coding instructions and the LLM prompt. Once the LLM reaches sufficient agreement with the human coders, it can be used to code the full material.

Taking this approach, you can calculate the reliability both across the human coders and with the LLM. This allows assessing how well the LLM performs the task compared to human coders, and tracks the convergence between the coders and the LLM. Ideally, the LLM should approach the reliability achieved among the human coders.

1. Define the concept: It is important to come in with an explicitly articulated idea of the concept you are trying to capture, to avoid being overly influenced by the interpretations of the LLM. Write up a first description of the task at hand in the codebook, with instructions for both the human coder(s) and for the LLM. Make the prompt clear, unambiguous and specific, using direct and instructional language. (While the human instructions and the LLM prompt should generally be similar, it is usually beneficial to provide separate instructions.) For instance, when using LLMs to code populism, we drew on existing discursive definitions of populism to develop detailed instructions for how to identify populism in textual messages (Mudde & Kaltwasser, 2017).

2. Code a small training dataset: Have the human coders code a small representative dataset to enable testing your prompts, and use the LLM to annotate the same data.

3. Examine points of disagreement: Check the agreement between coders, and between the coders and the LLM. Discuss cases where coders disagree amongst each other, and on cases where the LLM disagreed with the coders. Ask the model to motivate its annotations for these cases and compare with the motivations of the human coder — as this can be a useful tool for sharpening your operationalization. At the same time, it is important to remain self-critical and reflexive: experience from several projects has shown that coders risk being overly swayed by the model's interpretations, as the models can provide highly convincing explanations. When comparing the coding of populism of the human coders and the LLM, we identified challenging boundary cases among the human coders that needed to be spelled out in the codebook. The comparison with the LLM identified several additional aspects that were taken-for-granted by the human coders due to shared cultural background, enabling a more objective and universal operationalization.
4. Refine the codebook and prompt: Make necessary adjustments to the instructions of either the human coders, of the prompt, or both. The human coders should not be considered ground truth: you may find that the LLM's interpretation was superior to the human coder. When used mindfully, the LLM can be a powerful tool for conceptual work.
5. Repeat: Return to step 2. Continue this process until the desired output quality is achieved.
6. Validate: Code the validation dataset, and measure the final performance of the LLM (see section 5).

Note that the process described above is merely an example and may need to be adapted to the specific needs of the project. If the zero-shot prompt is not giving adequate results, it can be useful to add few-shot examples. If the results are still inadequate, consider fine-tuning the model based on labeled training data.

4 Develop a Prompt Codebook

Best practice involves developing a *prompt codebook* with annotation guidelines for the human coders combined with detailed description of the prompts and LLM settings. The coding guidelines should, as always (Glaser & Strauss, 2009), be detailed instructions with clear definitions of the coding categories, examples of text corresponding to each category, and instructions on how to handle ambiguous cases (Neuendorf, 2017). A coder (human or LLM) that reads the codebook should have sufficient information to code a given text, with minimal disagreement between coders.

The codebook should simultaneously describe the corresponding prompts and parameters for the LLM, providing all details necessary to reproduce the LLM coding. This enables full reproducibility of both the manually coded validation data and the LLM coding. Note that the prompt should be considered tailored to the model used for its development: applying the same prompt to a different LLM may produce different results, even with models of similar parameter size (Sanh et al., 2022). If you finetune your model for your specific annotation task, the data used should be provided.

With the example of coding populism in political messages, the prompt codebook was designed as a standard codebook, with an appended section providing all information needed to reproduce the coding: the LLM prompt, the model used, and the relevant parameters.

Figure 2: Example of a well-structured prompt.

As an expert annotator with a focus on social media content analysis, your role involves scrutinizing Twitter messages related to the US 2020 election. Your expertise is crucial in identifying misinformation that can sway public opinion or distort public discourse.
Does the message contain misinformation regarding the US 2020 election?
Provide your response in JSON format, as follows:
`{ "contains_misinformation:" "Yes/No/Uncertain", "justification": "Provide a brief justification for your choice." }`
Options:
-Yes
-No
-Uncertain
Remember to prioritize accuracy and clarity in your analysis, using the provided context and your expertise to guide your evaluation. If you are uncertain about the classification, choose 'Uncertain' and provide a rationale for this uncertainty.
Twitter message: `[MESSAGE]`

Answer:

5 Engineer Your Prompts

One of the main implications of the use of LLMs for text annotation is the emergence of the task of *prompt engineering*: developing instructions that guide the LLM. While prompts are written in natural language and do not require technical skills per se, there can be huge differences in performance depending on details of how the prompt is written. Prompt engineering is hence becoming an important social scientific skill (White et al., 2024). Writing effective prompts can require significant effort, with multiple iterations of modification and testing (Jiang et al., 2020). While many prompting techniques have been developed, there is still limited theoretical understanding of why a particular technique is suited to a particular task (Zhao et al., 2021).

While previous advances in computational methods within the social sciences have tended to require sophisticated technical skills, prompt engineering requires other social scientific skills, such as theoretical knowledge, communication ability, and capacity for critical thinking. The process of developing prompts can furthermore be a useful way of deepening our understanding of social scientific concepts. Prompt engineering can in this sense therefore be thought of as a new type of — or even extension of — qualitative social science (Karjus, 2023). This paper will not provide a complete introduction to prompt engineering, as such guides are already readily available (e.g., OpenAI, 2024; Saravia, 2022), but will provide some important general advice.

- Structured prompts: An annotation prompt should contain the following elements: *context*, *question*, and *constraints*. The *context* gives a brief introduction to orient the model with any necessary background information. It can be split into role (e.g. expert annotator) and context (e.g. conspiracy theories). The *question* guides the response, defines the coding task. The *constraint* specifies the output format. Figure 2 offers an example of a well-structured prompt.
- Give instructions in the correct order: Recent and repeated text in the prompts has the most effect on LLM generation. It is therefore advisable to start with the *context*, followed by *instructions*, followed by the *constraints*.

- Enumerate options: If the answer is categorical, list the options in alphabetical order so that the output is simply the highest-probability token. Each option should be separated by a line-break.
- Give an "I don't know" option: Provide an option for the LLM to respond if it is uncertain about the correct answer. This reduces the risk of stochastic answers.
- Use lists: If the instruction is complex, make use of explicit lists to help the model pay attention to all elements in the prompt.
- Use JSON format: If the answer should contain several pieces of information, request a response in JSON format. The JSON format is easy to parse, and familiar to LLMs.
- Use an LLM for improving your prompt: LLMs have been shown to be effective at improving prompts. It can be particularly beneficial to follow an iterative process while utilizing an LLM to provide feedback and produce new versions of a seed prompt (Pryzant et al., 2023).
- Balance brevity and specificity: Well-written prompts involve a balance of specificity and brevity. While specificity in a prompt can lead to higher accuracy, performance can fall with longer prompts. Long prompts also make the process more costly, as you will need to feed the prompt for every annotation call.
- Chain-of-Thought: For certain tasks, it may be useful to employ more advanced techniques, such as the *Chain-of-Thought* (CoT) technique, to help elicit reasoning in LLMs (Wei, Wang, et al., 2024) and improve instruction-following capabilities (Chung et al., 2024). This involves breaking down the task into several simpler intermediate steps, allowing the LLM to mimic a step-by-step thought process of how humans solve complicated reasoning tasks. It can also be useful to trigger the model to engage in reasoning by using a prefix such as "Let's think step by step."
- System instructions: For most LLMs, the prompt instructions are provided as a "system" instruction, with the input as a "user" request.
- Few-shot prompting: It is often beneficial to also provide examples to guide the desired output, so called *few-shot prompting*, sent as a separate "user" and "assistant" dialogue.

6 Validate Your Model

LLM performance has been found to be highly contingent on both the dataset and the type of annotation task: while LLMs can even outperform expert human coders on some annotation tasks (Törnberg, 2024b), they can perform poorly on others (Kristensen-McLachlan et al., 2023). It is furthermore highly difficult to *a priori* assess how well an LLM will do on a specific task. Hence, it is always necessary to carefully validate the models on a task-by-task basis (Pangakis et al., 2023), both to offer evidence for the validity, and to reduce the ever-present risk for biases in the annotation. Validation is, in short, a basic requirement for publications using LLMs.

Validation usually consists of manually labeling a sufficient number of texts and ensuring that the labels correspond to a sufficient degree with the model results (Karjus, 2023). When

the LLM is used to provide data for a supervised model, the validation data can be used both to validate the results of the LLM, and of the supervised model.

There are several requirements for satisfactory validation:

- The validation must take place *after* the annotation prompt has been finalized: it is not acceptable to use the validation data to improve the prompts, as this may lead to falsely reporting higher precision.
- The validation dataset needs to be sufficiently large: The exact amount of validation data needed depends on several factors, such as the number of categories and the balance of categories. If the categories are imbalanced (that is, some categories have many more examples than others), you might need more data to ensure that the model performs well on the less-represented categories. The practical minimum is to have at least 20–30 samples of each category for a basic level of confidence in the performance metrics, but more is generally better. For high-stakes applications, you may need significantly more to ensure robustness. (For a precise determination, consider performing a power analysis.)
- Use appropriate performance metrics: Accuracy — i.e., correct answers divided by total answers — *is generally not a sufficient measure* to evaluate model performance, as it can be highly misleading, in particular for imbalanced datasets (if, for instance, one of your categories represents 90% of the population, then a model that classifies everything as belonging to that category will achieve a seemingly impressive accuracy of 90%.) What measure is appropriate however depends on the task at hand. For classification, measures such as *F1 Score* (usually together with *precision* and *recall*), *weighted-F1 score*, *ROC-AUC*, or *Cohen's Kappa* can be appropriate, whereas correlation-based measures, *MAE* or *MSE* can be more relevant when the model is annotating numeric values. In short, you need to argue for why your measure is the most appropriate choice, and it is in practice often beneficial to use a combination of these metrics to get a comprehensive understanding of different aspects of the model's performance.
- Consider comparing with human performance: Certain tasks are inherently more challenging than others. For instance, guessing the gender of an author based on short text is nearly impossible, and even the best possible model will hence have low accuracy. The acceptable performance level hence therefore on the task at hand. Calculating the performance of human coders, using e.g., an inter-coder reliability score, can provide a useful benchmark for evaluating the relative performance of a model.
- Consider any subsets of the data: If your dataset includes several subsets for which the model's capacity may vary, for instance different languages or cultural contexts, they need to be separately validated as the model may vary in its precision for each group.
- Examine and explain failures: The performance of LLMs can vary in unexpected ways — possibly involving bias or problematic misinterpretation of the concept. While LLMs can achieve high performance on many challenging problems, they can fail on seemingly simple tasks. Such failures can lead to errors in the downstream analysis, which are not visible in the performance metrics. Moreover, model bias may not be detectable through validation performance metrics. Say, for instance, that 10% of the data describes a particular minority, and that 30% of these are misclassified due to model bias. The resulting 3% failure rate would often be seen as acceptable. Researchers should therefore always

examine the failures in detail, and verify that they are not systematic and that they do not undermine the validity of downstream results.

While it is likely that we will soon see certain prompts and models become well-established for certain analysis tasks, the general advice is that any automated annotation process using LLMs *must* validate their LLM for their specific prompt, settings, and data. Rigorous validation is the most important step in using LLMs for text annotation.

7 Specify Your LLM Parameters

When using an LLM, there are several parameters that can affect the results produced by your prompts. Tweaking these settings are important to improve reliability and desirability of responses, and it may take some experimentation to figure out the appropriate settings for your use cases. The following list shows some common settings you may come across when using LLMs:

- Max Length: Sets the maximal number of tokens the model generates. Specifying a max length allows you to control costs, and prevent long or irrelevant responses.
- Temperature: The temperature parameter controls how random the model output is, essentially increasing the weights of all other possible tokens. Low temperature leads to more deterministic results, while high temperature leads to more randomness, that is, more diverse or creative outputs. For data annotation, a lower temperature is usually recommended, such as 0.
- Top-P: Adjusts the range of considered tokens. A low Top P ensures precise, confident responses, while a higher value promotes diversity by including less likely tokens. For data annotation, a lower Top-P is usually recommended, such as 0.2 to 0.4. If using Top-P, your temperature must be above 0.
- Top-K: The top-k parameter limits the model's predictions to the top-k most probable tokens. By setting a value for top-k, you can thereby limit the model to only considering the most likely tokens.

Your parameters *must* always be explicitly specified — even if they are the default parameters — as this is necessary for reproducibility.

8 Consider Ethical and Legal Implications

Using LLMs for text analysis opens several ethical considerations compared to traditional text analysis methods, in particular when using platformed LLMs. In regulatory contexts such as the EU, the use of AI furthermore also puts higher legal requirements on data management and ethics (Sartor & Lagioia, 2020). The following describes a list of ethical and legal considerations to be made when using LLMs for text annotation, drawing on GDPR and influential ethics frameworks (e.g., BSA, 2017; Franzke et al., 2020; Sharma, 2019).

1. Transparency and consent: Ensure that you have explicit consent from individuals whose personal data you are using that you will employ LLMs for its analysis. Users should be

informed about the use of third-party services and the implications for their data. More generally, when using a platformed LLM such as ChatGPT, Claude, or Gemini, your input data is likely to be used as training data.

2. Data Processing Agreement: When using third-party services like OpenAI, it may be necessary to have a Data Processing Agreement (DPA) in place (Sharma, 2019). This agreement should outline how the data is processed, the purposes of processing, and the measures taken to protect the data. For instance, if you are using ChatGPT and you are required to be GDPR compliant, you may need to execute a DPA with OpenAI (such an application form is available on the OpenAI website.)

3. Changing expectations of privacy: The research use of text data that users have published publicly — such as on platforms like X/Twitter or Telegram — is often motivated by users posting such data may have a reduced expectation of privacy. However, the data was likely published without the user considering the substantial capacity of LLMs to extract information, and researchers should thus carefully identify and respect users' expectations of privacy (Zimmer, 2020).

4. Data anonymization: Before sending data to a platformed LLM, ensure that all personal data is adequately anonymized or pseudonymized. This means removing or replacing any information that could directly or indirectly identify an individual. *Never* send proprietary, secret, or confidential data to an API or web interface without careful consideration of the ethical and legal implications.

5. Data minimization: You should only use and send the minimum amount of data necessary. While this is always an important ethical guideline, data minimization is also a legal principle, as it is part of EU's GDPR and California's CCPA (Sharma, 2019).

6. Data storage and transfer: Be mindful of where the data is stored and processed. The GDPR requires that data transfers outside the EU and the EEA are subject to adequate protections or are made to countries that provide an adequate level of data protection (Sharma, 2019).

7. Copyright and Terms of Service violations: If you are using copyrighted material, such as news articles from a proprietary database, you may need to receive explicit permission or license to analyze the data with an API-based LLM. Without explicit permission or a license from the copyright owner, sending the data to an API can be considered an infringement.

Ethical issues often involve difficult trade-offs. As usual, researchers should handle ethical considerations through an explicit and careful discussion and motivation in their research paper.

9 Examine Model Stochasticity

LLMs behavior in relation to prompts can be brittle and non-intuitive, with even minor details in the prompt — such as capitalization, interpunctuation, or the order of elements or words — significantly impacting accuracy, in some cases even going from state-of-the-art to near random chance (Kaddour et al., 2023; Zhao et al., 2021). Examining whether the model's results are

stable can be a useful shortcut to examining whether the model is able to carry out the coding reliably and with replicability, without the need for a validation procedure. Does the same prompt return the same result for a given text if run several times? Do small variations in the prompt result in different results? Large variations in output for minor changes in the prompt can indicate issues with the model's stability and reliability for a given task, making its text annotation less trustworthy. If the results are highly sensitive to minor prompt changes, it can also be challenging for other researchers to replicate the study and validate the findings.

To carry out such a prompt stability analysis, create several paraphrases of the prompt and run the analysis for a subset of the data. You can then estimate the stability by comparing the results, for instance using Krippendorf's Alpha reliability measure (Krippendorff, 2004). Barrie et al. (2024) have recently released a library to allow researchers to easily carry out such prompt stability scoring.

10 Consider That Your Data Might Be in the Training Data

When using conventional machine learning models, it is crucial to keep the data you test on separate from the training data to ensure that the model is robust, generalizable, and that it provides a realistic estimate of its performance on unseen data (Alpaydin, 2021; Grimmer et al., 2021). This may suggest that LLMs cannot be properly validated, as their training data is often so massive that it should be assumed that nearly any publicly available data will be included. However, the general rule does not necessarily apply to LLMs. As the purpose of validating text annotation is to assess the model's capacity for the specific task, it does not matter that the prompt validation data is in the training data, as long as the data on which the model will be run is also in the LLM training data. In fact, it is often desirable that the time-period covered is included in the training data, as it is necessary for the model to draw on contextual knowledge when making inferences about meaning (see Törnberg 2024b). For instance, if the task is to identify the ideology of a poster based on a social media message, it may be necessary to have knowledge of specific policy positions in a given political context.

However, there are situations where this may become problematic. For instance, if *parts* of the text data that you are annotating are in the LLM's training data and other parts are not, the two should preferably be validated separately, as the model's performance may differ. You therefore need to be mindful of the period for which the specific model was trained: if the end date of the LLM training data is within the period of your dataset, you may find that the quality of annotation varies over time — which can cause problems in your downstream analysis.

For the same reason, you should try to avoid using publicly available databases as validation data, as they may be in the model's training data. For instance, if you are interested in annotating party manifestos, existing manually labeled datasets (such as Manifesto Project Database) are not reliable means of validation: the LLM has likely already seen this database and may simply be reproducing the labels. This implies that the performance may not generalize to tasks for which the answer is not already publicly available. While the risks of such data contamination are often overstated, as the LLMs are trained on massive datasets and are trained as a next-word predictor and may thus be unlikely to have "memorized" the columns of a CSV file, the burden of evidence is on the validator.

11 Conclusion

This brief essay has collected an emerging set of best practices for text annotation using LLMs, to support both those using the methods as part of their research, and reviewers seeking to evaluate an academic contribution. As the field is undergoing rapid development, it should be noted that the standards and practices should be expected to continue evolving.

LLMs are revolutionizing text-as-data, enabling undergraduate students to carry out research in mere weeks that would previously have represented major research endeavors. At the same time, LLMs bring important challenges. As LLMs fit poorly into our existing epistemic frameworks for text annotation, they have caused a significant academic debate on their role in social scientific research. While many scholars have welcomed the methods — at times with a perhaps overly acritical acclaim — others have rejected them for being unreliable and incompatible with the principles of open science (Kristensen-McLachlan et al., 2023; Ollion et al., 2024). The suggestion at the core of this paper is that the methods are capable of sophisticated and rigorous interpretation — given appropriate use. LLMs can constitute a powerful contribution to social scientific research, but require new standards for evaluating their use and a new epistemic apparatus.

We can neither understand LLMs through the established epistemic framework of conventional supervised machine learning models, nor through the lens of human coders. In employing LLMs, we must be careful to remember that while LLMs can seem in some ways eerily human, they are not human in their capabilities. On some tasks — even those long seen as belonging to the distinctly human realm — they can be superhuman in their capacities (Törnberg 2024b). On other tasks, they perform worse than a small child. This means that we should not take for granted that their coding matches our intuitive expectations, and that we must always validate their performance, assess systematic biases, and develop detailed and transparent documentation of our procedures.

While best practices such as those presented in this essay are important to provide valuable guidelines and frameworks for research, it must be acknowledged that procedures and standards such as those described in this paper does come at the cost of making the use of LLMs more cumbersome and challenging, in particular for scholars with limited technical background. It is therefore crucial to apply them with discernment and flexibility, as an overly rigid adherence can hinder creativity and responsiveness. There is however rapid growth in availability of guides and tools to make it easy to use LLMs for annotation (e.g., Kim et al., 2024; Törnberg, 2024a). If designed to encourage best practices, such tools represent powerful ways of shaping rigorous research procedures (Latour & Woolgar, 2013; Rogers, 2013).

While critics are largely accurate in describing LLMs as subjective, black-boxed, potentially biased, and prone to misunderstanding — these descriptions often apply similarly to human coders. To manage these problems, conventional coding is organized in rigorous processes that identify disagreements and validate the reliability. Rather than neither using LLMs uncritically or rejecting them altogether, this implies the possibility to instead structure, direct and formalize their use in ways that harnesses their capacities, while remaining conscious of their inherent weaknesses and risks. The black-boxed nature and unreliability of LLMs can to large extent be managed through careful validation, to identify any errors that may affect downstream analyses.

As this essay has argued, the subjectivity of LLMs could moreover be understood as an inherent feature of interpretative work. Just as coding manages subjectivity by relying on intercoder reliability to ensure consistency among human coders, researchers should develop hybrid

systems where human oversight and AI capabilities complement each other. Interpretation is inherently contested, and the models bring to the surface challenging questions of meaning, nuance, and ambiguity that researchers too often seek to avoid. In the authors' projects, the use of LLMs has often allowed a sharpening of the concept and operationalizations, by the challenge from the novel perspective brought by the language model.

While this essay has focused on integrating LLMs into quantitative approaches to text-as-data, it should be noted that the method has similar implications for qualitative approaches. The epistemic challenge that LLMs represent for social scientific research can moreover productively challenge established conventions by encouraging the exploration of the hinterlands between qualitative and quantitative approaches, by, for instance, making possible large-scale interpretative research.

By making it easy to carry out sophisticated studies of meaning, LLMs empower a focus on aspects of the social world that have thus been underemphasized in computational research (Törnberg & Uitermark, 2021). Students and early career scholars now can perform analyses that were previously only available to the well-funded lab leader who could afford a team of coders. Such benefits are not to be taken lightly. As Kuhn (1962) famously argued, the most radical scientific advances stem not from accumulated facts and discoveries, but it is the invention of new tools and methodologies that trigger paradigm shifts in scientific work. The social sciences are currently in the midst of such a paradigm shift.

References

Alizadeh, M., Kubli, M., Samei, Z., Dehghani, S., Zahedivafa, M., Bermeo, J.D., Korobeynikova, M., & Gilardi, F. (2023). Open-Source Large Language Models for Text-Annotation: A Practical Guide for Model Setting and Fine-Tuning. *arXiv*, 2307.02179. https://doi.org/10.48550/arXiv.2307.02179

Alpaydin, E. (2021). *Machine Learning*. Cambridge, MA: MIT press.

Barrie, C., Palaiologou, E., & Törnberg, P. (2024). Prompt Stability Scoring for Text Annotation with Large Language Models. *arXiv*, 2407.02039. https://doi.org/10.48550/arXiv.2407.02039

Bommasani, R., Liang, P., & Lee, T. (2023). Holistic Evaluation of Language Models. *Annals of the New York Academy of Sciences*, *1525*(1), 140–146. https://doi.org/10.1111/nyas.15007

BSA. (2017). *Statement of Ethical Practice*. British Sociological Association. https://www.britsoc.co.uk/media/24310/bsa_statement_of_ethical_practice.pdf

Byrne, D.S. (2002). *Interpreting quantitative data*. London: Sage. https://doi.org/10.4135/9781849209311

Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., Ye, W., Zhang, Y., Chang, Y., Yu, P.S., Yang, Q., & Xie, X. (2023b). A Survey on Evaluation of Large Language Models. *ACM Transactions on Intelligent Systems and Technology*, *15*(3), 39, 1–45. https://doi.org/10.1145/3641289

Chen, L., Zaharia, M., & Zou, J. (2023). How Is ChatGPT's Behavior Changing over Time?. *arXiv*, 2307.09009. https://doi.org/10.48550/arXiv.2307.09009

Chia, Y.K., Hong, P., Bing, L., & Poria, S. (2024). InstructEval: Towards Holistic Evaluation of Instruction-Tuned Large Language Models. In A.V. Miceli-Barone, F. Barez, S. Cohen, E. Voita, U. Germann, 6 M. Lukasik (Eds.), *Proceedings of the First edition of the Workshop on the Scaling Behavior of Large Language Models* (pp. 35–64). Association for Computational Linguistics. https://doi.org/10.48550/arXiv.2306.04757

Chung, H.W., Hou, L., Longpre, S., Zoph, B., Tai, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., Webson, A., Gu, S.S., Dai, Z., Suzgun, M., Chen, X., Chowdhery, A., Castro-Ros, A., Pellat, M., Robinson, K., ... Wei, J. (2024). Scaling Instruction-Finetuned Language Models. *Journal of Machine Learning Research*, *25*(70), 1–53. http://jmlr.org/papers/v25/23-0870.html

Fernandes, P., Madaan, A., Liu, E., Farinhas, A., Martins, P.H., Bertsch, A., de Souza, J.G.C., Zhou, S., Wu, T., Neubig, G., & Martins, A.F.T. (2023). Bridging the Gap: A Survey on Integrating (Human) Feedback for Natural Language Generation. *Transactions of the Association for Computational Linguistics*, *11*, 1643–1668. https://doi.org/10.1162/tacl_a_00626

Franzke, A.S, Bechmann, A., Zimmer, M., & Ess, C. (2020). *Internet Research: Ethical Guidelines 3.0*. Association of Internet Researchers. https://aoir.org/reports/ethics3.pdf

Gilardi, F., Alizadeh, M., & Kubli, M. (2023). ChatGPT Outperforms Crowd Workers for Text-annotation Tasks. *Proceedings of the National Academy of Sciences of the United States of America*, *120*(30), e2305016120. https://doi.org/10.1073/pnas.2305016120

Glaser, B.G., & Strauss, A.L. (2009). *The Discovery of Grounded Theory: Strategies for Qualitative Research* (4th ed.). New Brunswick, NJ: Aldine Transaction. (Original work published 1999)

Grimmer, J., Roberts, M.E., & Stewart, B.M. (2021). Machine Learning for Social Science: An Agnostic Approach. *Annual Review of Political Science*, *24*(1), 395–419. https://doi.org/10.1146/annurev-polisci-053119-015921

Heseltine, M., & Clemm Von Hohenberg, B. (2024). Large Language Models as a Substitute for Human Experts in Annotating Political Text. *Research & Politics*, *11*(1), 20531680241236239. https://doi.org/10.1177/20531680241236239

HuggingFace. (2024). *Open LLM Leaderboard*. https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard

Jiang, Z., Xu, F.F., Araki, J., & Neubig, G. (2020). How Can We Know What Language Models Know? *Transactions of the Association for Computational Linguistics*, *8*, 423–438. https://doi.org/10.1162/tacl_a_00324

Kaddour, J., Harris, J., Mozes, M., Bradley, H., Raileanu, R., & McHardy, R. (2023). Challenges and Applications of Large Language Models. *arXiv*, 2307.10169. https://doi.org/10.48550/arXiv.2307.10169

Karjus, A. (2023). Machine-Assisted Mixed Methods: Augmenting Humanities and Social Sciences with Artificial Intelligence. *arXiv*, 2309.14379. https://doi.org/10.48550/arXiv.2309.14379

Kim, H., Mitra, K., Chen, R.L., Rahman, S., & Zhang, D. (2024). MEGAnno+: A Human-LLM Collaborative Annotation System. In N. Aletras, O. De Clercq (Eds.), *Proceedings of*

the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations (pp. 168–176). Association for Computational Linguistics. https://doi.org/10.48550/arXiv.2402.18050

Krippendorff, K. (2004). Reliability in Content Analysis: Some Common Misconceptions and Recommendations. *Human Communication Research*, *30*(3), 411–433. https://doi.org/10.1093/hcr/30.3.411

Kristensen-McLachlan, R.D., Canavan, M., Kardos, M., Jacobsen, M., & Aarøe, L. (2023). Chatbots Are Not Reliable Text Annotators. *arXiv*, 2311.05769. https://doi.org/10.48550/arXiv.2311.05769

Kuhn, T.S. (1962). *The Structure of Scientific Revolutions*. Chicago, IL: University of Chicago Press.

Latour, B., & Woolgar, S. (2013). *Laboratory Life: The Construction of Scientific Facts*. Princeton, NJ: Princeton University Press. https://doi.org/10.2307/j.ctt32bbxc

Liesenfeld, A., Lopez, A., & Dingemanse, M. (2023). Opening Up ChatGPT: Tracking Openness, Transparency, and Accountability in Instruction-Tuned Text Generators. In M. Lee, C. Munteanu, M. Porcheron, J. Trippas, S.T. Völkel (Eds.), *Proceedings of the 5th International Conference on Conversational User Interfaces* (pp. 1–6). New York, NY: ACM Press. https://doi.org/10.1145/3571884.3604316

Mudde, C., & Kaltwasser, C.R. (2017). *Populism: A Very Short Introduction*. Oxford: Oxford University Press. https://doi.org/10.1093/oxfordhb/9780198803560.013.1

Neuendorf, K.A. (2017). *The Content Analysis Guidebook*. London: Sage. https://doi.org/10.4135/9781071802878

Ollion, É., Shen, R., Macanovic, A., & Chatelain, A. (2024). The Dangers of Using Proprietary LLMs for Research. *Nature Machine Intelligence*, *6*, 4–5. https://doi.org/10.1038/s42256-023-00783-6

OpenAI. (2024). *Prompt Engineering*. https://platform.openai.com/docs/guides/prompt-engineering/strategy-write-clear-instructions

Pangakis, N., Wolken, S., & Fasching, N. (2023). Automated Annotation with Generative AI Requires Validation. *arXiv*, 2306.00176. https://doi.org/10.48550/arXiv.2306.00176

Pryzant, R., Iter, D., Li, J., Lee, Y.T., Zhu, C., & Zeng, M. (2023). Automatic Prompt Optimization with "Gradient Descent" and Beam Search. In H. Bouamor, J. Pino, & K. Bali (Eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing* (pp. 7957–7968). Association for Computational Linguistics. https://doi.org/10.48550/arXiv.2305.03495

Rathje, S., Mirea, D.M., Sucholutsky, I., Marjieh, R., Robertson, C.E., & Van Bavel, J.J. (2024). GPT Is an Effective Tool for Multilingual Psychological Text Analysis. *Proceedings of the National Academy of Sciences of the United States of America*, *121*(34), e2308950121. https://doi.org/10.1073/pnas.2308950121

Rogers, R. (2013). *Digital Methods*. Cambridge, MA: The MIT Press.

Sanh, V., Webson, A., Raffel, C., Bach, S., Sutawika, L., Zaid, A., Antoine, C., Arnaud, S., Arun, R., & Manan, D. (2022). Multitask Prompted Training Enables Zero-Shot Task Generalization. *arXiv*, 2110.08207. https://doi.org/10.48550/arXiv.2110.08207

Saravia, E. (2022). *Prompt Engineering Guide*. GitHub. https://github.com/dair-ai/Prompt-Engineering-Guide>

Sartor, G., & Lagioia, F. (2020). *The Impact of the General Data Protection Regulation (GDPR) on Artificial Intelligence*. Study. Panel for the Future of Science and Technology. EPRS, European Parliamentary Research Service. https://doi.org/10.2861/293

Sharma, S. (2019). *Data Privacy and GDPR Handbook*. Hoboken, NJ: Wiley. https://doi.org/10.1002/9781119594307

Spirling, A. (2023). Why Open-source Generative AI Models Are an Ethical Way Forward for Science. *Nature*, *616*, 413. https://doi.org/10.1038/d41586-023-01295-4

Tan, Z., Li, D., Wang, S., Beigi, A., Jiang, B., Bhattacharjee, A., Karami, M., Li, J., Cheng, L., & Liu, H. (2024). Large Language Models for Data Annotation: A Survey. *arXiv*, 2402.13446. https://doi.org/10.48550/arXiv.2402.13446

Törnberg, P. (2024a). *How to Use Large-Language Models for Text Analysis*. London: Sage. https://doi.org/10.4135/9781529683707

Törnberg, P. (2024b). Large Language Models Outperform Expert Coders and Supervised Classifiers at Annotating Political Social Media Messages. *Social Science Computer Review*, https://doi.org/10.1177/08944393241286471

Törnberg, P., & Uitermark, J. (2021). For a Heterodox Computational Social Science. *Big Data & Society*, *8*(2). https://doi.org/10.1177/20539517211047725

Weber, M., & Reichardt, M. (2023). Evaluation Is All You Need. Prompting Generative Large Language Models for Annotation Tasks in the Social Sciences. A Primer Using Open Models. *arXiv*, 2401.00284. https://doi.org/10.48550/arXiv.2401.00284

Wei, J., Bosma, M., Zhao, V.Y., Guu, K., Yu, A.W., Lester, B., Du, N., Dai, A.M., & Le, Q.V. (2022). Finetuned Language Models Are Zero-Shot Learners. *arXiv*, 2109.01652. https://doi.org/10.48550/arXiv.2109.01652

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., & Zhou, D. (2024). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, A. Oh (Eds.), *NIPS'22: Proceedings of the 36th International Conference on Neural Information Processing Systems* (pp. 24824–24837). New Orleans, LA: Curran.

White, J., Hays, S., Fu, Q., Spencer-Smith, J., Schmidt, D.C. (2024). ChatGPT Prompt Patterns for Improving Code Quality, Refactoring, Requirements Elicitation, and Software Design. In A. Nguyen-Duc, P. Abrahamsson, F. Khomh (Eds.), *Generative AI for Effective Software Development*. Cham: Springer. https://doi.org/10.1007/978-3-031-55642-5_4

Yan, C.T., Birks, M., & Francis, K. (2019). Grounded Theory Research: A Design Framework for Novice Researchers. *Sage Open Medicine*, *7*, 205031211882292. https://doi.org/10.1177/2050312118822927

Yu, H., Yang, Z., Pelrine, K., Godbout, J.F., & Rabbany, R. (2023). Open, Closed, or Small Language Models for Text Classification?. *arXiv*, 2308.10092. https://doi.org/10.48550/arXiv.2308.10092

Zhao, Z., Wallace, E., Feng, S., Klein, D., & Singh, S. (2021). Calibrate Before Use: Improving Few-Shot Performance of Language Models. In M. Meila & T. Zhang (Eds.), *Proceedings of the International Conference on Machine Learning* (pp. 12697–12706). https://doi.org/10.48550/arXiv.2102.09690

Ziegler, D.M., Stiennon, N., Wu, J., Brown, T.B., Radford, A., Amodei, D., Christiano, P., & Irving, G. (2019). Fine-Tuning Language Models from Human Preferences. *arXiv*, 1909.08593. https://doi.org/10.48550/arXiv.1909.08593

Zimmer, M. (2020). "But the Data Is Already Public": On the Ethics of Research in Facebook. In K.W. Miller & M. Taddeo (Eds.), *The Ethics of Information Technologies* (pp. 229–241). London: Routledge. https://doi.org/10.4324/9781003075011-17

Petter Törnberg – Institute for Language, Logic and Computation, University of Amsterdam (The Netherlands)
https://orcid.org/0000-0001-8722-8646 | p.tornberg@uva.nl
https://www.pettertornberg.com
Petter Törnberg is an Assistant Professor in Computational Social Science at the University of Amsterdam (The Netherlands). He studies the intersection of AI, social media, and politics, and draws on computational methods and digital data for critical inquiry. His recent books include *Intimate Communities of Hate: Why Social Media Fuels Far-Right Extremism* (with Anton Törnberg, Routledge, 2024), and *Seeing Like a Platform: An Inquiry into the Condition of Digital Modernity* (with Justus Uitermark; Routledge, forthcoming in early 2025).