Special Issue: Integrating Generative AI into Social Science Research

Sociological Methods & Research
1–40

DOI: 10.1177/00491241251339188
journals.sagepub.com/home/smr


Updating "The Future of Coding": Qualitative Coding with Generative Large Language Models

Nga Than[1], Leanne Fan[1], Tina Law[2], Laura K. Nelson[3], and Leslie McCall[1]

Abstract

Over the past decade, social scientists have adapted computational methods for qualitative text analysis, with the hope that they can match the accuracy and reliability of hand coding. The emergence of GPT and open-source generative large language models (LLMs) has transformed this process by shifting from programming to engaging with models using natural language, potentially mimicking the in-depth, inductive, and/or iterative process of qualitative analysis. We test the ability of generative LLMs to replicate and augment traditional qualitative coding, experimenting with multiple prompt structures across four closed- and open-source generative LLMs and proposing a workflow for conducting qualitative coding with generative LLMs. We find that LLMs can perform nearly as well as prior supervised machine learning models in accurately matching hand-coding output. Moreover, using generative LLMs as a natural language interlocutor closely replicates traditional

[1]Stone Center on Socio-Economic Inequality, The Graduate Center, CUNY, New York City, USA
[2]Department of Sociology, University of California-Davis, Davis, USA
[3]Department of Sociology, University of British Columbia, Vancouver, Canada

Corresponding Author:
Laura K. Nelson, Department of Sociology, University of British Columbia, 6303 NW Marine Dr., Vancouver, BC V6T 1Z1, Canada.
Email: laura.k.nelson@ubc.ca

Data Availability Statement included at the end of the article

qualitative methods, indicating their potential to transform the qualitative research process, despite ongoing challenges.

Keywords

generative large language models, qualitative coding, epistemology, qualitative methods, computational methods, computational text analysis

Introduction

The exploration of automated methods for discovering, refining, and annotating concepts and other social artifacts in textual data has been a major topic of research in the social sciences over the past decade. Computational social scientists have focused on refining new techniques from computer and information sciences to enable greater accuracy in identifying the often subtle and complex subject matter present in texts of interest to social scientists (e.g., Grimmer, Roberts, and Stewart 2022). The goal for many is twofold: to reach a level of accuracy and reliability commensurate with hand coding, facilitating the incorporation of larger volumes of data, and to enable the discovery of new patterns and topics not readily discoverable via qualitative reading alone (Foster and Evans 2024). The introduction of each new application of a different computational technique generally requires advanced quantitative and programming skills (Kesari et al. 2023). That cycle was arguably broken with the emergence in 2022 of a comparatively user-friendly product, ChatGPT by OpenAI.

The decades-long shift from statistical natural language processing to the deep learning paradigm, represented in part by the GPT models (the models underlying ChatGPT), has fundamentally transformed computational text analysis in (at least) three respects. First, there has been a shift from methods that involve the manual engineering of text into numerical representations, such as document-term matrices, to deep learning techniques that learn, via neural network modeling, distributed representations of words and their context (Bengio et al. 2003; Manning and Schutze 1999). Second, transfer learning and large pre-trained models have led to greater accuracy for many tasks while greatly reducing data and compute requirements (Do, Ollion, and Shen 2022). Third, and more recently, there has been a shift from an interface that uses programming languages exclusively to an interface that also uses natural language inputs and outputs (Ibrahim

and Voyer 2024). As we argue here, the deep learning and transfer learning paradigms, when combined with natural language interfaces, have the potential to transform the qualitative coding process, when validated and used with care.

Indeed, the growing interest in testing the ability for generative Large Language Models (LLMs) like GPT to augment or substantially replace hand coding is a testament to their projected transformative power (see, e.g., Bail 2024; Bommasani et al. 2021; Davidson 2024; Do, Ollion, and Shen 2022; Rytting et al. 2023). However, the results on accuracy have thus far been mixed, and, in part because of the natural language interface, there are currently no widely agreed-upon guidelines on how best to use LLMs for qualitative coding practices that involve labeling, classification, annotation, or discovery tasks (but see Chae and Davidson 2025; Gilardi, Alizadeh, and Kubli 2023; Ollion et al. 2024; Reiss 2023). Moreover, much of the recent work examining the capabilities of LLMs has sought to compare the accuracy of LLM classifications to, for example, crowd workers or expert coders across a range of tasks (e.g., Alizadeh et al. 2024; Chae and Davidson 2025; Do, Ollion, and Shen 2022; Rytting et al. 2023; Ziems et al. 2024). Yet *generative* LLMs do more than produce text classifications; they take natural language as input and output natural language. This processual shift, we argue, could mark a turning point for the qualitative text analysis process in the social sciences (see also Ibrahim and Voyer 2024).

In this article, we examine whether instruction-tuned generative LLMs—LLMs fine-tuned to better understand and follow user instructions and then used to generate text—can replicate and reliably augment the traditional iterative qualitative process of alternating between deductive and inductive analysis when annotating complex social science textual data (Deterding and Waters 2021; Do, Ollion, and Shen 2022; Ibrahim and Voyer 2024; Nelson 2020). Our test case is derived from an original, in-depth, fully qualitative project in which graduate and undergraduate students hand coded a corpus of over 1,200 articles appearing in U.S.-based newsweeklies from 1980 to 2012. We refer to this process as text classification, in accordance with the terminology used in machine learning, but our data and classification tasks are distinct in two interconnected ways. First, the documents are longer than the data LLMs have been tested on thus far for classification and related content-analysis tasks in the social sciences.[1] Second, the classification task requires a holistic and contextual reading of entire documents; that is, the concept to be identified cannot usually be gleaned from a single sentence or paragraph. In addition to enabling us to benchmark our results against those of human coders, this dataset allows us to propose a more

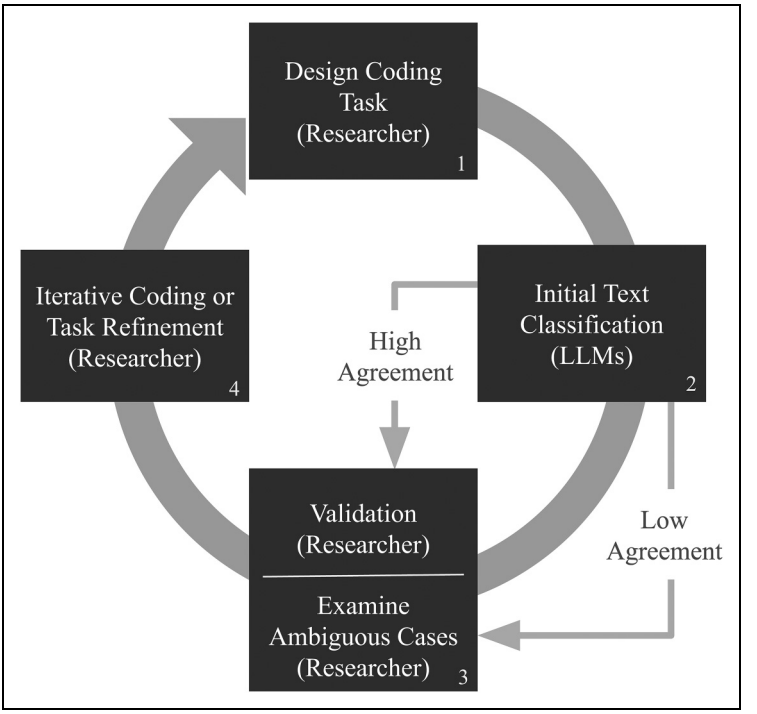

Figure 1. Researcher-LLM-Researcher workflow.

general workflow that converts qualitative coding processes into a new approach that emphasizes strategic interaction with machines while retaining key qualitative coding tools, such as a detailed coding guide and deep human reading at key points throughout the coding process (see Figure 1, discussed further below).

Concretely, we examine whether instructing LLMs to interpret and categorize text based on holistic qualitative criteria detailed in extensive coding guides can generate not only accurate outputs but also insights into the coding process itself comparable to those gained from the iterative process of training a group of research assistants to apply codes by hand. For instance, rather than hand coding a large number of documents for a supervised machine learning task, research assistants could instead participate in the development of instructions for LLMs and conduct more limited but essential hand coding for the purposes of concept refinement and output

validation (Bonikowski, Luo, and Stuhler 2022; Do, Ollion, and Shen 2022; Ibrahim and Voyer 2024; also, see again Figure 1). Alternatively, scholars without research assistants, including graduate students themselves, can engage in this same process on their own, including scholars who are embarking on a new text analysis project, or those wishing to modify an existing qualitative coding project or to extend it to a new and larger corpus. Given the format of our data, the processes we explore here are likely best suited for the content analysis of long-form documentary evidence (e.g., legacy news, organizational documents, oral histories), which constitutes a central line of investigation in the social sciences (Ferree et al. 2002; Gamson 1992; Gilens 1999; Law 2022; Nelson 2021; Nelson and King 2020).

With this overarching goal in mind, we contribute to the emerging literature on generative LLMs in two more specific ways. First, drawing from the existing literature, we experimented with multiple methods of interacting with, or "prompting," LLMs. We explore how to convert a coding guide into a prompt or sequence of prompts, and, in particular, whether and how to derive a definition of a multidimensional concept from the coding guide to use as part of a natural language prompt (or sequence of prompts). For instance, we deductively derived definitions of the focal concept from the hand-coding guide ourselves (i.e., manually) and then in a separate test deployed LLMs to generate these definitions from the guide, thus automating parts of the prompting process itself (Khattab et al. 2023; Wu et al. 2023; Yang et al. 2023). Like others, we also tested whether to provide examples of pre-labeled documents within the prompt (i.e., zero- and few-shot prompts, see Chae and Davidson 2025; Rytting et al. 2023; Ziems et al. 2024), though our work highlights the challenges of these formats when analyzing long documents specifically. Given the almost infinite variation in prompts that is possible with generative LLMs, due in large part to the vagaries of the natural language interface itself, our tests across a wide range of prompting strategies offer initial guidance for some of the more challenging tasks in automated textual analysis in the social sciences.

Second, across all the main tests in which we systematically vary the prompting strategy, we compare the performance of both closed-source and open-source LLMs. We focused in particular on open-source LLMs that are readily available on the platform Ollama, which allows others to immediately replicate our process without the need for specialized hardware or accounts with proprietary platforms.[2] There is also greater transparency about the training data for open-source models, enabling users to determine whether their own data are part of the training set, which could lead to

increased accuracy. We compare the results from (pre-quantized) open-source LLMs to GPT4, as GPT4 continues to be the industry standard, though, like others, we maintain, and illustrate why, it should not generally be used for social science research because of its hidden, proprietary architecture (Ollion et al. 2024; Spirling 2023).[3]

In the remainder of the paper, we begin in Section "Reconciling Qualitative and Computational Approaches to Textual Analysis" by further describing our overarching goal, which is to use the traditional qualitative coding process of developing a detailed codebook and training research assistants as a template for instructing and interacting with LLMs. In the following section, we turn to a discussion of LLMs themselves and the potentially important distinction between closed- and open-source models, which we test in the empirical section of the paper. We then describe our data and specific classification task, and how it relates to other kinds of textual analysis in the social sciences, in Section "Data and Classification Task," followed in Section "Prompting Strategies" by a description of how we convert that classification task into prompting strategies. Sections "Results" and "Discussion" provide the results and concluding discussion. Although experimentation with LLMs will be ongoing for some time to come, in part because the terrain of LLMs is itself changing rapidly, our tests suggest that LLMs have the potential to allow researchers to deploy (invaluable yet resource-intensive) qualitative hand coding more selectively and strategically in their projects (Madsen, Munk, and Soltoft 2023).

Reconciling Qualitative and Computational Approaches to Textual Analysis

Traditional qualitative coding in the social sciences typically involves researchers iteratively—through both deductive and inductive approaches—developing comprehensive codebooks to define and measure complex themes in the text, followed by extensive training of research assistants to apply these guidelines accurately while refining the codebook and measures during this process (Chong and Druckman 2011; Deterding and Waters 2021; Ferree et al. 2002; Griswold 1987; Nelson 2020). This process ensures that elaborate and sometimes fuzzy themes are defined clearly enough that multiple research assistants independently code them in the same way, and it preserves nuanced human understanding across analyses. In some cases, including the case we explore here, the coding of themes cannot happen at the word, phrase, sentence, or even paragraph level, but requires a holistic and contextualized

reading of an entire document (as illustrated further below). For all these reasons, a humanistic approach will certainly continue to play an essential role in the analysis of social texts even when automated methods are introduced to "augment" the qualitative coding process (Grimmer, Roberts, and Stewart 2022).

Nonetheless, the process of qualitative coding presents several challenges, all well recognized by qualitative researchers. Hand coding is time consuming and thus inflexible: once a significant amount of text has been hand coded, it is difficult to change coding schemes if desired, although this rigidity is relaxed to a certain extent by qualitative data analysis software (Deterding and Waters 2021). Reliability across (and even within) coders is often difficult to achieve, necessitating complicated intercoder reliability tests and metrics (e.g., Krippendorff 2012). And, as datasets grow in size and complexity, qualitative researchers often do not feel hand coding is capable of discovering and capturing the full nuance and range of themes contained within. Indeed, quantitative researchers bemoan the inadequacy of conventional deductive quantitative methods for the same reason (Brandt and Timmermans 2021; Do, Ollion, and Shen 2022; Grimmer, Roberts, and Stewart 2022; Wagner-Pacific, Mohr, and Breiger 2015).

Still, many qualitative researchers have already invested an enormous amount of time in the development of lengthy hand-coding guides, and still others will need to begin new qualitative coding projects with the development of a guide. This is because such researchers are often interested in subject matter—what we will refer to as concept identification—that cannot be easily traced in an accounting of specific words deductively produced by an expert in the field of analysis (Bonikowski, Luo, and Stuhler 2022; Do, Ollion, and Shen 2022; Knight 2022; Nelson et al. 2018; Stoltz and Taylor 2019; Voyer et al. 2022). This may be true for many reasons. Relevant words may be either polysemic or nearly absent in texts where alternative—more accessible, euphemistic, subtle, implicit, or intentionally obscured—ways of referencing the concept are used. Specific words may also be insufficient on their own to describe the multifaceted or multidimensional nature of many social science concepts. Under these circumstances, even computational researchers advocate for the use of highly trained human coders for the labeling of a subset of data for validation purposes. In comparison, training coders is relatively straightforward and less time-intensive for tasks involving entity recognition and stance detection in short pieces of text (Do, Ollion, and Shen 2022; Rytting et al. 2023).

A growing body of research in the computational analysis of textual data recognizes the challenges of using computational methods to code complex

concepts (Bonikowski, Luo and Stuhler 2022; Do, Ollion, and Shen 2022; Jensen et al. 2022; Nelson et al. 2018; Voyer et al. 2022), but a gap in the literature remains in how to translate the process of defining concepts in a coding guide into the process of instructing and interacting with generative LLMs. As noted above, potential similarities in these processes are enabled by the fact that the generative LLM interface takes natural language as input and produces natural language as output. Moreover, LLMs model recurring patterns in extended text sequences, leveraging (typically) dense representations of words to encode both common syntactic structures and subtle semantic variations, such as diverse styles, tones, and topics of discussion. We thus investigate the practical use of generative LLMs for iterative qualitative coding, testing the accuracy of LLM output as well as their ability to replicate the processes traditionally done by human researchers while maintaining a crucial role for human interpretation and intervention (Grimmer, Roberts, and Stewart 2022).

Variation among Generative LLMs

Thus far, we have stressed the fact that instruction-tuned generative LLMs produce human-like output by mimicking the complexities of natural language. Yet, scholars have strongly cautioned social scientists against some of the AI industry's "hype" (Ollion et al. 2024). Closed-source or proprietary LLMs, such as ChatGPT, have garnered particularly harsh critique for a wide range of reasons (Ollion et al. 2024; Spirling 2023). The lack of transparency in training data and algorithms prevents researchers from being able to fully account for what the models encode or what they can accurately do. Closed-source models also restrict customization and adaptability, which limits researchers' ability to tailor them to specific research questions or contexts. The reliance on closed-source models additionally raises issues of reproducibility, as researchers cannot verify results without access to the same (often expensive) tools and data, and even then, the same models can produce different results when the underlying model structure is updated without notice. These challenges pose ultimately insurmountable barriers to reproducible and ethical social science research (Ollion et al. 2024; Spirling 2023).

In response, corporations and the scientific community are embracing open-source models for their greater transparency and collaborative potential. The release of models such as OpenAI's GPT2 laid the groundwork for subsequent open-source initiatives. Organizations like Hugging Face, The Allen Institute for AI, and EleutherAI and platforms such as Ollama have made

significant strides in democratizing access to powerful language models, enabling researchers to inspect, modify, and build upon existing architectures using local machines without specialized hardware. And Meta and Google have released their own open-source models—Llama (Dubey et al. 2024) and Gemma (Mesnard et al. 2024), respectively—that are beginning to perform on par with closed-source alternatives for many tasks (as we show below, this is the case with our experiments). These open-source models provide researchers with the flexibility to tailor models to specific needs, ensuring greater accuracy in and relevance to their work. Additionally, open-source initiatives promote reproducibility and peer validation, as researchers can freely access and test models. This collaborative environment accelerates innovation but also invites the kind of broader scrutiny needed to mitigate well-documented biases in language models (Bender et al. 2021; Ollion et al. 2024; Spirling 2023).

In short, with the boom in the development of open-source LLMs and their intermediary platforms (e.g., Ollama), the social science community is in a position to rigorously consider how LLMs of various kinds—and the interactive, generative use of them—might change the landscape of computational, qualitative text analysis. Here we focus in particular on the implications of LLMs for coding subject matter in the social sciences that is challenging even for humans to identify systematically.

Data and Classification Task

Our current work extends an original, fully qualitative project conducted by one of our co-authors (McCall 2013) and a replication study conducted by two of our co-authors (Nelson et al. 2018). In the original project, more than a half dozen undergraduate and graduate research assistants coded a sample of 1,253 newsweekly articles appearing in *Time*, *Newsweek*, and *U.S. News & World Report* between 1980 and 2012 (McCall 2013). The original task entailed the identification of a broad, multifaceted concept used widely in the social sciences—socio-economic inequality—and its manifestation through a new empirical reality unfolding over time—the rise in wage, earnings, income, and wealth inequality beginning in the late 1970s. This coding task is like many others in politics and political sociology where scholars qualitatively analyze media and other discursive data to better understand whether and how a new issue becomes known and politicized in the public sphere (e.g., Gilens 1999, on welfare; Ferree et al. 2002, on abortion). Additionally, the multifaceted nature of the concepts to be identified, which can straddle multiple paragraphs of a long document, are characteristics

shared by other kinds of long-form qualitative data, such as legacy news, interview/audio transcripts, oral histories, and organizational documents (Deterding and Waters 2021; Litterer et al. 2024; Nelson 2020).

The specific data and classification task in this project has three other characteristics that could be of more general interest to qualitative researchers: (1) the process of identifying articles that are relevant to a concept is time-intensive and unfolds over multiple stages of a long-term research project; (2) keyword searches are insufficient for selecting relevant articles; and (3) relevant articles discuss a concept either implicitly or explicitly.

First, the challenges encountered in "simply" identifying, or classifying, articles in which the target subject matter appeared (about a quarter of the sample in our data) meant that we were unable in the original analysis to hand code other aspects of the articles, such as content related to the causes and consequences of rising economic inequality or to intersections with race, ethnic, and gender inequality. Yet these aspects of an article can critically impact how readers might potentially understand and respond to the issue of socio-economic inequality. In subsequent work, then, we can experiment with automated methods not only to corroborate the original hand-coding results but to extend the substantive research and coding process into new, adjacent territory, and into new corpora as well. In this respect, our extension of this work using LLMs is an example of the "agnostic" approach to textual analysis advocated by Grimmer, Roberts and Stewarts (2022: 26–28), in which "building, refining, and testing social science theories requires iteration and cumulation" across multiple methods and targets of analysis.

Second, and related, the data and classification task adopted in this article provides an illustration of the steps needed to build a corpus appropriate for deeper content analysis (Pardo-Guerra and Pahwa 2022). These steps are necessary for researchers who do not have access to a pre-defined corpus (i.e., a corpus where all documents are by definition relevant, such as an oral history archive or an organization's newsletters). For instance, in the first step of the original study, a comprehensive keyword-based search on topics potentially related to the rise in economic inequality in the United States was conducted. A random 10–15 percent sample of articles obtained from this search was then selected in each year from 1980 to 2012 to reduce the scale of hand coding. However, in the final step involving hand coders, only about a quarter of the sampled articles ended up in the crucial sub-corpus of articles hand coded as mentioning economic inequality. Because the original research sought to determine *whether* and *when* the media was covering inequality, this labor-intensive step was not solely in service of a third, more in-depth step focused on inductively exploring *how*

the media was doing so in the relevant sub-corpus. Nonetheless, researchers without such an incentive may still want to be more discerning in the selection of their corpus—in other words, in the initial classification steps—than keyword searches allow, and we suggest an expedited way of doing so with generative LLMs.

Third, the reason that only a minority of articles was originally classified as being about economic inequality was because we had developed—through first deductive and then subsequent inductive iterations through the data—a strict but elaborate *definition* of both the concept of economic inequality and its empirical manifestation in the current era of rising income, wealth, and earnings inequality. This definition was spelled out in a 14-page, single-spaced, hand-coding guide, as well as a set of appendices. (A one-page outline of the definition of inequality from the original study is provided in Appendix A in the online supplement.) Namely, the guide sought to isolate all articles that *potentially* conveyed the idea, either explicitly or implicitly, of inequalities stemming from a wide range of phenomenon. Executive compensation and other firm-level wage-setting policies, minimum wage regulations, attacks on collective bargaining, social insurance and safety net spending, and changes to the tax code, all could result in discussions of income inequality without the term inequality itself ever being used. Conversely, economic phenomenon that was *not* about *income/wealth/earnings* inequality, such as trends in unemployment, needed to be defined as negative cases in the guide as well.

We provide an example of the challenges of this coding task using an article selected from the corpus and reproduced in Figure 2. This article was hand coded as "implicitly" mentioning inequality, which is a coding category described in Appendix A, section 4a in the online supplement. According to the original codebook, the classification of this article requires contextual knowledge that two economic groups in hierarchical relation to one another are being mentioned: yachts are for the rich and welfare is for the poor without the article using the words "rich" or "poor" or "low" or "high" income. Moreover, the reader must be able to identify, via the gently snarky tone in the article, the implicit connection between yachts, welfare, and inequality through a holistic reading of the entire document, as these references do not occur together in a single sentence or paragraph and are never explicitly compared in the article (leading to the "implicit" rather than "explicit" designation by the original hand coders). In short, the very nature of our task—and, we contend, many others like it—means that text cannot be analyzed at the level of words, sentences, or paragraphs as is common in most other classification research in the social sciences.[4]

A Modest Proposal:
Public policies that perform

U.S. News & World Report
August 10, 1987

Section: U.S. NEWS; The Presidential Race; Pg. 18

Instead of Crockett and Tubbs boogie-ing their way around Biscayne Bay it was a few out-of-shape gumshoes from the Massachusetts Department of Revenue walking around the North Shore yacht clubs taking down the tag numbers of big boats. Call it Marblehead Vice, but it was more productive than Miami Vice as far as the tax man was concerned, because it yielded $ 5 million from more than 2,400 owners who had dogged sales taxes on the purchase of their boats. "Those guys hated it when the harbor patrol posted those day-glo 'seizure' stickers on their bows and slapped a 'Denver boot' [lock] on their moorings," chuckled state tax boss Ira Jackson.

A "golden age" for hotshots

The prominent burst of big-time yachtsmen and, later, of owners of planes, helicopters and fancy cars was Jackson's brainchild. It was his notion that publicity from stepped-up enforcement would be a bracing prelude to the state's innovative tax-amnesty program in late 1983. Sure enough, 52,000 folks paid $ 86.5 million in back taxes during the amnesty window, and collection of delinquent taxes in the past three years has soared $ 1.3 billion since enforcement was boosted after the amnesty.

Jackson, 39, is one of several thoroughbreds in the stable of Governor Michael Dukakis who have helped project the image of his administration as path-breaking and uncommonly effective. If it is the Golden Age for hotshot entrepreneurs in the Bay State, it's also a heyday for "policy entrepreneurs" with ideas on how to make government work better.

Best known among Dukakis's innovations is the Employment and Training Choices (ET) program, which since 1983 has helped more than 30,000 welfare recipients get off the dole and into jobs that pay an average of nearly $ 6 an hour. The program is relatively expensive, paying $ 3,400 per placement for training, day care, health care and transportation for welfare mothers to their jobs. But it has finally started to put a dent in the state's welfare polls, paring them by 5,000 (about 7 percent) last year, according to Public Welfare Commissioner Charles Atkins, a nationally admired figure. ET is voluntarily, unlike "workfare" proposals that require recipients to take public-service jobs in order to receive their checks.

The state's newest innovation marries the tax man and the welfare problem. Enforcement of child-support laws is not being handled by Jackson's "junkyard dogs," who will be tracking down absent dads (through blood tests if necessary) and getting their employers to deduct support funds from their paychecks. The tax code has long been seen as a social-policy tool. Dukakis and Jackson are showing that tax enforcement itself can be an engine for social change, too.

Figure 2. Example news article discusses inequality without using the word inequality.
Note: Article ("A Modest Proposal: Public Policies that Perform" 1987) taken from sample of 1,253 newsweekly articles hand coded for content on inequality. This article was hand coded as "implicitly" about inequality by the hand coders (see Table 1). All models classified this article correctly in the "relevant" test with no definition, and all but Llama3 classified it correctly in the "inequality" test.

Table 1. Categories and Binary Schemas From Hand-Coding Guide and Analysis.

Categories and Binary Schemas	Number
Categories (shorthand in hand-coding guide)[a]:	
Relevant Articles	
Inequality[b] (relinequality)	1
Economic Conditions (releconomy)	2
Changes in Wages and Income (relchanges)	3
Irrelevant Articles (irrelevant)	4
Binary Classification Schemas:	
Relevant versus Irrelevant	1/0
Inequality versus Other Relevant	1/0

Notes: [a] In the hand-coding guide from the original study (McCall 2013), these categories are given shorthands which are referenced in some of our prompts because the guide itself is evaluated by the LLMs. [b] This category is broken down into two categories (explicit and implicit inequality) in the original work and in the replication but is simplified in most of the present work to be a single category (except for tests in Appendix G in the online supplement, Table G1, Panel B, Rows 1 and 2).

Drawing from the prior research but adapting it to the generative LLM context, our full classification task is presented in Table 1. In terms of the categories of analysis, articles that were hand coded as mentioning *economic inequality*, both explicitly and implicitly, are referred to as category 1. The remaining articles fell into three mutually exclusive categories: a category that mentions *employment conditions* without mention of wages or income (category 2); a category that mentions *wages and income* for single groups of workers or occupations without mention of differences among groups of higher (e.g., rich) and lower (e.g., poor) economic status (category 3); and a final category that was deemed *irrelevant* in the original project because the discussion focused on other countries or on gender and/or racial inequality without any mention of income or class inequality (category 4). The first three categories are also together denoted as *relevant* articles.

In terms of classification schemas, our prior replication of the original study using non-LLM computational methods examined multiple classification schemas (Nelson et al. 2018), whereas we have simplified the classification schemas for our analyses given the variations that we introduce instead around the definition of the targeted concept of inequality and with respect to different kinds of LLMs. In the prior replication, we examined schemas with two categories (e.g., inequality versus non-inequality articles, or category 1 versus categories 2–4) or three categories (inequality versus other economic versus irrelevant articles, or category 1 versus categories 2–3 versus category

4). We focus on two different binary schemas in this article because our prompting workflow evolved over time, described further below, into a two-step sequence of first removing irrelevant articles (category 4 versus categories 1–3) and then isolating articles on inequality (category 1) from articles on other economic topics (categories 2–3) among the relevant articles (Gilardi, Alizadeh and Kubli 2023).

In assessing the performance of LLMs in classifying articles into these schemas, we again draw from our prior replication for guidance. In that work, we examined whether three categories of computational methods—dictionaries, supervised machine learning, and unsupervised machine learning—could produce results comparable to hand coding, We found that supervised machine learning methods performed both well and the best of the three tested methods. Our objective in terms of assessing accuracy in the present work, then, is to determine whether LLMs can match or exceed the performance of supervised machine learning methods on the same corpus of hand-coded articles but using natural language as the input and output.[5] Our efforts focused on using the coding guide (created in the original project) to develop prompts for generative LLMs that will enable them to correctly classify the corpus of articles into the binary classification schemes described in the preceding paragraph. We tested three open-source models that have previously performed well: Llama3:70b, Gemma2:27b, and Llama3.1:70b, comparing the results from the open-source models to the industry standard but proprietary GPT4 (OpenAI 2023).

Prompting Strategies

We have mentioned the term "iterative" several times in the course of describing the qualitative hand-coding process of alternating between deductively and inductively derived instructions for identifying target concepts in documents. This is represented more formally by the first box of our proposed workflow in Figure 1. We now highlight two broad approaches that researchers can take in adapting this process to generative LLMs, as represented in the second box of the figure.

The first involves interacting in real time with generative LLMs in the chatbot or context window with queries in natural language, where the goal is to develop and refine a set of coding instructions for the LLM to successfully execute with a set of documents. This approach will work especially well when a coding guide does not already exist, and Ibrahim and Voyer (2024) provide an excellent description of this approach. The second approach involves converting an already existing coding guide, or a new

but well-developed coding guide, into a prompting strategy that is run non-interactively on the full corpus (or smaller sub-samples in a preliminary testing phase) through an API (Application Programming Interface) that is called using simple programming code. In practice, researchers often use both approaches, depending on how much interactive testing, along the lines of the first approach, is needed when first embarking on a project. As described in this section, we found the need for the first approach to be limited and thus moved quickly to the second approach.

We began with some initial discussion and experimentation because of the wide range of parameters we and others have considered in this endeavor (Ollion et al. 2024; Törnberg 2024). One set of elementary considerations concerned the use of specific terms of art to describe inequality—simply, whether LLMs first needed a tailored definition of the specific terms used in a more elaborate definition of the ways that inequality could be covered in the media. Initial inquiries using the interactive chatbot window, along the lines of the first approach described above, convinced us that systematic testing and defining of terms of art was unnecessary because they were already recognizable by the pre-trained LLMs. Our tests then followed the second, non-interactive approach of prompting generative LLMs through an API. We describe our final prompting strategy in terms of the *structure, length,* and *sequence* of the prompts.

There are many ways to *structure* a prompt to provide background knowledge for a generative LLM to complete a task. We experimented with three main features. First, we provided references drawn from our data to enable the LLM to better understand our specific task: sample articles and their codes in a binary schema (see again Table 1) were inserted into the prompt as examples to further train the model, which is called "few-shot'" learning in LLM speak (Brown et al. 2020; Chae and Davidson 2025; Rytting et al. 2023). When no sample input of articles and codes is inserted into the prompt, this is called "zero-shot" learning (Kojima et al. 2022). Few-shot and zero-shot learning are different ways of structuring a prompt, which can have implications for the length of a prompt but need not do so. In our analyses of long documents, we found that the length of the sample input in few-shot learning most likely affected the performance of the task negatively (as discussed further below).

Second, we manipulated whether the prompt includes a definition (e.g., "Inequality is defined as …") in addition to a command (e.g., "If an article is about inequality, respond yes."). Both types of instruction are included in the prompts we discuss below, though we focus especially on the construction of definitions. As a baseline, we also experimented with prompts without

definitions of inequality, allowing the LLMs to rely on their own training data to determine what it means to cover the topic in the media. Finally, all prompts begin with a directive about the type of persona that LLMs should emulate (e.g., "You are a helpful assistant.") and in some cases also include a more extended description of the task to be completed by LLMs. These are called system prompts and meta-statements, respectively. Because the natural language interface permits the structure of a prompt to vary in nearly infinite and not currently well-understood ways, we cannot say with certainty that the structure we adopted is the most optimal, only that it produced the most consistently accurate results among the variations we tested.[6]

The *length* of a prompt is a major consideration in work like ours because of the complexity of the subject matter, the length of the articles (mean length is 1,089 words and 1,389 tokens), and the limits on the length of prompts imposed by various versions of LLMs (i.e., the maximum number of tokens, a sub-word unit, is 8,182 for Llama3, Gemma2, and GPT 4.0, and 128,000 for Llama3.1). Although we hypothesize that a major advantage of LLMs is their ability to recognize complex thematic content in long documents, it is still possible that longer prompts perform less well than shorter prompts even when they are within the token limit. In other words, longer token limits may enable the input of longer definitions and documents but there is no guarantee that accuracy will increase as a result (at least at the current stage of LLM development).

Finally, the *sequencing* aspect of prompting strategies can operate in a variety of ways—sequencing of code within a single prompt, which we refer to above as part of the *structure* of the prompt, or sequencing of multiple prompts. The latter is the type of sequencing that we are referring to and that is connected to the format of our data. As described above and in Table 1, the data are partitioned into four mutually exclusive categories, with some groups likely sharing more porous boundaries than others, even though all articles contain economic content of some kind. A comprehensively inductive sequencing strategy might have been to test all logical sequence permutations. Instead, we qualitatively decided to test, as the first target category, what we hypothesized to be the most distinct of the categories, which is the irrelevant category (category 4 versus categories 1–3). This proved to be successful, and thus we moved on to testing a second step of the sequence, which was to isolate inequality articles among the remaining relevant articles (i.e., category 1 versus categories 2 and 3).

These tests of different prompt *structures*, *lengths*, and *sequences* were at first researcher-created and largely deductive: we began with the detailed

hand-coding guide and a summary of the guide that appeared in the original and replication studies (reproduced in Appendix A in the online supplement) to inform our own design of the structure, length, and sequence of the prompts (see also Törnberg 2024). Subsequently, we developed a core prompt structure within which we experimented with an automated, inductive version of the concept definition performed nearly entirely by LLMs (Wu et al. 2023). Our final tests took a stepped approach, beginning with no definition, then using definitions produced by researchers themselves, and finally using LLM-generated definitions, the latter two mimicking the use of hand-coding guides traditionally used for complex coding tasks. Crucially, this approach not only encourages researchers to be deeply embedded in their coding guides and definition and prompt development, something we consider integral to qualitative research, it also produces multiple estimates of the classification task that later can be compared to one another in a reliability and validation step (see the third box in Figure 1), as we demonstrate in the next section.

Results

We first present the final, core prompt structure that is replicated across tests, and then follow with three subsections of results from the experiments with no definition (the baseline), a researcher-generated definition, and an LLM-generated definition, respectively. In the final two subsections, we discuss the possibility of qualitatively interpreting the natural language output from the LLMs and of utilizing the output across prompts and models to calculate interprompt and intermodel agreement metrics for reliability and validation purposes.

Core Prompt Structure

Table 2 provides a description of the core prompt structure that we implement. First, we used the system prompt "You are a news classifier," aligning with our task and priming the LLM with the word *news,* a genre of text. Second, we included a meta-statement: "We categorize articles that are related to issues of income inequality, changes in income or wealth, general economic conditions." This statement incorporated the word "categorize" followed by key terms from the qualitative codebook, guiding the output toward our conception of the relevant classification categories. Third, we added a definition of the target category, depending on which step in the sequence of tests we were at (i.e., the first step of identifying all

Table 2. Core Prompt Structure for Zero-Shot Prompts.

Zero-shot Prompts

Panel A: Relevant vs. irrelevant articles (categories 1-3 vs. category 4; $N = 1{,}253$)

```
"role": "system",
"content": "You are a news classifier."
"role": "user",
"content": "We categorize articles that are related to issues of
  income inequality, changes in income or wealth, general
  economic conditions."
"role": "user",
"content": "Read this definition: "+ irrelevant_definition[a],
"role": "user",
"content": "Read this article: "+ text,
"role": "user",
"content": "Is the article relevant? Answer relevant or
  irrelevant, and why in 1 sentence."
```

Panel B: Inequality vs. non-inequality among relevant articles (category 1 vs. categories 2-3, $N = 786$)

```
"role": "system",
"content": "You are a news classifier."
"role": "user",
"content": "We categorize articles that are related to issues of
  income inequality, changes in income or wealth, general
  economic conditions."
"role": "user",
"content": "Read this definition: "+ inequality_definition,
"role": "user",
"content": "Read this article: "+ text,
"role": "user",
"content": "Does the article reference American economic
  inequality? Respond with 'Yes' if article meets any or all
  criteria referencing American economic inequality and 'No' if
  article meets none of the criteria, and explain why in 1
  sentence."[b]
```

Notes: [a] As discussed in the text, we provide a definition of irrelevant rather than relevant articles. [b] This command is provided when researcher- and LLM-generated definitions are included, otherwise the command is shorter when no definition is provided.

relevant articles in Table 2, Panel A, or the second step of identifying inequality articles among all relevant articles in Table 2, Panel B), prefaced with "Read this definition: ". Fourth, we included the actual article to be classified, prefaced with "Read this article: ". Finally, we issued our command, such as "Does the article reference American economic inequality? Respond with 'Yes'…." We found that this prompting strategy resulted in more consistent output with clear binary indicators followed by a justification than other prompt structures we tested, which are described in Appendix C in the online supplement.

Looking at the zero-shot examples of code in Table 2, we see the sequence of two tests represented in Panels A and B, respectively. The *irrelevant_definition* and the *inequality_definition* variables take on the values of either no definition, researcher-generated definition, or LLM-generated definition. The variable *text* contains the title of each article followed by the entire text of the article, truncated as necessary if the prompt length goes beyond the maximum context length of each respective LLM. For our few-shot tests, not shown in Table 2, we appended two randomly selected sample articles as input followed by the output for each after the final command prompt. Note that we only performed the few-shot learning tests for the second step in the sequence of prompts shown in Panel B, because this step represented a more challenging task, and thus our total number of tasks is three (i.e., Panel A and Panel B in Table 2, and a few-shot version of Panel B). For all tests, we set a seed and set the temperature parameter to 0 to minimize output variation and to enhance reliability and reproducibility,[7] and we set the context window to the maximum allowed for each model. Table 3 shows the weighted average recall, precision, and F1 scores across the three tasks in the rows, varying the LLM and the type of definition we provided in the columns.

Baseline: No Definition

In our first baseline test ($N = 1{,}253$), providing no definition, using the entire corpus of articles (categories 1–4) and identifying relevant versus irrelevant articles (i.e., categories 1–3 versus category 4), we achieved weighted F1 scores of 0.74 (Llama3), 0.80 (Llama3.1), 0.79 (Gemma2), and 0.83 (GPT4) (see Table 3, row 1, columns 1–4). With careful prompt structures but no definition of the concept we were using to classify documents, and no examples to learn from, these accuracy scores are on par with acceptable human-coding intercoder reliability scores (O'Connor and Joffe 2020), and with the acceptable F1 scores for traditional supervised machine learning in the prior replication study.[8]

Table 3. Accuracy Metrics for Three Classification Tests (rows) by LLM and Definition Input (columns).

	No Definition				Researcher Definition				LLM-Generated Definition			
	Llama3 (70b)	Llama3.1 (70b)	Gemma2 (27b)	GPT4	Llama3 (70b)	Llama3.1 (70b)	Gemma2 (27b)	GPT4	Llama3 (70b)	Llama3.1 (70b)	Gemma2 (27b)	GPT4
Relevant:	P: 0.74	P: 0.80	P: 0.78	P: 0.82	P: 0.76	P: 0.79	P: 0.82	P: 0.83	P: 0.82	P: 0.79	P: 0.79	P: 0.81
Zero Shot	R: 0.73	R: 0.81	R: 0.79	R: 0.84	R: 0.74	R: 0.78	R: 0.83	R: 0.83	R: 0.82	R: 0.82	R: 0.83	R: 0.83
($n = 1253$)	F1: 0.74	F1: 0.80	F1: 0.79	F1: 0.83	F1: 0.73	F1: 0.78	F1: 0.82	**F1: 0.83**	F1: 0.82	F1: 0.80	F1: 0.80	F1: 0.81
Inequality:	P: 0.71	P: 0.72	P: 0.73	P: 0.75	P: 0.74	P: 0.72	P: 0.74	P: 0.77	P: 0.76	P: 0.73	P: 0.75	P: 0.74
Zero Shot	R: 0.72	R: 0.72	R: 0.73	R: 0.74	R: 0.75	R: 0.72	R: 0.74	R: 0.77	R: 0.77	R: 0.73	R: 0.75	R: 0.74
($n = 786$)	F1: 0.70	F1: 0.72	F1: 0.73	F1: 0.74	F1: 0.73	F1: 0.71	F1: 0.74	**F1: 0.77**	F1: 0.75	F1: 0.73	F1: 0.75	F1: 0.73
Inequality:	P: 0.72	P: 0.73	P: 0.61	P: 0.75	P: 0.69	P: 0.64	P: 0.49	P: 0.75	–	–	–	–
Few Shot	R: 0.73	R: 0.75	R: 0.77	R: 0.75	R: 0.72	R: 0.77	R: 0.94	R: 0.75				
($n = 786$)	F1: 0.73	F1: 0.70	F1: 0.60	F1: 0.75	F1: 0.69	F1: 0.62	F1: 0.59	**F1: 0.75**				

Note: Average weighted precision (P), recall (R), and F1 metrics across all tests. F1 is the harmonic mean of recall and precision. All tests use the core prompt structure presented in Table 2.

In our second baseline test ($N = 786$), providing no definition of inequality, using only relevant articles (i.e., excluding articles hand-coded in category 4), and identifying whether an article mentions economic inequality (i.e., category 1 vs. categories 2–3), we achieved weighted F1 scores of 0.70 (Llama3), 0.72 (Llama3.1), 0.73 (Gemma2), and 0.74 (GPT4) (see Table 3, row 2, columns 1–4). While lower than the first baseline tests, which is perhaps expected given the more nuanced task, these weighted F1 scores are still within acceptable accuracy scores for most complex hand-coding tasks. For these two baseline tests, GPT4 achieved the highest F1 scores, but only by 0.03 and 0.01 points, respectively (on a 0.0 to 1.0 scale).

For our third baseline test ($N = 786$), the accuracy metrics across the four models varied. Providing two example articles (one about inequality and one not, each randomly selected) improved Llama3's performance slightly over the second baseline test, with a weighted F1 of 0.73, and a comparable performance with GPT4, with a weighted F1 of 0.75. By contrast, the accuracy was worse with Llama3.1 and Gemma2, with weighted F1s of 0.70 and 0.60, respectively. We think this might be due to the inclusion of three full-length articles—two examples and one to be classified. The prompt context window may have become too long for these LLMs to properly parse. Few-shot learning thus might work better for short-text (e.g., sentence- or paragraph-length) classification tasks (e.g., Chae and Davidson 2025; Do, Ollion, and Shen 2022; Rytting et al. 2023) than for ones such as ours. Because these initial few-shot tests were not encouraging, we did not engage in additional systematic tests (e.g., varying the types or numbers of articles selected), which should be the subject of further research.

Researcher-Generated Definition

Seeking to match and improve upon these results (and following suggestions by Törnberg 2024), we generated definitions of irrelevant and inequality articles for the two steps in the sequence of tests, respectively. This portion of the prompt was issued in the third section of the core prompt in Table 2 (i.e., "Read this definition:" followed by our definition). Our two definitions are reported in Appendix D in the online supplement (in the sections under researcher-generated definitions). We found a concise, one-paragraph excerpt from the original coding guide of the "irrelevant" category and used that as our researcher-generated definition (i.e., the *irrelevant_definition* variable in Table 2) for that step. The analogous definition of inequality (i.e., the *inequality_definition* variable in Table 2) is based on the one-page summary definition of media coverage of inequality published in the prior work (reproduced in

Appendix A in the online supplement), which itself is a summary of the detailed 14-page hand-coding guide. Still, it is a long definition if incorporated verbatim, and therefore we sought to substantially reduce it.

Specifically, the original summary is comprised of four sections or criteria defining "types of inequality," "causes and policy solutions associated with inequality," "social class groups," and "relational or comparative language connecting social class groups." These four sections in turn had subsections that defined, for instance, the "types of inequality" and "social class groups" that were to be identified in media coverage of inequality. Our initial tests reported in Appendix G in the online supplement consisted of variations on these four section headings and subsection descriptions to help clarify the intended content of the four main sections. For instance, we needed to convey whether all four criteria needed to be present or some combination of them. In the end, we settled on a definition with three bullet points, any one of which was sufficient to identify an article on inequality. This decision rule is reflected in the command statement (see bottom of Table 2, Panel B).

The results from these tests are reported in Table 3, columns 5–8. The F1 scores across the three tasks (in the rows) and the four models (in the columns) were similar to the F1 scores from the baseline test, sometimes slightly lower, sometimes slightly higher. In other words, providing a definition of inequality did not consistently improve performance. The test for the first step in the sequence (relevant or irrelevant) had weighted F1 scores of 0.73 (Llama3), 0.78 (Llama3.1), 0.82 (Gemma2), and 0.83 (GPT4), compared to 0.74, 0.80, 0.79, and 0.83 for the baseline tests without definitions, respectively. For the second step in the sequence (mentioning inequality or not among relevant articles), the weighted F1 scores were 0.73 (Llama3), 0.71 (Llama3.1), 0.74 (Gemma2), and 0.77 (GPT4), compared to 0.70, 0.72, 0.73, and 0.74 for the baseline tests without definitions, respectively. For the few-shot test, the weighted F1 scores were 0.69 (Llama3), 0.62 (Llama3.1), 0.59 (Gemma2), and 0.75 (GPT4), compared to 0.73, 0.70, 0.60, and 0.75 for the baseline tests without definitions, respectively. For these tests, GPT4 again performed the best, but not more than 0.03 points better than the best-performing open-source model, except, notably, in the few-shot tests where it was far superior.

We contemplated additional tests that would modify and expand the definition in different ways or use a larger number of articles as examples in the few-shot learning approach, but most LLMs' token limit precluded us from greatly increasing the length of the definition and number of input articles. Simultaneously, we tested an alternative approach to providing a definition—allowing the LLM to generate its own definition from the codebook.

LLM-Generated Definition

One of the purported capabilities of LLMs is to accurately summarize documents into easily digestible bullet points (Zhang et al. 2024). We thus tested the possibility that LLMs could summarize nuanced codebooks into definitions to guide each LLM in its classification task, which also further automates the LLM-assisted coding process. This was accomplished in two separate, pre-processing prompts followed by a hand-editing step to merge the output from the two separate prompts. In the first pre-processing prompt, we gave each LLM the full coding guide and a command to create a longer definition (than our researcher-generated definition) with 7–10 bullet points of the irrelevant or inequality categories (depending on which step in the sequence we were testing). We did the same in the second pre-processing prompt, except that a detailed appendix to the main coding guide was submitted instead of the main guide itself (prompts and results not shown). The results from the two steps were merged and duplicates were removed, all by hand. Note that the LLM-generated definitions differ across the models, as each model generated a different definition. These definitions are reported in Appendix D in the online supplement (in the sections under LLM-generated definitions).

The results from using the LLM-generated definition are reported in Table 3, columns 9–12. The accuracy metrics are similar to or marginally higher compared to the previous tests. For the first step in the sequence (identifying relevant vs. irrelevant articles), the weighted F1s were 0.82 (Llama3), 0.80 (Llama3.1), 0.80 (Gemma2), and 0.81 (GPT4). For the more nuanced second step (identifying inequality vs. non-inequality articles), the LLM-generated definition performed marginally better across all models except GPT4, compared to both the baseline (no definition) and the researcher-generated definition test, with weighted F1s of 0.75 (Llama3), 0.73 (Llama3.1), 0.75 (Gemma2), and 0.73 (GPT4).

Natural Language Output

The input in the prompts for these models is in natural language, but here we discuss the fact that the output is also in natural language, which presents both drawbacks and affordances. One drawback is that the model may not always provide the desired binary classification (e.g., relevant/irrelevant or yes/no), necessitating an additional step where the researcher must interpret the output to fit the desired classification (which is discussed further in Appendix C in the online supplement). However, the natural language

output can also benefit researchers, particularly those in qualitative fields. Recall that our instructions for the models' output included a request for a one-sentence summary. Researchers can use this summary to assess which words may influence the LLM's classification decisions. This serves as a validity check on the output itself and provides additional qualitative insights into the corpus, the codebook, and the chosen categories. For the tests using the core prompt structure in Table 2, the model did reliably produce a yes/no or relevant/irrelevant output and an additional justification sentence(s).

For example, in the Llama3 model reported in Table 3 (row 1, column 5), the output reliably produced a single-word classification for every document, and the subsequent sentence provided a substantive justification, for example:

> Irrelevant, because while the article mentions immigration reform and its potential impact on the economy, it does not explicitly discuss income or wealth inequality in the United States.

> Relevant, because although the article primarily discusses affirmative action and racial inequality, it also touches on broader issues of economic inequality, such as how government policies can create unfair advantages and concentrate wealth among a few individuals.

The sentence-length justification often referenced aspects of the definition provided. For example, the definition provided explicitly mentioned that articles about racial inequality should be considered irrelevant, except where forms of class inequality are also mentioned. The sentence in the second example above gave this precise rationale, stating that although the article was primarily about racial inequality, it was also about the concentration of wealth. The justifications also typically detail which groups are mentioned (e.g., immigrants), which could provide further insight into how inequality is discussed over time, though we did not do that here. As is the case with training research assistants, this informational justification provides insight into the entire classification endeavor.

Interprompt and Intermodel Agreement Tests

Finally, assessing the reliability and validity of the classification task is a crucial component of both qualitative and quantitative coding procedures. Fortunately, the multiple estimates generated from our tests of different prompts and models enable us to conduct such an analysis (as shown in

the third box of Figure 1). Recall that the F1 scores presented in Table 3 varied across prompting strategies and models, ranging from 0.59 to 0.83, but this variation was not consistent across prompts and models. For example, sometimes Gemma2 performed the best of the open-source models, sometimes the worst. Similarly, the LLM-generated definition for the second (inequality) test produced the best F1 score for Gemma2 compared to other prompts, but the worst for GPT4. To our knowledge, there is currently no way to know a priori which models and prompting strategies might be more or less accurate. In our case, we know the variation in accuracy because we were working with already hand-coded documents. Yet one goal of introducing LLMs into the qualitative coding pipeline is to reduce the time spent hand-coding documents or at least to redirect that time in a more strategic and efficient manner.

We propose leveraging the multi-model analysis, and resulting variation across models, for reliability and validity testing of classification tasks with no pre-determined ground truth (see, e.g., Zhang et al. 2024), as well as for benchmarking (as was done in Do, Ollion, and Shen 2022 and Rytting et al. 2023). To do so, we used standard "interrater reliability" measures to calculate two forms of prompt and model agreement from our output: *interprompt agreement*, defined as the degree of agreement within one LLM across prompts, and *intermodel agreement*, defined as the degree of agreement within one prompt across different LLMs. Full results are presented in Appendix E in the online supplement, but, in summary, interprompt and intermodel Fleiss' kappa (Fleiss 1971) ranged from 0.62 to 0.88. Like the F1 metrics, these metrics range from low to high.

Scholars without pre-hand-coded documents could use these metrics in three ways. First, similar to the use of interrater reliability, the metric itself can be used to assess stability across prompts and models—a test of the clarity of the classification task itself. Second, scholars could use a "majority voting" strategy to produce the ultimate classifications. For the latter, we assigned the classification agreed on by two or more prompts or models and recalculated the F1 scores across our tests (see Appendix E in the online supplement). Using this majority voting method did not achieve significantly higher F1 scores (the highest F1 score in these tests was 0.84 versus 0.83 in the original tests in Table 3), but the floor was much higher (the lowest F1 score was 0.75 versus 0.59 in our original tests). In other words, the majority voting method can screen out low-accuracy outliers.

Third, identifying documents with full agreement among all prompts or models can help researchers strategically decide where to invest time in qualitative coding. Even when LLMs and prompts show high agreement,

researchers should remain cautious, as such consistency may reflect shared model biases or training data rather than accuracy vis-à-vis human evaluation. Validation involving human judgment remains essential. Researchers could use these agreement metrics to, for example, identify documents where all prompts/models agreed on the classification, validate a selection of those by hand, and then intensively analyze the (hopefully smaller) set of documents with disagreements, also by hand. In our case, the three prompts or three models agreed on one classification for 74 to 91 percent of the documents (see Appendix E in the online supplement). This approach will not necessarily reduce the total amount of time spent interpreting and hand-coding documents, but it could allocate that time in more efficient and strategic ways.

Discussion

We have examined whether LLMs can classify documents into given categories on par with either human coders or previous supervised machine learning methods, but our main contribution has been to focus on the process of qualitative coding itself via LLMs using data from an original qualitative hand-coding project. Interacting with generative LLMs using natural language is a marked shift from previous computational text analysis methods that required translating text and instructions into machine-readable code. Some code is also necessary for the present work (e.g., see Table 2), but the shift to generative LLMs means that qualitative researchers can, more or less, replicate traditional methods involving the construction of comprehensive codebooks and the iterative testing of the reliability of those codebooks (see Figure 1). We systematically varied LLM prompts derived from such a codebook and found that LLMs can achieve classifications comparable to hand coding and supervised machine learning. Despite many remaining challenges discussed below, this finding suggests not only that LLMs may eventually be able to serve as reliable coding assistants, but that they could also offer scalability (e.g., analyzing larger datasets), flexibility (e.g., easily modifying classification criteria and building on pre-trained models), and efficiency (e.g., dramatically reducing or strategically redirecting the need for hand coding).

Main Findings

Our tests revealed several key features of prompts and models that should be considered when adopting LLMs for qualitative coding projects. We summarize the lessons from each in turn (prompts and models), but we first reiterate

the nature of our data and classification task, which shaped our approach to LLMs in crucial ways. Our units of analysis are long documents (U.S. news-weekly articles from 1980 to 2012), which are relatively understudied in the social science literature on classification analysis with LLMs despite the known strengths of LLMs in thematically and holistically summarizing large corpora (Karell et al. 2024; Zhang et al. 2024). Within long documents, we sought to identify the multi-faceted concept of economic inequality, which potentially is elaborated piece by piece across multiple sentences and paragraphs (see Figure 2). The documents also contained adjacent economic content, making them "noisy" and challenging for humans to code. In all these ways, our findings should generalize to other qualitative coding projects concerned with concept identification in rich, long-form documents, such as oral history or interview transcripts, organizational documents, and legacy/historical media.

Concerning the development of prompts for generative LLMs using these types of data and classification tasks, we highlight four findings. First, even without providing a definition of the concept to be labeled, the baseline test was performed on par with acceptable interrater reliability metrics for hand-coding and accuracy metrics for traditional supervised machine learning. For our use case, which involves classifying English-language newspapers—known to be part of the training data for many LLMs—this means we can effectively prompt LLMs using the same, though much reduced, language we would use to train research assistants and still obtain reliable and accurate classifications. This also implies that LLMs come pre-trained with relevant textual knowledge for our use case, and likely those of many other researchers in the social sciences (though this should be checked whenever possible if open-source models allow it).

Second, and to our surprise, the addition of researcher- and LLM-generated definitions to the prompt did not consistently improve the accuracy of the classification tasks over the baseline prompt without a definition. The researcher-generated and LLM-generated definitions were generally comparable in accuracy to the baseline, though the longer, LLM-generated definitions did perform the best for some models and some tests. For those starting a qualitative project without a ground truth, in other words, we cannot provide direction on how detailed of a definition is needed to obtain the highest accuracy, which instead will necessitate a careful, iterative approach (Törnberg 2024). Such an approach is best practice in any case, and it should include both hand coding for validation purposes and the calculation of intermodel and interprompt agreement scores to locate potentially easy-to-classify cases and those that are less so (see

section "Interprompt and Intermodel Agreement Tests" and the third box of Figure 1).

Third, for our use case, the zero-shot prompts performed better than the few-shot learning prompts (note that all LLM-created definitions were embedded in zero-shot prompts). This may result from the small number of articles included in the few-shot learning prompts, which itself is a consequence of the token limit for most of the LLMs we tested. Or it could be a result of the length of our articles in a different way: adding two example articles on top of the article to be classified introduced too much noise into the prompt structure even when the full prompt was within token limits.[9] Future work should more thoroughly test the impact of varying the number, type, and length of articles submitted in the few-shot learning prompts, particularly with models (such as Llama3.1) that have expanded token limits. At the same time, it may be possible to include all articles and their labels as input to LLMs, in addition to providing the detailed coding guides (as we did), in order to generate a definition (whose length and detail could also vary) from the LLM. This would mimic a supervised machine-learning environment in which a substantial volume of already labeled data is needed to inform/train the model (Ziems et al. 2024).

Fourth, and finally, our testing of different prompting strategies suggests both the importance of prompting strategies and the unnerving sensitivity of LLM output to variations in prompt structure. We found, like others (e.g., Törnberg 2024), that structured prompts were advantageous. For instance, placing the command for the desired output at the end of the prompt produced more consistent classification output, perhaps due to the volume of textual input in our prompts (see Appendix C in the online supplement). Although non-trivial variation in LLM output across prompts poses a serious challenge when previously hand-coded output is unavailable, iterating the process by varying prompts (e.g., by modifying definitions) and models (e.g., by using multiple LLMs), and assessing agreement across the output classifications, is akin to iterating a codebook and calculating human-rater reliability scores while training research assistants. Using interprompt and intermodel majority voting strategies further mitigates the risk of low-performing outliers, though we emphasize that even when models or prompts produce the same classification, human validation is still needed. In short, the variation in LLM output could be interpreted as mimicking the hand-coding process, and researchers could use this variation for insight into their categories and definitions (Ibrahim and Voyer 2024).

Concerning our tests across different open- and closed-source models, and models of substantially different sizes, we first highlight two somewhat

contradictory findings and then a third finding. First, although the full GPT4 model achieved marginally better accuracy metrics almost across the board, these metrics were not substantially better than the smaller, pre-quantized, open-source models, ranging from 0.01 to 0.03 points (on a 0.0 to 1.0 scale) over the best performing open-source LLM. These findings, combined with both the cost and the extensive methodological, ethical, and environmental problems associated with using foundational LLMs for academic research (discussed in Section "Variation among Generative LLMs"), confirm that academic researchers can and should use open-source LLMs for text-based classification tasks. At the same time, and secondly, GPT4 performed markedly better in the few-shot tests, suggesting that it can better manage larger streams of input data. Moreover, accuracy improvements of even a few points may not be inconsequential for some qualitative researchers and projects.

Third, we also observed a potential for unpredictable differences among the models themselves. Llama3, for example, achieved relatively high weighted F1 scores across a wide range of tests, yet Llama3.1, an arguably superior model, performed worse in the first few-shot learning test. And Gemma2 was the highest performing open-source model in many of our zero-shot tests, but all but collapsed in the few-shot tests. Differences in training data and weights and the fine-tuning done by engineers can lead to unpredictable behavior when LLMs are applied to tasks that depart further from the standard benchmark tests (Boelaert et al. 2025). In the case of open-source models, researchers have access to the fine-tuning process, but it is not always clear how that fine-tuning impacts model behavior across different types of prompts and tasks, and we return to this point below.

Together, our findings suggest that generative LLMs can be a productive and reliable tool for qualitative researchers, especially those without advanced computer programming skills, who want to systematically analyze a collection of texts, either inductively or iteratively, using standard qualitative methodologies. To this end, in Table 4, we highlight three general takeaways that may help researchers consider how to apply these models in their work, given the rapidly changing terrain of LLM development.

Limitations

Our experiments presented both a difficult and a straightforward test case for LLMs, suggesting both benefits and limitations of our work. Our classification task was particularly difficult for generative LLMs because we were instructing LLMs to predict whether there was *any* discussion of inequality

Table 4. High-Level Summary of Findings.

Guideline	Key Insight	Why It Matters	Practical Application
Researcher-LLM-Researcher Workflow	Use LLMs as part of an iterative process, with researchers designing tasks, LLMs making an initial pass, and humans validating/refining results.	Ensures LLMs support, rather than replace, qualitative judgment, and redirects researcher efforts in strategic ways.	Use LLMs for first-pass coding and focus researcher effort on validating clear cases and analyzing ambiguous cases.
Leverage Interprompt and Intermodel Agreement	Compare multiple prompts and models to improve consistency and highlight ambiguities.	Reduces risks from biases and/or inaccuracies in any single LLM or prompt.	Identify stable classifications by comparing different prompt structures and models, targeting the human-validation step.
Prioritize Reproducibility and Transparency	Closed models may perform slightly better, but adopting open-source options offers transparency, reproducibility, and accessibility.	Promotes research integrity, long-term usability, and accessibility.	Favor open-source and smaller models, unless accuracy trade-offs are particularly critical.

in a long document often covering multiple themes, which introduces noise into the classification task. This feature of our work also led to mechanical limitations, as LLMs have strict maximum context windows and will truncate text over those limits. There are also, apparently, computational limitations stemming from our classification task, with accuracy diminishing as the prompts became longer in some Llama3.1 and Gemma2 tests (even though they were within the token limit). Yet, precisely because our classification task was noisy, the acceptable levels of accuracy presented in Table 3 suggest that LLMs can be incorporated into these types of classification tasks.

On the other hand, our test case was straightforward for the LLMs because it involved English-language newspapers, which are similar to a large portion of the training data used for many LLMs. In fact, the articles we are classifying may be in the actual training data for the LLMs we tested (though, of course, our classifications of those articles are not). We did not investigate whether our corpus was in fact part of the training data of the models we used, but it should be possible to do so when using open-source models, which is another advantage of such models. The hypothesized alignment between our data and the LLMs' training data may have been one reason we achieved a high F1 score in the baseline test without a definition of our classification concept. It is almost certain that classification tasks based on data or concepts further from the type of training data used in these LLMs would lead to worse, and perhaps unacceptably low, accuracy metrics. The scientific community is increasingly seeking to address this limitation. Qualitative researchers will, of course, need to weigh this limitation against their particular data and classification task, and likely use detailed definitions, fine-tuning, or specialized models for data that is further afield from the standard training data in existing LLMs. Future research may also examine how LLMs can be used for different kinds of interpretive text analysis tasks, such as whether an LLM adopting varying political personas would label vignettes about inequality as being normatively good or bad (e.g., see related work by Kim and Lee 2023; Kozlowksi, Kwon and Evans 2024).

Our final limitation is that we did not engage in extensive fine-tuning tests with all the models employed in our main zero-shot and few-shot tests. Fine-tuning can potentially enhance the performance of LLMs by adapting them more specifically to the nuances of the task and data at hand (Alizadeh et al. 2024). For instance, the accuracy levels in our tests did not exceed those of supervised machine learning (e.g., Nelson et al. 2018), which suggests that supervised machine learning methods, including especially more recent transformer methods, may be a better option if the goal is to strictly apply a pre-determined classification task with the highest

accuracy possible (Bonikowski, Luo, and Stuhler 2022). We therefore experimented with fine-tuning one of the open-source LLMs, Gemma2, as discussed further in Appendix F in the online supplement. We found that using up to 1000 documents for fine-tuning did not result in higher accuracy rates compared to our main tests in Table 3, and that fine-tuning with fewer documents led to substantially worse accuracy, with accuracy peaking at around 600 documents. In a separate fine-tuning analysis using BERT models, we came to the same conclusion. These results confirm that fine-tuning is not only computationally expensive (see Appendix F in the online supplement) but also can be unpredictable; indeed, it is not yet clear which type of fine-tuning is best for the types of tasks and data we are examining. Future research should continue to assess when, and how much, fine-tuning might be needed for different kinds of models (see, e.g., Chae and Davidson 2025 for proposed guidelines).

Conclusion

Despite the fact that the field of generative LLMs and their use in sociology is still in its early moments and presents many challenges, our findings suggest that these models can open new frontiers in qualitative research methodology. They provide a productive blend of automation and nuanced understanding, at least for contemporary English-language text, enabling researchers to analyze vast amounts of text with scalability, flexibility, and efficiency in an interactive way. These advances encourage us both to rethink traditional qualitative inductive and/or iterative coding practices and to integrate them into the workflow of classification using LLMs. As the research community continues to refine and improve LLM capabilities, we are poised to enter an era of qualitative analysis that embraces both the depth of human insight and the breadth of machine learning, without having to leave the comfortable world of natural language behind.

Acknowledgments

We are grateful for funding to support this research from the Russell Sage Foundation Visiting Scholars program and grant #83-13-05 and from the PSC-CUNY Research Award #64715-52. We are also grateful for feedback on an earlier version of this paper from participants at the Generative Artificial Intelligence and Sociology workshop at Yale University. Finally, we thank the special issue editors and reviewers for their careful engagement with our work. Their feedback helped clarify and refine our framework and argument. At times, their input felt closer to collaboration than review, in a very productive way.

Declaration of Conflicting Interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The authors disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the PSC-CUNY Research Award, Russell Sage Foundation (grant number 64715-52, 83-13-05).

ORCID iDs

Nga Than https://orcid.org/0000-0002-6845-6253
Tina Law https://orcid.org/0000-0001-7631-6763
Laura K. Nelson https://orcid.org/0000-0001-8948-300X
Leslie McCall https://orcid.org/0000-0002-7700-3969

Data and Code Availability Statement

The full text of the news articles is under copyright. The news article metadata (identifying the original news articles), all generated output, and all replication and reproduction materials are available at: https://github.com/lknelson/future-of-coding-revisited#

Supplemental Material

Supplemental material for this article is available online.

Notes

1. Some examples include tweets and Facebook posts (Chae and Davidson 2025), summaries of documents or short survey responses (Rytting et al. 2023), or sentence-level chunks of longer documents (Do, Ollion, and Shen 2022).
2. Ollama relies on llama.cpp to implement LLMs. Depending on the way the user imports models, Ollama uses quantization to make the models efficient enough to operate without specialized hardware. There has not been enough testing yet to determine whether quantization impacts the performance of the model. In the meantime, anyone can implement these models on more-or-less standard hardware, which helps to make the process more reproducible compared to a situation in which researchers are downloading and implementing models themselves. With an eye toward maximum reproducibility, we thus opted for using the Ollama platform, despite the use of quantization that may impact model performance.

3. We note here that GPT4 is also costly by social science standards, at least at this point in time, even though it is heavily subsidized by OpenAI. Each test we ran with GPT4 cost between $50 and $70 depending on sample sizes. Fine-tuning and other alterations would cost even more (which is why we perform them using open-source models, as discussed below). Open-source models are free to use, if you have access to the relevant hardware and electricity. Cost disparities between GPT and open-source models may change in the future, however.
4. At an earlier stage of experimentation when we were hindered by token limits, we attempted to conduct paragraph-level analyses. However, we found them to be *a priori* infeasible if we were to maintain consistency with the original qualitative coding task, which assessed content holistically over the full span of an article.
5. We use F1 scores to measure accuracy using the following terms and equations: the weighted_average_precision = average of precision scores multiplied by the proportion of total rows that are true positives for each category; the weighted_average_recall = average of recall scores multiplied by the proportion of total rows that are true positives for each category; weighted average total F1 = (2 * weighted_average_precision * weighted_average_recall)/(weighted_average_precision + weighted_average_recall). In the prior replication, weighted F1 = 0.74–0.86 for two separate binary schemas across 25 randomized training/test sets (Nelson et al. 2018, Table 1, also reproduced in Appendix B in the online supplement). Not all tests in this paper are strictly comparable to the tests in Nelson et al. (2018). The most comparable F1 scores from that article to the F1 scores we report in this paper are in their Table 1, Column 9 (reproduced in Appendix B) for the first two schemas (relevant vs. irrelevant, median F1=0.83; inequality vs. not inequality, median F1=0.78) in the first row for SML. These are scores obtained for the full corpus, not only for the subset of relevant articles, which is the second step in the sequence of tests followed in this paper.
6. New applications to reduce LLMs' sensitivity to natural language prompts are being developed, such as DSPy, but these applications tend to substitute programming for natural language interfaces, which limits their appeal for qualitative researchers (Khattab et al. 2023).
7. The temperature parameter is designed for users to control the randomness of text output from LLMs, ranging from 0 (more deterministic) to 1 (less deterministic). However, it is currently unclear whether setting the temperature parameter to 0 results in completely deterministic output, as that depends on the specific engineering choices of each model, which are not always clearly documented.
8. Some might argue that our meta statement includes our definition of inequality. But note that our tests using Llama3 with no definition and without a meta statement resulted in similar weighted F1 scores, with the highest score being 0.76 (see Appendix C, Table C1, columns 1–3).

9. A potential additional reason could be that our tests could not take advantage of memory outside of the instruction window, which may become available in future versions of the models.

References

"A Modest Proposal: Public Policies that Perform." *U.S. News & World Report*, August 10, 1987.

Alizadeh, Meysam, Maël Kubli, Zeynab Samei, Shirin Dehghani, Mohammadmasiha Zahedivafa, Juan Diego Bermeo, Maria Korobeynikova, and Fabrizio Gilardi. 2024. "Open-Source LLMs for Text Annotation: A Practical Guide for Model Setting and Fine-Tuning." *arXiv*. http://arxiv.org/abs/2307.02179.

Bail, Christopher A. 2024. "Can Generative AI Improve Social Science?" *Proceedings of the National Academy of Sciences* 121(21):e2314021121.

Bender, Emily M., Timnit Gebru, Angelina McMillan-Major, and Shmargaret 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, pp. 610-623.

Bengio, Yoshua, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. "A Neural Probabilistic Language Model." *The Journal of Machine Learning Research* 3:1137-55. via ACM Digital Library.

Boelaert, Julien, Samuel Coavoux, Etienne Ollion, Ivaylo D. Petev, and Patrick Präg. 2025. "Machine Bias. Generative Large Language Models Have a View of Their Own." *Sociological Methods & Research*. doi:10.1177/00491241251330582

Bommasani, Rishi, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, et al. 2021. "On the Opportunities and Risks of Foundation Models." *arXiv*. doi:10.48550/ARXIV.2108.07258

Bonikowski, Bart, Yuchen Luo, and Oscar Stuhler. 2022. "Politics as Usual? Measuring Populism, Nationalism, and Authoritarianism in U.S. Presidential Campaigns (1952–2020) with Neural Language Models." *Sociological Methods & Research* 51(4):1721-87.

Brandt, Philipp and Stefan Timmermans. 2021. "Abductive Logic of Inquiry for Quantitative Research in the Digital age." *Sociological Science* 8:191-210.

Brown, T., B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, and D. Amodei. 2020. "Language Models are few-Shot Learners. Advances in Neural Information Processing Systems." *Conference on Neural Information Processing Systems* 33:1877-901.

Chae, Youngjn and Thomas Davidson. 2025. "Large Language models for Text Classification: From Zero-Shot Learning to Instruction-Tuning." *Sociological Methods & Research*. doi:10.1177/00491241251325

Chong, Dennis and James N. Druckman. 2011. "Identifying Frames in Political News." Pp. 238-267 in *Sourcebook for Political Communication Research: Methods, Measures, and Analytical Techniques*, edited by E. P. Bucy and R. L. Holbert. New York: Routledge.

Davidson, Thomas. 2024. "Start Generating: harnessing Generative Artificial Intelligence for Sociological Research." *Socius: Sociological Research for a Dynamic World* 10(January):23780231241259651.

Deterding, Nicole M. and Mary C. Waters. 2021. "Flexible Coding of In-Depth Interviews: a Twenty-First-Century Approach." *Sociological Methods & Research* 50(2):708-39.

Do, S., É. Ollion, and R. Shen. 2022. "The Augmented Social Scientist: Using Sequential Transfer Learning to Annotate Millions of Texts with Human-Level Accuracy." *Sociological Methods & Research* 53(3):1167-200.

Dubey, Abhimanyu, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, et al. 2024. "The Llama 3 Herd of Models." *arXiv*. http://arxiv.org/abs/2407.21783.

Ferree, Myra Marx, William Anthony Gamson, Jurgen Gerhards, and Dieter Rucht. 2002. *Shaping Abortion Discourse: Democracy and the Public Sphere in Germany and the United States*. NY: Cambridge University Press.

Fleiss, J. L. 1971. "Measuring Nominal Scale Agreement among Many Raters." *Psychological Bulletin* 76(5):378-82.

Foster, Jacob G. and James A. Evans. 2024. "Algorithmic Abduction: robots for Alien Reading." *Critical Inquiry* 50(3):375-401. doi:10.1086/728933

Gamson, William. 1992. *Talking Politics*. New York: Cambridge University Press.

Gilardi, Fabrizio, Meysam Alizadeh, and Mael Kubli. 2023. "ChatGPT Outperforms Crowd Workers for Text-Annotation Tasks." *PNAS* 120(30):e2305016120.

Gilens, Martin. 1999. *Why Americans Hate Welfare: Race, Media, and the Politics of Antipoverty Policy*. Chicago: University of Chicago Press.

Grimmer, Justin, Margaret E. Roberts, and Brandon M. Stewart. 2022. *Text as Data: A new Framework for Machine Learning and the Social Sciences*. Princeton: Princeton University Press.

Griswold, Wendy. 1987. "A Methodological Framework for the Sociology of Culture." *Sociological Methodology* 17:1-35. doi:10.2307/271027

Ibrahim, Elida I. and Andrea Voyer. 2024. "The Augmented Qualitative Researcher: Using Generative AI in Qualitative Text Analysis." SocArXiv. January 24. doi:10.31235/osf.io/gkc8w

Jensen, Jeffrey L., Daniel Karell, Cole Tanigawa-Lau, Nizar Habash, Mai Oudah, and Dhia Fairus Shofia Fani. 2022. "Language Models in Sociological Research: an Application to Classifying Large Administrative Data and Measuring Religiosity." *Sociological Methodology* 52(1):30-52.

Karell, Daniel, Matthew Shu, Keitaro Okura, and Thomas Davidson. 2024. Artificial Intelligence Summaries of Historical Events Improve Knowledge Compared to Human-Written Summaries. *SocArXiv*. September 12. doi:10.31235/osf.io/3gsqw

Kesari, Aniket, Jae Yeon Kim, Sono Shah, Taylor Brown, Tiago Ventura, and Tina Law. 2023. *Training Computational Social Science Ph.D. Students for Academic and Non-Academic Careers.* PS: Political Science & Politics.

Khattab, Omar, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. 2023. "DSPy: Compiling declarative language model calls into self-improving pipelines." Preprint. https://arxiv.org/abs/2310.03714.

Kim, Junsol and Byungkyu Lee. 2023. "Ai-augmented surveys: Leveraging large language models for opinion prediction in nationally representative surveys." *arXiv preprint arXiv:2305.09620*.

Knight, Carly. 2022. "When Corporations are People: agent Talk and the Development of Organizational Actorhood, 1890–1934." *Sociological Methods and Research* 51(4):1634-80.

Kojima, T., S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa. 2022. "Large Language Models are Zero-Shot Reasoners." Pp. 22199–213 in *Proceedings of the 36th International Neural Information Processing Systems*.

Kozlowski, Austin C., Hyunku Kwon, and James A. Evans. 2024. "In silico sociology: forecasting COVID-19 polarization with large language models." *arXiv preprint arXiv:2407.11190*.

Krippendorff, Klaus H. 2012. *Content Analysis: An Introduction to Its Methodology*. Los Angeles; London: Sage Publications, Inc.

Law, Tina. 2022. Parsing the Language of Rebellion: Impacts of the 1960s Black-Led Urban Uprisings on American Political and Legal Discourse. Doctoral dissertation, Northwestern University.

Litterer, Benjamin, David Jurgens, and Dallas Card. 2024. "Mapping the podcast ecosystem with the structured podcast research corpus." https://arxiv.org/pdf/2411.07892.

Madsen, Anders Koed, Anders Kristian Munk, and Johan Irving Soltoft. 2023. "Friction by Machine: How to slow down reasoning with computational methods." Ethnographic Praxis in Industry Conference Proceedings. https://www.epicpeople.org/friction-by-machine-and-computational-methods/.

Manning, Christoper and Henrich Schutze. 1999. *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press.

McCall, Leslie. 2013. *The Undeserving Rich: American Beliefs About Inequality, Opportunity, and Redistribution*. Cambridge: Cambridge University Press.

Mesnard, Thomas, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, et al. 2024. "Gemma: Open Models Based on Gemini Research and Technology." *arXiv*. http://arxiv.org/abs/2403.08295.

Nelson, Laura K. 2020. "Computational Grounded Theory: a Methodological Framework." *Sociological Methods & Research* 49(1):3-42.

Nelson, Laura K. 2021. "Cycles of Conflict, a Century of Continuity: the Impact of Persistent Place-Based Political Logics on Social Movement Strategy." *American Journal of Sociology* 127(1):1-59.

Nelson, Laura K., Derek Burk, Marcel Knudsen, and Leslie McCall. 2018. "The Future of Coding: a Comparison of Hand-Coding and Three Types of Computer-Assisted Text Analysis Methods." *Sociological Methods & Research* 50(1):202-37. doi:10.1177/0049124118769114

Nelson, Laura K. and Brayden G King. 2020. "The Meaning of Action: linking Goal Orientations, Tactics, and Strategies in the Environmental Movement." *Mobilization: An International Quarterly* 25(3):315-38.

O'Connor, Cliodhna and Helene Joffe. 2020. "Intercoder Reliability in Qualitative Research: debates and Practical Guidelines." *International Journal of Qualitative Methods* 19. doi:10.1177/1609406919899220

Ollion, Étienne, Rubing Shen, Ana Macanovic, and Arnault Chatelain. 2024. "The Dangers of Using Proprietary LLMs for Research." *Nature Machine Intelligence* 6(1):4-5.

OpenAI. 2023. "GPT-4 Technical Reports." Preprint. https://cdn.openai.com/papers/gpt-4.pdf.

Pardo-Guerra, Juan Pablo and Prithviraj Pahwa. 2022. "The Extended Computational Case Method: a Framework for Research Design." *Sociological Methods & Research* 51(4):1826-67.

Reiss, Michael. 2023. "Testing the reliability of ChatGPT for annotation and classification: A cautionary remark." Preprint. https://arxiv.org/abs/2304.11085.

Rytting, Christopher Michael, Taylor Sorensen, Lisa Argyle, Ethan Busby, Nancy Fulda, Joshua Gubler, and David Wingate. "Towards coding social science datasets with language models." arXiv preprint arXiv:2306.02177 (2023).

Spirling, Arthur. 2023. "Why Open-Source Generative AI Models Are an Ethical Way Forward for Science." *Nature* 616(7957):413-413.

Stoltz, Dustin S. and Marshall A. Taylor. 2019. "Concept Mover's Distance: Measuring Concept Engagement via Word Embeddings in Texts." *Journal of Computational Social Science* 2:293-313.

Törnberg, Petter. 2024. "Best Practices for Text Annotation with Large Language Models." *Sociologica* 18(2): 67-85. doi:10.6092/issn.1971-8853/19461

Voyer, Andrea, Zachary D. Kline, Madison Danton, and Tatiana Volkova. 2022. "From Strange to Normal: computational Approaches to Examining Immigrant Incorporation Through Shifts in the Mainstream." *Sociological Methods and Research* 51(4):1540-79.

Wagner-Pacifici, Robin, John Mohr, and Ronald Breiger. 2015. "Ontologies, Methodologies, and new Uses of Big Data in the Social and Cultural Sciences." *Big Data & Society* July-Sept:1-11.

Wu, Yu, S. Prabhumoye, S. Y. Min, Y. Bisk, R. Salakhutdinov, A. Azaria, T. Mitchell, and Y. Li. 2023. "SPRING: Studying the paper and reasoning to play games." Preprint. https://arxiv.org/pdf/2305.15486.pdf.

Yang, Chengrun, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. 2023. "Large Language Models as Optimizers." Preprint. https://arxiv.org/abs/2309.03409.

Zhang, T., F. Ladhak, E. Durmus, P. Liang, K. McKeown, and T. B. Hashimoto. 2024. "Benchmarking Large Language Models for News Summarization." *Transactions of the Association for Computational Linguistics* 12:39-57.

Ziems, C., W. Held, O. Shaikh, J. Chen, Z. Zhang, and D. Yang. 2024. "Can Large Language Models Transform Computational Social Science?" *Computational Linguistics* 50(1): 237-91.

Author Biographies

Nga Than is a Research Associate at Stone Center on Socio-Economic Inequality, The Graduate Center, CUNY. Her research has utilized computational social science methods to study public opinion on immigration. She received her Ph.D. in sociology from the Graduate Center, City University of New York. She has published in *Ethnic and Racial Studies*, *Frontiers in Sociology*, *Journal of Community & Applied Social Psychology*, and *Communication Research and Practice*.

Leanne Fan is a Sociology student at The Graduate Center, City University of New York. Her research interests focus on the conditions that lead to the development of multiracial working-class political coalitions, exploring the intersections of race, class, and social movements through a variety of methodological approaches.

Tina Law is an Assistant Professor of Sociology at the University of California, Davis. She studies inequality, race and ethnicity, and democracy and specializes in computational and quantitative methods. She uses computational and quantitative methods to understand the social and political experiences of racially minoritized and low-income residents living in U.S. cities and explores ways to adapt computational methods for sociological research.

Laura K. Nelson is an associate professor of sociology at the University of British Columbia, where she also directs the Centre for Computational Social Science. She uses computational methods to study social movements, gender, culture, and institutions, and to advance qualitative computational text analysis methods. She has published in outlets such as *American Journal of Sociology, American Sociological Review,* and *Gender and Society,* among others.

Leslie McCall is Presidential Professor of Sociology and Political Science and Associate Director of the Stone Center on Socio-Economic Inequality at The Graduate Center, City University of New York. Using a variety of methodological approaches, she studies public opinion and media coverage about inequality, opportunity, and related economic and policy issues; trends in actual earnings and family income inequality; and patterns of intersectional inequality.