Dunivin *EPJ Data Science* (2025) 14:28
https://doi.org/10.1140/epjds/s13688-025-00548-8

EPJ Data Science
a SpringerOpen Journal

EPJ.org

RESEARCH Open Access

# Scaling hermeneutics: a guide to qualitative coding with LLMs for reflexive content analysis

Zackary Okun Dunivin[1*]

*Correspondence:
zdunivin@ucdavis.edu
[1]Department of Communication, University of California, Davis, Davis, California, USA

**Abstract**

Qualitative coding, or content analysis, is more than just labeling text: it is a reflexive interpretive practice that shapes research questions, refines theoretical insights, and illuminates subtle social dynamics. As large language models (LLMs) become increasingly adept at nuanced language tasks, questions arise about whether—and how—they can assist in large-scale coding without eroding the interpretive depth that distinguishes qualitative analysis from traditional machine learning and other quantitative approaches to natural language processing. In this paper, we present a hybrid approach that preserves hermeneutic value while incorporating LLMs to scale the application of codes to large data sets that are impractical for manual coding. Our workflow retains the traditional cycle of codebook development and refinement, adding an iterative step to adapt definitions for machine comprehension, before ultimately replacing manual with automated text categorization. We demonstrate how to rewrite code descriptions for LLM-interpretation, as well as how structured prompts and prompting the model to explain its coding decisions (chain-of-thought) can substantially improve fidelity. Empirically, our case study of socio-historical codes highlights the promise of frontier AI language models to reliably interpret paragraph-long passages representative of a humanistic study. Throughout, we emphasize ethical and practical considerations, preserving space for critical reflection, and the ongoing need for human researchers' interpretive leadership. These strategies can guide both traditional and computational scholars aiming to harness automation effectively and responsibly—maintaining the creative, reflexive rigor of qualitative coding while capitalizing on the efficiency afforded by LLMs.

**Keywords:** LLM; Text categorization; Content analysis; Qualitative coding; Digital humanities

## 1 Introduction

Text categorization, commonly referred to as content analysis and qualitative coding in the social sciences, plays an important role in scholarly research and industrial applications. This process traditionally relies on human expertise to interpret the nuanced and often complex meanings embedded in texts [1, 2]. The challenge lies in capturing the multi-faceted nature of meaning and translating real-world complexity into discrete categories—a task that even skilled researchers must approach with care. These challenges have histor-



Springer

ically made qualitative coding resistant to automation, with machine learning approaches struggling to match the interpretive depth required [3], despite robust attempts [4, 5].

Recent developments in artificial intelligence, notably the advent of transformers with billions of parameters known as large language models (LLMs), have begun to challenge this notion. These models demonstrate increasing capabilities in knowledge, interpretation, reasoning, and creativity expressed in natural language, approaching or even surpassing human performance [6–9]. The processing speed of artificial intelligence opens up the possibility of categorizing vast quantities of text, far exceeding the limitations of human coding teams restricted to smaller samples.

However, scaling alone is not enough. How can researchers integrate these tools while preserving the interpretive depth and rigor that distinguish qualitative coding from other approaches? This challenge calls for a workflow that combines the efficiency of automation with the reflexive, interpretive practices central to qualitative research. By retaining the reflexive process of iterative codebook development and rigorously validating machine-applied codes against a human-derived gold standard, researchers can balance hermeneutic depth with automation's scalability. These principles provide the foundation for the hybrid approach detailed throughout this paper.

Qualitative coding is not merely the act of labeling text; it is an iterative process of interpretation and framework development. Rooted in traditions like grounded theory, this process involves successive readings of text to refine theoretical insights and develop a qualitative codebook—a structured guide comprising categories, descriptions, and coding instructions [4, 10]. The approach advanced here builds on this foundation while introducing automation selectively. Specifically, LLMs are incorporated only after human researchers have developed the codebook. Subsequently, the codebook is adapted to enhance machine comprehension, ensuring that automated coding aligns with human standards of reliability.

This hybrid workflow, visualized in Fig. 1, retains the iterative refinement employed by qualitative researchers while introducing a second layer of hermeneutic engagement: tailoring code descriptions to maximize LLM performance. Moreover, by measuring machine outputs against human-derived benchmarks—using intercoder reliability to measure coherence with a gold standard—this approach ensures quality and preserves the interpretive rigor of qualitative research.

Further, noting the explosive growth of LLMs generally, and LLM-assisted text categorization more specifically, we contribute evidence that LLMs are capable of human-quality interpretations of complex concepts expressed in paragraph-long passages of text. Our case study analyzes news media accounts referencing W.E.B. Du Bois to examine how

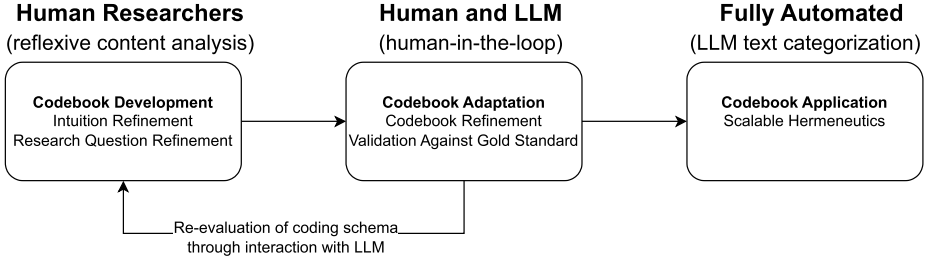


**Figure 1** Hybrid automated workflow for LLM-assisted qualitative coding

the man's legacy has been invoked to understand the interplay of race, scholarship, and activism. Through this case study in LLM-assisted qualitative coding, we add to the growing body of work that builds confidence in the rigor of LLM-based text categorization [9, 11–13], a field that will expand as these models continue to evolve. Our report emphasizes the redesign of codebooks specifically for LLMs. We demonstrate how the structure of prompts, the specific requests made to the generative model for categorizing passages, significantly impacts coding fidelity. Even as these models continue to rapidly improve, we expect most of the principles of prompt design we report will remain useful and informative as methodologists explore new models and empiricists automate their coding workflows. Our results are presented through narratives detailing our approach and highlighting potential challenges and demonstrated by LLM-generated analyses compared to a human-derived gold standard. A summary of best practices for LLM-assisted content analysis is also presented in Table 4 for quick reference.

Key empirical findings of our study include:

- GPT-4 exhibits human-equivalent performance with zero-shot prompts. Eight of nine tasks exceed the 0.6 threshold for substantial agreement using Cohen's $\kappa$. Three of nine tasks exceed the 0.75 threshold for excellent agreement;
- GPT-3.5, when given the same prompts, has an average intercoder reliability of 0.34 across all codes;
- Codebooks designed for human coders need reworking for LLM application, requiring iterative manual testing to refine phrasing and improve model comprehension;
- Agreement improves when the LLM provides rationale for code assignments: $\mu(\kappa) = 0.68$ vs. $\mu(\kappa) = 0.59$;
- Agreement improves when presenting each code as a separate prompt, rather than the codebook as a whole: $\mu(\kappa) = 0.68$ vs. $\mu(\kappa) = 0.60$.

### 1.1 Automating content analysis: past and present

Prior work on automating content analysis entailed training machine learning models on large quantities of text. Supervised models, typically some form of linear regression, learn to associate text features with user-specified categories [14]. This process captures half the traditional human-coded process by using human-derived codes and examples, but fails to leverage abstract code descriptions found in a codebook, as well as requiring large quantities of human-annotated data. Unsupervised models, such as LDA [15, 16] or BERTopic [17], develop their own categorizations from unlabeled training sets. This process does not require time-intensive labeling, but rarely captures the specific categories that the researcher intends to target.

The latest generation of LLMs (e.g., GPT [18], Llama [19], Mistral [20], Claude [21]) differ notably from previous machine learning models in that they can perform new tasks specified through natural language prompts. A user can specify a task that the model was not trained on, give few (single digit) or no examples, and the model will return output conforming to the specifications. Demonstrated successes include computer code generation [22], creative writing [23], and quantitative reasoning [24]. We are only beginning to understand and expand upon the limitations of these models. By converting natural language requests into highly intelligent output across vast and indeterminate domains, LLMs lower the technical barriers to machine learning by making its application more naturalistic and eliminating the need for large training data. Beyond this, LLM's capacities in

many domains far exceed the specialized machine learning models that preceded them, suggesting that for many applications, including scholarly inquiry, artificial intelligence is overwhelmingly more accessible and capable in 2025 than it was just two years prior.

Methodological examinations of content analysis with LLMs are encouraging. Xiao et al. [11] demonstrate moderate success, Cohen's $\kappa = 0.61$ and $\kappa = 0.38$, in two linguistic tasks using GPT-3. Gilardi, Azedeh, and Kubil [9] find that GPT-3.5 outperforms crowd workers (nonexperts) in identifying political speech, performing comparably to trained experts. Chew et al. [12] report high success on many of 19 tasks across three datasets, and results that are indistinguishable from random for others. It is difficult to evaluate their results due to the choice of Gwet's AC1, which is effectively biased toward agreement on negative codings rather than positive, whereas standard measures of intercoder reliability do the opposite [25]. A survey of 20 empirical pieces reports "mixed-results" of using GPT-3 to automate "text annotation," a term that ties their framing to "data annotation," labeling data for use in machine learning [26], rather than content analysis in the tradition of grounded theory [10]. As in this study, several earlier methodological inquiries examine the capabilities of GPT-4 (the frontier model for all of 2023 and much of 2024) as a text classifier. Yu et al. [27] find that GPT-4 outperforms GPT-3.5 for discourse analysis, approaching expert performance in identifying apologies. Törnberg [13], paralleling Gilardi, Azedeh, and Kubil [9], reports that GPT-4 outperforms both experts and crowd workers in identifying political signals.

While 2023-24 saw an abundance of articles demonstrating successes and limitations of LLMs for qualitative coding in various contexts, there are comparatively few that are intended to serve as a guide for practitioners. Chew et al. [12] give an account of adapting codebooks for LLMs, which is communicated with great detail and clarity, but the authors fail to demonstrate that LLM-assisted content analysis is effective, due to their choice of evaluation metric. Halternman and Keith [28] provide a less detailed account of codebook adaptation than Chew et al., but give excellent examples of structured prompt design and suggest many useful tests to identify flaws in prompt structure. Törnberg [29] gives a thorough review of best practices, but does so at a high level of abstraction with minimal evidence, and does not demonstrate concrete examples of codebook adaptation, prompt design, or codebook evaluation as we do here.

In our estimation, while these guides offer useful insights for practitioners transitioning from traditional NLP methods to LLM-assisted workflows, they fail to fully address how these tools can be integrated into the tradition of qualitative coding. By contrast, our approach emphasizes how LLM-assisted coding retains the core interpretive practices central to qualitative research while introducing the transformative advantage of scalability. Unlike conventional NLP methods, which typically neglect the hermeneutic foundations of qualitative analysis both in process and product, this approach respects and preserves those principles. Additionally, this article is explicitly written to appeal to traditional qualitative researchers, providing a practical and methodologically grounded guide that bridges established practices with the opportunities afforded by LLMs. The following section expands on this argument, illustrating how this hybrid approach aligns with the values and objectives of qualitative coding while enabling researchers to scale their analyses effectively.

## 1.2 Why transition from manual to LLM-assisted qualitative coding?

The rigor and depth of traditional qualitative coding rest on decades of methodological and theoretical scholarship. Traditionally, researchers immerse themselves deeply in their data, performing analytical and interpretive tasks beyond mere categorization. Coding is not merely a mechanical task of categorization; it is an inductive, reflexive process through which researchers uncover patterns and generate insights [2, 30]. This interplay between data, observation, and theory enables the scholar to refine their understanding of the research questions and theoretical frameworks, making coding a central act of qualitative inquiry rather than a trivial task [31, 32]. Therefore, the transition from manual to LLM-assisted coding must be approached with a clear rationale and methodological rigor.

### *1.2.1 Preserving hermeneutic value*

A central critique of automating qualitative coding is the potential erosion of the hermeneutic process—the interpretive interaction between researchers and their data [3, 4]. Traditional qualitative methodologies explicitly position researchers as analytical instruments, whose subjectivities and reflexivity are integral rather than detrimental [31, 33]. Iterative human engagement enables researchers to continuously refine categories and theoretical insights, generating nuanced hypotheses rooted firmly in the empirical material [34].

Yet, the incorporation of LLM-assisted coding need not imply the abandonment of this hermeneutic richness. In our approach, automation functions primarily to augment, rather than replace, human interpretive work, enabling researchers to extend rigorous qualitative analysis to larger data sets. Human researchers retain their central analytical roles through inductive, collaborative refinement of codebooks before the introduction of an LLM. Furthermore, adapting code descriptions specifically for LLM interpretation adds an additional layer of hermeneutic engagement. This step compels researchers to articulate their analytical categories more explicitly, responding critically and iteratively to model outputs. Thus, the iterative adaptation of the coding schema maintains—and potentially enhances—the theory-grounded and reflexive dimensions integral to qualitative inquiry, even as the practical task of coding is operationalized through automation.

### *1.2.2 Balancing losses and gains*

Transitioning to LLM-assisted coding inevitably involves trade-offs. A notable loss is the experiential learning and data intimacy gained through manual application of a codebook [35]. For students and novice researchers, this immersion is invaluable as it builds both valuable practical skills and theoretical insights [32]. However, experiential engagement can be preserved by incorporating manual coding exercises within the training pipeline, such as actively adapting codebooks for machine interpretation or manually verifying LLM outputs. Such hybrid methods offer avenues for maintaining deep engagement while harnessing automation's efficiency.

Another significant absence in our hybrid workflow is the process wherein human coders negotiate and resolve interpretive disagreements to achieve high-quality labeling [36, 37]. These collaborative discussions improve coding fidelity by reconciling divergent interpretations. LLM-assisted workflows, lacking this human negotiation, might omit opportunities to address ambiguous data through collaborative resolution. Although these

collaborative steps might be partially replicated by flagging challenging passages for human review or utilizing multi-agent systems involving multiple LLMs, such solutions exceed the scope of this study and require further research into multi-agent collaboration [38].

Despite these trade-offs, the substantial improvements in scalability and efficiency provided by LLM-assisted coding strongly compensate for these limitations. Automation allows researchers to analyze extensive datasets otherwise impractical for manual coding, significantly enhancing statistical power and uncovering patterns potentially obscured in smaller samples. As statistical power increases, the effects of noise and complexity that obscure genuine patterns diminish [39], improving the robustness and generalizability of findings.

Moreover, compared to other computational text annotation methods, LLM-assisted coding uniquely supports rigorous hermeneutic engagement [4]. Even without adopting a full grounded theory approach, iterative refinement of codebooks for LLM comprehension introduces additional analytical rigor. Adapting codebooks for automated interpretation necessitates precise definitions and instructions, thereby deepening researchers' data insights and enhancing coding precision, aligning closely with traditional qualitative research objectives while benefiting from automation's efficiency.

#### *1.2.3 A hybrid paradigm for qualitative analysis*

Rather than positioning LLM-assisted coding as a replacement for manual methods, it is more productive to conceive it as complementary [cf. [40]]. Leveraging the strengths of both human and machine renders a hybrid paradigm balancing efficiency with hermeneutic depth. This approach retains the core principles of qualitative inquiry—reflexivity, subjectivity, and theory-grounded rigor—while extending the analytic scope to accommodate increasingly complex and voluminous datasets.

As qualitative researchers—both those experienced and inexperienced with computational methods—navigate these emerging opportunities, preserving the integrity of traditional methodologies is essential. The values and experiences of traditional qualitative researchers will be especially valuable in the further development LLM-assisted content analysis practices. This is especially important in light of the past decades of NLP research, which has been dominated by computational social scientists without grounding in traditional methodologies.

### 1.3 Case study: W.E.B. Du Bois's characterization in news media

In order to present a realistic challenge of using an LLM to do qualitative coding, we make a case study of our own work. We adapted a codebook written by the authors to understand how the scholar and activist W.E.B. Du Bois has been characterized in news media over time. The codebook is composed of nine codes in three categories. Due to multiple layers of agency (who is doing what) and voice (who is saying what), the tasks are difficult even for human interpreters. Applying the codes is also complicated because it can be difficult to differentiate Du Bois's scholarship from his political activism, as Du Bois's theoretical contributions have profound implications for understanding race and the social-historical position of Black persons in the United States and beyond, making them powerful activist tools. We are particularly interested in understanding how different facets of Du Bois's activities contributed to his canonization in the public imagination as the preeminent figure for understanding Black political struggle. Table 1 gives the codes in brief. Complete

**Table 1** Categories and descriptions for 9 codes

| | |
|---|---|
| Characterization of Du Bois | |
| *Scholar* | Describes Du Bois as a scholar or intellectual. |
| *Activist* | Refers to Du Bois's political or social activism. |
| General Themes | |
| *Monumental Memorialization* | Refers to an enduring cultural object named after Du Bois. |
| *Mention of Scholarly Work* | Mentions or quotes specific academic works by Du Bois. |
| *Social/Political Advocacy* | Mentions or implies social or political activism, advocacy, or critique. |
| Canonization Processes | |
| *Coalition Building* | Refers to Du Bois's activities with activist or academic organizations. |
| *Out of the Mouth of Academics* | Describes an academic organization engaging with Du Bois's legacy. |
| *Out of the Mouth of Activists* | Describes an activist organization engaging with Du Bois's legacy. |
| *Collective Synecdoche* | Mentions Du Bois alongside other figures in order to represent some facet of a culture, era, or ideology. |

examples of the original human and modified-for-GPT codebook are included in the Appendix.

The training and test data for our study were random samples of passages from New York Times articles (1970–2023) that mention "W.E.B. Du Bois." 232 passages were automatically extracted as concurrent paragraphs containing "Du Bois." The average number of words was 94 ($\sigma$ = 70), and the average number of sentences was 3.75 ($\sigma$ = 2.88). To give a better sense of the size of our passages, this paragraph has 76 words across four sentences.

### *1.3.1 Initial codebook development*

Initially, a qualitative research team developed a codebook intended for human coders. This process involved exploratory readings to define and refine codes. Codes were derived to probe particular substantive hypotheses, some of which preceded our exploratory reading, and others which resulted from it. Codes were repeatedly tested against a random sample of passages, refined through discussions, and adjusted until high intercoder reliability was achieved and theoretical value was clear. Table 3 gives the ultimate human-human intercoder reliability scores for two human coders who led codebook development; any remaining disagreements between these coders were resolved by consensus [37] to form the gold standard data set of 111 passages.

The team began with eight codes. Two, retained in the final codebook, focused on differentiating Du Bois's roles as Scholar and Activist. Two others, Mention of Scholarly Work and Social/Political Activism, were also retained. The four remaining codes were drawn from initially crude hypotheses about canonization processes. In response to coding, subsequent discussions, and parallel development of theory through literature review, these codes were heavily refined, as most changed scope. A code titled Organizations was imprecise and ultimately refined into Coalition Building, relating to how Du Bois built ties to organizations and constituencies. Later, Academic Repute and Activist Repute were added to capture when members of either type of institution engaged with Du Bois's legacy. These three categories emerged in tandem from development of a theory of canonization through literature review, and dissatisfaction with the crudeness of the original code label. Another code, Broadly Listed, was replaced by Collective Synecdoche, which distinguished the use of Du Bois as a symbol by listing him among similar figures. This new code, along with Monumental Memorialization stood in for measurements that an orig-

inal ambiguous code, Prestige, attempted to capture. A final original code, Public Media, was ultimately dropped due to definitional ambiguity and weak theoretical alignment.

Three additional role-specific codes—Professor, Sociologist/Historian, and Founder—were introduced but later abandoned due to insufficient occurrences for reliable coding. Likewise, Living Achievement, meant to capture contemporary recognition of Du Bois, was omitted for lack of robust examples in the corpus.

## 2 Adapting a codebook for an LLM

Having developed a robust qualitative codebook through iterative human-centered refinement, we adapted these descriptions for large language model (LLM) interpretation. We evaluated the LLM's performance against our gold standard set of 111 passages, iteratively revising definitions to resolve ambiguities revealed through the model's interpretations. This adaptation follows Nelson's Computational Grounded Theory paradigm [4], which emphasizes iterative cycles of definition, evaluation, and refinement common to qualitative methods, irrespective of automation [10].

The following paragraphs highlight key insights from this adaptation process. Readers interested in a comprehensive guide to similar approaches may consult Chew et al. [12], who detail their own experiences integrating LLMs into qualitative content analysis workflows.

Table 2 shows the final results of codebook adaptation for LLM-assisted coding, however, the results of model performance are more fully addressed in Sect. 3. The full codebook with the original and final, modified codes may be viewed in the Appendix. In addition to Cohen's $\kappa$, Table 2 shows Gwet's AC1 and F1. The discrepancy between $\kappa$ and AC1 is easily observed for those codes that are infrequent in the dataset, suggesting high performance, when $\kappa$ and F1 better reflect an intuitive understanding of coding fidelity.

**Table 2** GPT-4 performance under the optimal prompt design (chain-of-thought, one prompt per code) for 111 passages on code descriptions from Initial human codebook and Final LLM-adapted codebook

| Code | Codebook Version | Count | | Agreement | Cohen's $\kappa$ | Gwet's AC1 | F1 |
|---|---|---|---|---|---|---|---|
| | | Gold Standard | GPT-4 | | | | |
| Scholar | Initial | 27 | 44 | 0.76 | 0.46 | 0.57 | 0.62 |
| | Final | 27 | 32 | 0.85 | 0.61 | 0.75 | 0.71 |
| Activist | Initial | 23 | 10 | 0.91 | 0.48 | 0.82 | 0.55 |
| | Final | 23 | 24 | 0.94 | 0.81 | 0.91 | 0.85 |
| Monumental Memorialization | Initial | 13 | 13 | 1.00 | 1.00 | 1.00 | 1.00 |
| | Final | — | — | — | — | — | — |
| Mention of Scholarly Work | Initial | 24 | 21 | 0.87 | 0.58 | 0.80 | 0.67 |
| | Final | 24 | 25 | 0.90 | 0.71 | 0.85 | 0.78 |
| Social/Political Advocacy | Initial | 51 | 39 | 0.78 | 0.56 | 0.58 | 0.73 |
| | Final | 51 | 49 | 0.82 | 0.64 | 0.64 | 0.80 |
| Coalition Building | Initial | 9 | 10 | 0.94 | 0.60 | 0.93 | 0.63 |
| | Final | — | — | — | — | — | — |
| Out of the Mouth of Academics | Initial | 30 | 45 | 0.81 | 0.59 | 0.66 | 0.72 |
| | Final | 30 | 28 | 0.86 | 0.63 | 0.77 | 0.72 |
| Out of the Mouth of Activists | Initial | 11 | 21 | 0.80 | 0.21 | 0.74 | 0.31 |
| | Final | 11 | 6 | 0.90 | 0.30 | 0.89 | 0.35 |
| Collective Synecdoche | Initial | 26 | 31 | 0.92 | 0.79 | 0.87 | 0.84 |
| | Final | — | — | — | — | — | — |

Evaluating model performance and prompting strategy are more thoroughly addressed in Sect. 3.

Although our code definitions evolved during their adaptation for LLM hermeneutics, we did not find it necessary to re-evaluate the overall scope or interpretation of any codes. As we discuss in Sect. 1.2.1 and illustrate in Fig. 1, the process of adapting a codebook for LLM-assisted coding presents an opportunity for broader re-evaluation of the coding schema. However, in this case, no such need emerged, and our focus remained on ensuring that each code's intended meaning was accurately interpreted by the model.

Notably, a more extensive revision process could require relabeling the gold standard, raising concerns that the gold standard might be altered to align with machine interpretation rather than the other way around. Additionally, excessive revision of code definitions introduces the risk of "overfitting" the codes to the specific dataset, making them too narrowly tailored to existing cases rather than broadly applicable. A potential safeguard against this issue is to divide the gold standard into separate "train" and "test" sets, deferring evaluation of LLM performance on the test set until after final codebook adjustments have been made. Importantly, this concern is not unique to LLM-assisted content analysis; manual coding is also vulnerable to overfitting, particularly when researchers iteratively refine definitions based on a limited set of texts without verifying their generalizability [41].

### 2.1 LLM-generated rationale are essential for evaluating performance

In adapting the codebook, we wanted to understand not just which codes the model struggled to interpret correctly, but what aspects of the code the model failed to capture. To achieve this, we structured our prompts to require GPT to justify its decision to apply or not apply each code. These rationales were invaluable. They often highlighted parts of the code description that were ambiguous or imprecisely defined, leading the model to misinterpret them. Whenever a rationale repeatedly pointed to such an issue, we revised the corresponding code. We then retested the passage to check that the code was correctly applied and the rationale aligned with our intended interpretation of the code. Sometimes a revision would not improve the interpretations for the passages in question; other times it corrected those case but introduced new problems in passages which were previously coded correctly.

Figure 2 demonstrates an effective method of prompting GPT to provide rationale for its code selections. The initial instruction is given by the Justification section of the prompt, and solicited again in the Decision/Formatting box.

### 2.2 LLMs require more precise descriptions than do human readers

Human coders do not rely solely on a written codebook. Their interpretation of the codes is enriched through the codebook development process, discussions with fellow coders, and supplementary oral instructions. An LLM lacks this interactive and historical context and must interpret codes entirely from written descriptions. Our work modifying the codebook for GPT revealed information that, while implicitly understood by the code developers, wasn't explicitly stated in the code descriptions. This process not only aided in refining the codebook for automated coding, but also improved our own understandings of the codes. This ultimately led to clearer definition of the codes, thereby enhancing future manual coding processes as well. Figure 3 demonstrates how the Monumental

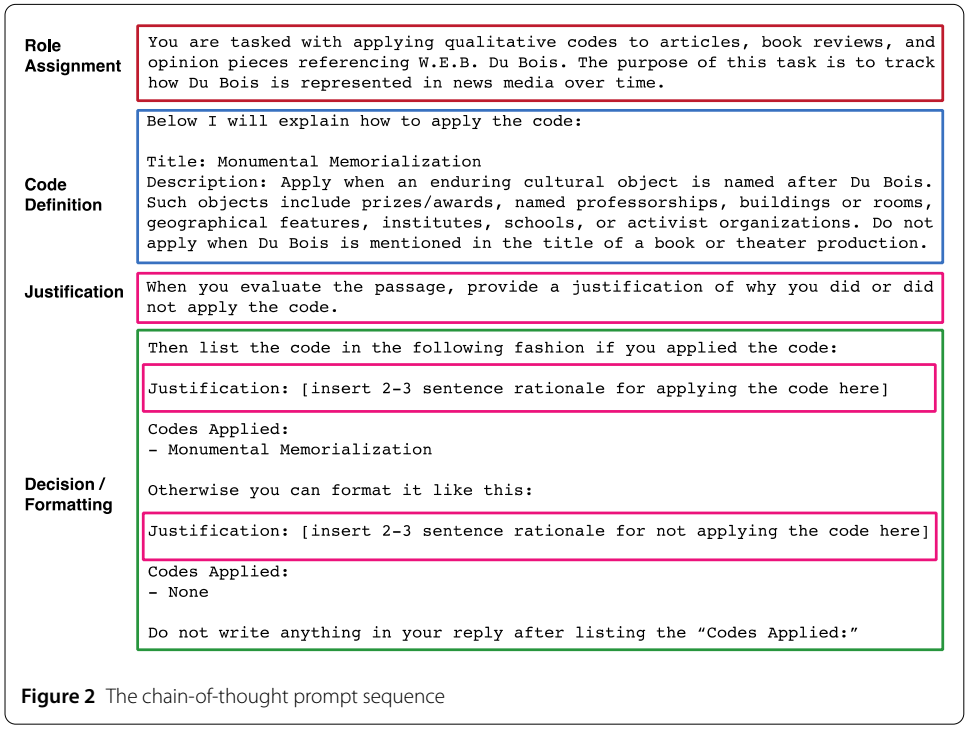


**Figure 2** The chain-of-thought prompt sequence

**Original: Activist**

Apply when Du Bois is developing activist organizations, giving public speeches, participating in meetings with politicians and organizers, running for office, promoting a candidate, organization or initiative. **Also apply when Du Bois is *explicitly* described as an activist or leader.** (A)

**Redefined: Activist**

(A) **Apply this code when Du Bois is *explicitly* called an "activist" or "leader",** or **when his political or social activism is either *explicitly noted or clearly implied through context.*** Examples include being mentioned in the context of leadership, activism, developing activist organizations, giving public speeches, participating in meetings with politicians and organizers, running for office, or promoting a candidate, organization, or initiative. (B)

**Original: Social/Political Activism** (C)

(D) **Refers to any form of social or political participation in promoting change in society. Can be used in conjunction with Du Bois' scholarly work, specifically when Du Bois's ideas are used to frame Black political struggle. Can be used** (E) (F) **to describe Du Bois' political work or those who invoke Du Bois in service of their activism.** (G)

**Redefined: Social/Political Advocacy** (C)

(D) **This code applies when a passage mentions or *implies* any form of social or political activism, advocacy, critique, or discourse, including discussions about current or historical social problems. This includes not only direct activism** (F) **of Du Bois and others, but also the framing and challenging of social norms,** (E) **historical narratives, and racial or cultural identities. Apply this code when Du Bois's work, persona, or ideas are invoked in discussions that critically** (G) **engage with Black identity, positionality, or broader systemic circumstances of Black people. Adjacency to other activists, such as inclusion in a list, is** (H) **insufficient; advocacy must be *explicitly mentioned* in the passage.**

**Figure 3** Two examples of prompt redefinition. Colored, alphabetically labeled blocks of text show alterations derived through iterative code refinement. Italics draw attention to direction to constrain interpretive scope to implicit or explicit information

Memorialization and Social/Political Advocacy codes were redefined to improve GPT's comprehension.

Often, we encountered cases where ambiguous phrasing was obvious to humans, but challenging for the LLM. Our codebook contains two codes that relate to Du Bois's reputation among academics and activists. These codes are meant to evaluate whether Du Bois appears in a news story because either an academic or activist mentioned him. Initially, we titled this code "Academic Repute," which worked well for human coders. GPT, however, consistently misinterpreted this code as pertaining to Du Bois's esteem *as* an academic, rather than *among* or *by* academics (the meaning of "among" remains ambiguous even here). We tried numerous iterations of this code without success. Nevertheless, altering the title of the code to the far more literal "Out of the Mouth of Academics" improved performance, even when paired with the original code description. Other research on prompting for content analysis and more broadly support the importance of word choice for code labels [27, 28] and task descriptions [28, 42, 43].

In another case, the code titled "Social/Political Activism" was revised to "Social/Political Advocacy" (Fig. 3 C) because GPT did not consider social critique to be a form of activism, even when it was specifically instructed to. While none of these changes in word choice dramatically improved LLM-interpretations as measured by Cohen's $\kappa$, they greatly reduced systematic error in GPT's output. This systematic error and its correction was evident from reviewing GPT's rationales, and is suggested by the large difference in the number of codes identified by GPT as compared to the gold standard data visible in Table 2. We found that words indicating how much the model should draw on context or its own outside knowledge had substantial impacts on the model's outputs, often to the desired effect. In particular, instructing the model to restrict itself to "explicit" meanings, or to draw on "implicit" meanings, often helped the model with part of a code description it had struggled with, a finding supported by White et al. [42]. Figure 3 B and D demonstrate the addition of such phrasing to control scope. For instance, the inclusion of edit B, replacing "when his political or social activism is referenced" with "when his political or social activism is either explicitly noted or clearly implied through context" lifts Cohen's $\kappa$ from 0.73 to 0.81.

Both mandatory (do) and prohibitory (do not) phrasing were observed by the model, though mandatory phrasing seemed more successful, a finding reported by others [28, 42, 44]. The ordering of directives also impacted how likely the model was to follow them. We found that moving a phrase that was ignored in the coding rationale toward the front of the definition made the model more likely to follow its specifications, as in Fig. 3 A. When a very specific problem was observed repeatedly, it was sometimes necessary to add a directive to correct it, as in Fig. 3 H.

### 2.3 Prompting for machine-readable output

To fully automate the coding process, model output must be reliably readable by a computer. The LLM generates text, which must be parsed by another script into a data structure, such as a table, for further analysis. Instructing exactly how to format the output produced machine-readable results with GPT-4 and GPT-3.5. Critically, this involves specifying a tag that the interpreting script locates, after which follows a reliably formatted list of codes. The Decision/Formatting component Fig. 2 illustrates how to constrain model output and produce consistent results across queries. Subsequent to completing this analysis we learned that the LLM can be instructed to output results in json, which is machine

readable using standard parsing libraries. Additionally, because models can be excessively verbose and summarize their output, particularly at higher temperatures, we informed the model that we did not want any output to follow the code list.

## 3 Evaluating model performance and prompting strategy

Once the code descriptions have been revised for LLM text categorization, numerous other decisions remain about how to prompt a model to execute the content analysis. We present these as a separate step for the sake of clarity, but in reality, we developed our approach iteratively and in tandem with revising the code descriptions. We hope future methodologists and empiricists will benefit from what we learned during this process, and that less exploration of these components will be necessary so practitioners can focus on application or exploration of calibrations not explored here. We summarize all our recommendations for qualitative coding with an LLM in Table 4.

There is a large and growing body of academic and nonacademic literature on prompt engineering [e.g., [45]]: constructing user-defined input to elicit the best model output. In fact, the codebook adaptation in the previous section was in large part an exercise in prompt engineering. However, in this second section, prompt engineering refers more to the broader context of task description than the code definitions. In this section we report how different prompts influence the quality of machine categorization. Additionally, we compare performance when the LLM is tasked with assigning each code independently to when the model is given the full codebook and instructed to code all nine codes as a single task. We refer to these as the "Per Code" and "Full Codebook" approach respectively.

Studies have shown that LLM decision-making improves when the model is prompted to account for its decisions [43, 46]. This is generally known as chain-of-thought (CoT) prompting or reasoning, and refers to breaking down tasks into specific components, one or more of which involves planning for future steps or reflection on previous ones. While some studies have demonstrated that chain-of-thought prompting does not always result in reliable explanations of LLM output [47, 48], it is widely recognized that chain-of-thought improves model performance and provides insight into model operations [49–51]. Our prompts apply chain-of-thought prompting by including 1) a role assignment step, informing the machine of its purpose, 2) a task description step, specifying the code definition, 3) a justification step, instructing the model to provide a rationale for its decision, and 4) a decision step, wherein the model delivers its ultimate analysis in a consistent, machine-readable format. An example of the chain-of-thought prompt sequence used in this study is given by Fig. 2.

We use zero-shot prompts throughout this study. Zero-shot refers to providing the model only the task description, without giving examples of correctly executed responses. Xiao et al. found few-shot prompting improves coding and performance on other tasks [11], whereas Chew et al. largely employed zero-shot prompts [12]. Our case study involves evaluating paragraph-long passages rather than single clauses. We found that information in the examples was drawn upon by the LLM and interfered with its coding decisions. We also found that in Full Codebook prompts, giving examples greatly expanded the prompt, negatively impacting results. We suggest that when content is more literary or historical, zero-shot prompts are probably preferred, but that most coding tasks will benefit from few-shot prompting as demonstrated by the results of many other studies across domains.

We used the default settings for the GPT API where temperature is set to 0 and nucleus sampling (top_p) is set to 1. We specified the task description as a "system prompt," and provide each passage as a "user prompt." A system prompt gives the LLM its purpose, clearly specifying the task it is meant to address, whereas a user prompt provides the input to which the model responds by generating output. We did not investigate whether intercoder agreement suffers with the default system prompt, while combining the task description and passage as a user prompt.

Performance comparisons are relative to the human-derived gold standard following application of the initial codebook to 111 passages.

### 3.1 GPT-4 greatly outperforms GPT-3.5

We found that GPT-4 approaches human performance for three codes: Activist: $\kappa$ = 0.81; Monumental Memorialization: $\kappa$ = 1.00; Collective Synecdoche: $\kappa$ = 0.79. GPT-4 prompted for rationales provides considerably higher quality code assignments than GPT-3.5, except in the case of the Out of the Mouth of Activists code, which no configuration handled well. It is especially notable that GPT-4 and GPT-3.5 differed in their most accurately interpreted codes. In the three tasks GPT-4 executed best, GPT-3.5's performance was slightly below its own average, $mean(\kappa_{all})$ = 0.34 vs. $mean(\kappa)$ = 0.32.

### 3.2 Coding fidelity improves when codes are presented as individual tasks

We adapted our codebook by presenting the entire codebook to GPT along with task instructions. However, we found in testing that performance improved when GPT was given each task independently. This "per code" approach was taken by one recent study exploring content analysis with non-mutually exclusive codes (permitting multiply coded passages) [12], but not another, which tested only two codes [11]. Table 3 compares the GPT-4 performance when presented individual tasks for each code ("Per Code") and when presented with all tasks in a single prompt ("Full Codebook"). We found that for the three human-equivalent tasks (Activist, Monumental Memorialization, and Collective Synecdoche) the Per Code performance far exceeded the Full Codebook for two tasks, and was

**Table 3** Intercoder reliability (Cohen's $\kappa$) for all codes on 111 gold standard passages. The third column gives the agreement of the human coders. Remaining columns indicate agreement between LLM and human-derived gold standard for different configurations. Best overall performance by an LLM is shown in bold. Italics indicate the highest intercoder reliability between pairs with and without prompting for rationale (CoT vs. No CoT); if the pair are equivalent neither is italicized. Two values are considered equivalent if their difference does not exceed 0.02

| Code | Human Coders | GPT-4 | | | | GPT-3.5 | |
|---|---|---|---|---|---|---|---|
| | | Per Code | | Full Codebook | | Per Code | |
| | | CoT | No CoT | CoT | No CoT | CoT | No CoT |
| Scholar | 0.76 | 0.61 | 0.52 | 0.59 | 0.42 | *0.29* | 0.21 |
| Activist | 0.76 | 0.81 | 0.65 | *0.67* | 0.62 | *0.39* | 0.32 |
| Monumental Memorialization | 0.94 | 1.00 | 0.91 | *0.75* | 0.48 | 0.29 | 0.31 |
| Mention of Scholarly Work | 0.78 | 0.71 | 0.69 | *0.52* | 0.44 | 0.33 | *0.39* |
| Social/Political Advocacy | 0.70 | 0.64 | 0.60 | 0.60 | 0.60 | *0.55* | 0.51 |
| Coalition Building | 0.69 | 0.60 | 0.44 | *0.43* | 0.13 | *0.33* | 0.17 |
| Out of the Mouth of Academics | 0.71 | 0.63 | 0.65 | 0.65 | 0.62 | *0.37* | 0.33 |
| Out of the Mouth of Activists | 0.64 | *0.30* | 0.09 | 0.34 | 0.18 | *0.21* | 0.09 |
| Collective Synecdoche | 0.86 | 0.79 | 0.78 | 0.81 | 0.71 | 0.27 | 0.27 |
| Mean | 0.78 | 0.68 | 0.59 | *0.60* | 0.46 | *0.34* | 0.29 |

comparable for one. For two other tasks, Mention of Scholarly Work and Coalition Building, we found that the Per Code configuration produced considerably higher agreement, whereas Full Codebook performed comparably to the Per Code in the remaining four tasks.

### 3.3 Coding fidelity improves when the model is prompted to explain its coding decisions

Consistent with other experiments with chain-of-thought (CoT) reasoning in LLMs, we found that coding agreement benefited strongly from prompting the model to explain itself [52]. Table 3 shows the effect of prompting for rationale on three pairs of conditions: Per Code GPT-4, Full Codebook GPT-4, and Per Code GPT-3.5. We found that across all codes and conditions, with one exception, CoT prompting produces higher or equivalent intercoder reliability with the gold standard. Using GPT-4, average Per Code agreement improved from 0.59 to 0.68, and average Full Codebook agreement improved from 0.46 to 0.60. Moreover, a majority of pairs showed substantial improvement when the codes were assigned after providing reasoning for coding decisions.

## 4 Discussion

This paper is designed as a practical guide for researchers seeking to integrate large language models (LLMs) into qualitative coding workflows. Specifically, it outlines strategies for maximizing LLM performance while retaining critical aspects of exploratory hermeneutics and human oversight. To this end, we have:

- Demonstrated a hybrid automated workflow that retains key elements of the traditional qualitative coding process, substituting an LLM for human coders while maintaining human-led codebook development;
- Provided a detailed example of an effective chain-of-thought prompt design (Fig. 2);
- Shared insights and challenges encountered during the iterative process of adapting a codebook for LLM use;
- Offered guidance for evaluating and comparing performance across models and prompt designs.

We hope this framework equips practitioners with actionable methods for leveraging LLMs in qualitative research. The best practices summarized in Table 4 serve as a reference point for designing prompts that align model classifications with human interpretive standards.

Critically, this guide does not serve as a blanket endorsement of GPT-4 or other LLMs for automated coding. Instead, it provides a structured approach for researchers to evaluate and improve LLM performance within specific contexts. By closely replicating the workflow for manual coding, this process ensures that codes with low fidelity are either excluded from the dataset or flagged for manual coding, preserving the rigor of qualitative analysis. Unlike traditional machine learning workflows, which often focus on algorithm selection and optimization, our human-in-the-loop approach relies on an iterative process of refining both the codebook and model prompts. This ensures that the interpretative nuance required for qualitative research remains central to the analysis, fostering a deeper understanding of model behavior and limitations. In the best cases, as in traditional grounded theory, adapting a codebook for an LLM can lead to a better understanding of the case under study and thus improved intuition and hypotheses.

**Table 4** Principles of prompting an LLM for qualitative coding with references to additional supporting studies

| | |
|---|---|
| Task Instructions | |
| *Prompt for Rationale* | Model fidelity improves when instructed to justify its coding decisions [12, 28, 42]. |
| *One Task Per Code* | Model fidelity improves when given each code as a separate task [12, 28, 42]. |
| *Brevity* | Shorter task descriptions will be more faithfully interpreted by the model [28, 42, 53]. |
| *Structured Output* | Instruct the model to format its output to ensure uniform responses, e.g., json [12, 27, 42]. |
| Code Definitions | |
| *Word Choice* | A single high-content word can be changed to align with the LLM's built-in ontology [28, 42, 43]. |
| *Clause Order* | Clauses are more likely to be observed when introduced earlier in the code description [27, 28, 42]. |
| *Mandates/Prohibitions* | Both can be effective, but it is easier to get the model to "do" than "do not" [28, 42, 44]. |
| *Code Titles* | Altering the code title can have a large effect even without altering the definition [27, 28]. |
| *Interpretation Scope* | Use words like "implicit" and "explicit" when interpretation is too limited or expansive [42]. |
| Chain-of-Thought Prompt Sequence | |
| 1. *Role Assignment* | Supply the model its purpose, e.g., "You will be applying category labels to passages." |
| 2. *Code Definition* | Provide the code title(s) and description(s). |
| 3. *Justification* | Request that the model provide evidence of its reasoning. |
| 4. *Decision* | Instruct the model to list the codes that apply to the passage in a consistent format. |

### 4.1 Determining appropriate domains for LLM-assisted qualitative coding

Previous methods of automated text categorization, both supervised and unsupervised, rarely met the standards of traditional social scientists and humanists, and were instead generally employed by data scientists. Capturing meaning, particularly complex meaning, through machine learning has largely been an elusive goal [3]. Despite our own former skepticism, we predict that LLMs will be capable of applying most qualitative codebooks within the year. However, our results show that even within the scope of a single codebook, interpretation quality varies. Thus, different disciplines and domains should expect model success and the ease of transitioning a codebook to vary considerably. We suspect that more humanistic and "softer" scientific approaches will (continue to) be more resistant to machine interpretation than problems posed by scholars who identify with "harder" sciences, to say nothing of their ability to convince their peers of its validity. We do not oppose developing evaluation benchmarks for qualitative coding to assess which models are adept at what variety of task, but neither do we advocate it; meaning is manifold and emergent, and much of its beauty derives from its resistance to reduction and definition. Instead, we suggest those who wish to employ an LLM to perform content analysis survey similar attempts and simply experiment on their own. The process of discovering triumphs, workarounds, and limitations of working with these models was not only fascinating, but tremendously fun.

### 4.2 Practical aspects of transitioning to content analysis with LLMs

While artificial intelligence potentially opens much larger datasets to qualitative scholars, there is still a considerable technical barrier to automating content analysis. Development of an LLM-adapted codebook is feasible for anyone regardless of technical skill by interacting with an LLM through chat-like Web platforms provided by proprietary model developers. However, systematically testing prompts or applying a completed codebook to the full dataset requires moderate skill in writing scripts in a language such as Python.

Rather than suggest that all scholars become programmers, we encourage researchers to develop partnerships with students or community members seeking programming or research experience as a form of project-based education. Conversely, we suggest that data scientists pursue partnerships with traditional social scientists and humanists, who are often better positioned to develop coding schema to flush out complex meanings embedded in text, which are now more tractable to machine learning.

### 4.3 Handling passages where model interpretation is poor

Overwhelmingly, GPT-4's interpretations were accurate and human-like. However, we found repeatedly that GPT-4, like a human reader, struggled with edge cases, especially where implicit information was required to make a judgment. We are encouraged by this finding, and argue that with automated analysis, fidelity is less important than it is with humans. Because statistical power increases with the number of observations, noise is more tolerable in machine-applied codes, as automated coding potentially increases sample size by orders of magnitude. Notably, this assumes that error is restricted to edge cases and is not otherwise systematically biased. We also advise against automated coding where datasets are small, as in interviews, where it is likely as efficient to code entirely by hand. As models improve and can provide confidence estimates for their statements [54–56], ML content analysis workflows should include manual review of passages with uncertain code assignments. Anecdotally, we found that GPT-4 could intelligently reflect on its responses when prompted to do so. When presented the output of another model instance, GPT-4, acting as an untrained "critic" model [57], was often able to identify when it had encountered an edge case without prompting, as well as recognize and revise obvious mistakes. Our experiences suggest that a human-in-the-loop tag-for-manual-review workflow or a multi-step automated reflect-and-revise workflow is feasible with frontier models.

### 4.4 Model selection: versions and open-source vs. proprietary

When considering LLM-assisted coding, we must acknowledge the non-stationarity of LLMs and the implications of model selection. Model names such as GPT-4, Claude 3.5, or Llama 3 refer not to singular entities but to families of models, each subject to iterative updates. Proprietary model APIs typically allow selection of specific versions, while open-source platforms, such as Hugging Face, provide even greater control, enabling researchers to lock in precise versions. Despite these differences, reliance on any LLM requires ongoing validation to ensure fidelity in outputs and one cannot assume that a model that worked for one task will work equally well for another task, even if that task is similar or even the same.

Our own subsequent work using GPT-4 for LLM-assisted coding illustrates the importance of model version in LLM performance. We conducted the present study in January 2024 using gpt-4-1106-preview. Subsequently, OpenAI has released many models which are nominally GPT-4. This includes gpt-4o-mini, which costs less than 20% the rate of gpt-4o per token at the time of publication. In applying this method toward later empirical studies, we found that GPT-4o-mini did not perform with nearly the fidelity of GPT-4o and struggled to follow formatting instructions. Particularly concerning, we found that GPT-4o-mini did not follow instructions to classify authors' gender as "man" or "woman" and frequently returned "none," despite explicit instruction to the contrary; GPT-4o always followed these instructions. The human-in-the-loop workflow advocated here, in particular the validation of model coding fidelity, is critical to ensure that analysis is derived from

high quality. Moreover, we should not necessarily trust model producers to release superior models over time, as model performance is weighed against computational efficiency and developers' goals and intended model uses may shift.

This variability in LLM performance intersects with broader considerations about the accessibility and ethics of open versus proprietary models. While open-source models promote transparency and reproducibility, they often lag behind proprietary systems in performance metrics. At the time of our study, open-source alternatives to GPT-4 could not meet the demands of our coding tasks. However, proprietary models also pose challenges, particularly their non-auditable architectures and potential susceptibility to unannounced updates. As Palmer et al. [58] advocate, researchers should explicitly justify their choice of a proprietary model, detailing why their use is critical and what open-source developments would mitigate such reliance.

To align with these best practices, we justify our reliance on GPT-4 on account of its frontier capability that was unmatched by open-source models at the time of our study. This state-of-the-art performance was necessary for our case study, as evidenced by GPT-3.5's inability to sufficiently interpret any of the codes. Moreover, proprietary APIs offer infrastructure that eliminates the need for extensive computational resources or advanced technical expertise, allowing researchers to access and deploy cutting-edge models with minimal setup. While researchers with strong technical skills and access to high-performance computing systems can run open-source models themselves, most practitioners will rely on APIs. At the time we conducted this study, open-source model availability through APIs was more limited. However, now open-source models such as Llama and Mistral are now available through APIs by Meta, Mistral, and Google's Vertex AI. Using these APIs, virtually any researcher can access open-source models, though they will still most likely be patronizing the producers of proprietary software.

## 5 Conclusion

Our findings demonstrate that large language models (LLMs) can serve as a transformative tool for qualitative coding, offering an opportunity to scale traditional workflows while preserving their interpretive depth. By positioning iterative codebook refinement downstream of human-led codebook development, we have shown how these models can be effectively integrated into content analysis without undermining the reflexive rigor central to qualitative research. This hybrid approach not only aligns automated methods with human interpretive standards but also encourages researchers to remain at the forefront of innovation by adapting their practices to evolving technologies.

While the efficiency of automation is compelling, the true promise of LLM-assisted coding lies in its ability to expand the scope of inquiry. By making it feasible to analyze datasets numbering in the thousands or more, these tools open new possibilities for identifying subtle patterns and rare phenomena that would otherwise remain obscured in smaller samples. As models continue to advance—incorporating memory [59], collaborative (multi-agential) reasoning [60, 61], and the capacity to process larger contexts [62]—their potential to enhance qualitative research should grow significantly.

However, as we scale the application of qualitative codes, it is vital to preserve the reflexive and critical elements that define qualitative research. LLMs should not be seen as a substitute for human interpretation but as a tool that enhances researchers' ability to engage with complex data and generate deeper insights. By adopting these tools thoughtfully, researchers can embrace their potential to scale inquiry while maintaining the interpretive

rigor necessary for meaningful analysis. The strategies outlined in this paper highlight a path forward, emphasizing a collaborative relationship between human expertise and machine efficiency. This synergy has the potential to not only extend the scope of qualitative research but also deepen its impact, uncovering patterns and dynamics that advance our understanding of social and cultural phenomena.

## Appendix: Code descriptions

Scholar

*Original Definition:*
Describes Du Bois as a scholar, intellectual, or sociologist/historian/anthropologist. When Du Bois is invoked through his ideas that speak to Black politics or Black positionality, such as double consciousness, he should be classified as a scholar, not an activist.

*Revised Definition:*
Applies when Du Bois is described as a scholar or intellectual, especially in connection to Black politics, racial identity, or social theory. When Du Bois is invoked through his ideas on social theory, he should be classified as a scholar, not an activist, unless it is a call to action, related to his organizing, or other non-scholarly political activity. Do not apply when Du Bois is merely the focus in the context of historical and academic study or his scholarship is only implied by loose connections to other scholars.

Activist

*Original Definition:*
Discussions on W.E.B. Du Bois's political or social activism for social change and racial justice in the US or global context; referenced as a "leader." Should be used when Du Bois is described as both an activist and prominent Black intellectual.

*Revised Definition:*
Apply this code when Du Bois is explicitly called an "activist" or "leader," or when his political or social activism is either explicitly noted or clearly implied through context. Examples include being mentioned in the context of leadership, activism, developing activist organizations, giving public speeches, participating in meetings with politicians and organizers, running for office, or promoting a candidate, organization, or initiative.

Monumental Memorialization

*Definition:* (unchanged)
Apply this code when an enduring cultural object is named after Du Bois. Such objects include prizes/awards, named professorships, buildings or rooms, geographical features, institutes, schools, or activist organizations.

Mention of Scholarly Work

*Original Definition:*
Mentions or quotes specific academic works by Du Bois. This includes both named and unnamed pieces of writing, major theoretical concepts, or references to his body of scholarly writings.

*Revised Definition:*
Apply this code when academic works or major theoretical concepts by W.E.B. Du Bois are mentioned or quoted. This includes explicit naming or direct quotations of his writings and references to his key academic ideas, even if unnamed, provided they are clearly attributed to him. Only use when a quote comes from a scholarly work; use context to determine whether a quote comes from a scholarly work, such as a history or social theory, or some other piece, such as a letter or speech. Avoid using this code for general references to Du Bois's influence, body of writings, or activities outside of his scholarly work. Do not apply it when discussing others' work, unless Du Bois's scholarly concepts or writings are explicitly and centrally mentioned.

Social/Political Advocacy

*Original Title:* Social/Political Activism *Original Definition:*
Refers to any form of social or political participation in promoting change in society. Can be used in conjunction with Du Bois's scholarly work, specifically when Du Bois's ideas are used to frame Black political struggle. Can be used to describe Du Bois's political work or those who invoke Du Bois in service of their activism.

*Revised Definition:*
This code applies when a passage mentions or implies any form of social or political activism, advocacy, critique, or discourse, including discussions about current or historical social problems. This includes not only direct activism of Du Bois and others, but also the framing and challenging of social norms, historical narratives, and racial or cultural identities. Apply this code when Du Bois's work, persona, or ideas are invoked in discussions that critically engage with Black identity, positionality, or broader systemic circumstances of Black people. Adjacency to other activists such as inclusion in a list is insufficient; advocacy must be explicitly mentioned in the passage.

Coalition Building

*Definition:* (unchanged)
Du Bois is described as an agent establishing his reputation through organizational and institutional sponsorship.

Out of the Mouth of Academics

*Original Title:* Academic Repute

*Original Definition:*
Used when a specific member of an academic organization is engaging with Du Bois's work or Du Bois as representing a concept.

*Revised Definition:*
Apply this code when a specific member of an academic organization is engaging with Du Bois's work or Du Bois as representing a concept. Apply also when an activist organization has named itself or a subdivision of itself after Du Bois, such as an institute or named professorship.

Out of the Mouth of Activists

*Original Title:* Activist Repute

*Original Definition:*
Used when a specific member of an activist organization is engaging with Du Bois's work or Du Bois as representing a concept. Also when an activist organization has named itself or a subdivision of itself after Du Bois, such as a foundation or prize. Activist organizations include but are not limited to political parties, groups such as the NAACP, and the Black church.

*Revised Definition:*
Apply this code when an individual described as a leader, activist, or politician, or as a member of a political or activist organization (e.g., political party, the NAACP, the Black church, Black Lives Matter) references or draws upon W.E.B. Du Bois's work or legacy. Apply when the person or organization invokes Du Bois in the context of supporting or advancing their social or political agenda, not when a tangential connection is made by the author of the passage (i.e., "From the Mouth of"). Also apply the code when governments, political parties, or activist organizations connect their agenda to Du Bois, such as commemorating Du Bois or by naming initiatives (like foundations or prizes) after him. Do not apply the code if Du Bois is mentioned solely as a member of an organization, unless the organization is using Du Bois's membership to further their agenda.

Collective Synecdoche

*Definition:* (unchanged)
Mentioned with other famous intellectuals, activists, or public figures in order to represent some facet of a culture, era, or ideology. Examples include race scholarship, civil rights activism, left political leaders, Black excellence, and 20th-century political commentators.

**Abbreviations**
CoT, Chain-of-thought; LLM, Large language model.

**Acknowledgements**
The author thanks Harry Yan, Pat Wall, Patrick Kaminski, Adam Fisch, Alicia Chen, Francisco Muñoz and the anonymous reviewers for their comments toward improving this manuscript. I am especially grateful to Tania Ravaei for collaborating on codebook development.

**Author contributions**
Tania Ravaei assisted in developing the initial codebook and applying codes to establish the gold standard data as part of a yet-to-be-published study. The sole author is responsible for all other work. The author read and approved the final manuscript.

**Funding information**
The author has no funding to declare.

**Data availability**
Code and data available at https://osf.io/k4fg9.

## Declarations

**Ethics approval and consent to participate**
There are no human subjects in this study.

**Consent for publication**
The author consented to publication of this work. A previous version of this paper was published at https://arxiv.org/abs/2401.15170 as Scalable qualitative coding with LLMs: Chain-of-thought reasoning matches human performance in some hermeneutic tasks.

**Competing interests**
The author declares no competing interests.

Received: 8 February 2024 Accepted: 31 March 2025 Published online: 02 April 2025

**References**


1. Strauss AL (1967) The discovery of grounded theory: strategies for qualitative research. Aldine, New York. https://doi.org/10.2307/2094063
2. Saldaña J (2021) The coding manual for qualitative researchers, 4th edn. SAGE, Thousand Oaks. https://doi.org/10.29333/ajqr/12085
3. Malik MM (2020) A hierarchy of limitations in machine learning. https://doi.org/10.1088/0954-898x_4_3_007. arXiv preprint arXiv:2002.05193
4. Nelson LK (2017) Computational grounded theory: a methodological framework. Sociol Methods Res 49(1):3–42. https://doi.org/10.1177/0049124117729703
5. Dhar A, Mukherjee H, Dash NS, Roy K (2021) Text categorization: past and present. Artif Intell Rev 54:3007–3054. https://doi.org/10.1007/s10462-020-09919-1
6. Bubeck S, Chandrasekaran V, Eldan R, Gehrke J, Horvitz E, Kamar E, Lee P, Lee YT, Li Y, Lundberg S, et al (2023) Sparks of artificial general intelligence: early experiments with GPT-4. https://doi.org/10.3390/a16090432. arXiv preprint arXiv:2303.12712
7. Romera-Paredes B, Barekatain M, Novikov A, Balog M, Kumar MP, Dupont E, Ruiz FJ, Ellenberg JS, Wang P, Fawzi O, et al (2023) Mathematical discoveries from program search with large language models. Nature 625:1–3. https://doi.org/10.1038/s41586-023-06924-6
8. Anil R, Borgeaud S, Wu Y, Alayrac JB, Yu J, Soricut R, Schalkwyk J, Dai AM, Hauth A, et al (G Team) (2023) Gemini: a family of highly capable multimodal models. https://doi.org/10.2139/ssrn.4703935. arXiv preprint arXiv:2312.11805
9. Gilardi F, Alizadeh M, Kubli M (2023) Chatgpt outperforms crowd workers for text-annotation tasks. Proc Natl Acad Sci 120(30):e2305016120. https://doi.org/10.1073/pnas.2305016120
10. Strauss A, Corbin JM (1997) Grounded theory in practice. SAGE, Thousand Oaks. https://doi.org/10.2307/2655359
11. Xiao Z, Yuan X, Liao QV, Abdelghani R, Oudeyer PY (2023) Supporting qualitative analysis with large language models: combining codebook with GPT-3 for deductive coding. https://doi.org/10.1145/3581754.3584136
12. Chew R, Bollenbacher J, Wenger M, Speer J, Kim A (2023) LLM-assisted content analysis: using large language models to support deductive coding. https://doi.org/10.1145/3613905.3650828. arXiv preprint arXiv:2306.14924
13. Törnberg P (2023) ChatGPT-4 outperforms experts and crowd workers in annotating political Twitter messages with zero-shot learning. https://doi.org/10.1016/b978-0-12-809276-7.00018-7. arXiv preprint arXiv:2304.06588
14. Kadhim AI (2019) Survey on supervised machine learning techniques for automatic text classification. Artif Intell Rev 52(1):273–292. https://doi.org/10.1007/s10462-018-09677-1
15. Blei DM, Ng AY, Jordan MI (2003) Latent Dirichlet allocation. J Mach Learn Res 3(1):993–1022. https://doi.org/10.1002/0471684228.egp03419
16. Jelodar H, Wang Y, Yuan C, Feng X, Jiang X, Li Y, Zhao L (2019) Latent Dirichlet allocation (LDA) and topic modeling: models, applications, a survey. Multimed Tools Appl 78:15169–15211. https://doi.org/10.1007/s11042018-6894-4
17. Grootendorst M (2022) BERTopic: neural topic modeling with a class-based TF-IDF procedure. https://doi.org/10.1109/ictech55460.2022.00112. arXiv preprint arXiv:2203.05794
18. Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, et al (2020) Language models are few-shot learners. Adv Neural Inf Process Syst 33:1877–1901. https://doi.org/10.18653/v1/2021.mrl-1.1
19. Touvron H, Lavril T, Izacard G, Martinet X, Lachaux MA, Lacroix T, Rozière B, Goyal N, Hambro E, Azhar F, et al (2023) LLaMa: open and efficient foundation language models. https://doi.org/10.21203/rs.3.rs-4240043/v1. arXiv preprint arXiv:2302.13971
20. MistralAI (2023) Mixtral of experts: a high quality sparse mixture-of-experts. https://doi.org/10.1109/ssp.2011.5967785. https://mistral.ai/news/mixtral-of-experts. Accessed: 13 Jan 2024
21. Anthropic (2023) Claude 2. https://doi.org/10.1093/benz/9780199773787.article.b00038962. https://www.anthropic.com/index/claude-2. Accessed: 18 Jan 2024
22. Nowakowski J, Keller J (2024) AI-powered patching: the future of automated vulnerability fixes. Google Secur Eng Tech Rep. https://doi.org/10.1109/ase56229.2023.00021
23. Gómez-Rodríguez C, Williams P (2023) A confederacy of models: a comprehensive evaluation of LLMs on creative writing. https://doi.org/10.18653/v1/2023.findings-emnlp.966. arXiv preprint arXiv:2310.08433
24. Bischoff M (2024) AI matches the abilities of the best. Math Olympians Sci Am. https://doi.org/10.5948/9781614444046. https://www.scientificamerican.com/article/ai-matches-the-abilities-of-the-best-math-olympians/
25. Vach W, Gerke O (2023) Gwet's AC1 is not a substitute for Cohen's kappa – a comparison of basic properties. MethodsX 10:102212. https://doi.org/10.1016/j.mex.2023.102212
26. Ollion E, Shen R, Macanovic A, Chatelain A (2023) ChatGPT for text annotation? Mind the hype! SocArXiv preprint https://doi.org/10.31235/osf.io/x58kn
27. Yu D, Li L, Su H, Fuoli M (2024) Assessing the potential of LLM-assisted annotation for corpus-based pragmatics and discourse analysis: the case of apology. Int J Corpus Linguist 29:534–561. https://doi.org/10.1075/ijcl.23087.yu
28. Halterman A, Keith KA (2024) Codebook LLMs: adapting political science codebooks for LLM use and adapting LLMs to follow codebooks. https://doi.org/10.1109/reepe51337.2021.9388063. arXiv preprint arXiv:2407.10747
29. Törnberg P (2024) Best practices for text annotation with large language models. Sociologica 18. https://doi.org/10.5220/0013267100003911
30. Merriam SB, Tisdell EJ (2015) Qualitative research: a guide to design and implementation, 4th edn. Wiley, New York. https://doi.org/10.46743/2160-3715/2022.5748
31. Gadamer HG (1975) Truth and method. Seabury Press, New York. https://doi.org/10.2307/430490
32. Weston C, Gandell T, Beauchamp J, McAlpine L, Wiseman C, Beauchamp C (2001) Analyzing interview data: the development and evolution of a coding system. Qual Sociol 24:381–400. https://doi.org/10.1016/s07408188(99)00024-9

33. Wa-Mbaleka S (2020) The researcher as an instrument. In: Moreira A, Costa AP, Reis LP (eds) Computer supported qualitative research: new trends on qualitative research (WCQR2019), pp 33–41. https://doi.org/10.1037/t37909-000
34. Braun V, Clarke V (2021) Thematic analysis: a practical guide. SAGE, Thousand Oaks. https://doi.org/10.1177/1035719x211058251
35. Maher C, Hadfield M, Hutchings M, De Eyto A (2018) Ensuring rigor in qualitative data analysis: a design research approach to coding combining NVivo with traditional material methods. Int J Qual Methods 17:1–13. https://doi.org/10.1177/1609406918786362
36. Campbell JL, Quincy C, Osserman J, Pedersen OK (2013) Coding in-depth semistructured interviews: problems of unitization and intercoder reliability and agreement. Sociol Methods Res 42(3):294–320. https://doi.org/10.1177/0049124113500475
37. Cascio MA, Lee E, Vaudrin N, Freedman DA (2019) A team-based approach to open coding: considerations for creating intercoder consensus. Field Methods 31(2):116–130. https://doi.org/10.1177/1525822x19838237
38. Talebirad Y, Nadiri A (2023) Multi-agent collaboration: harnessing the power of intelligent LLM agents. https://doi.org/10.1016/b978-0-12-809633-8.20328-2. arXiv preprint arXiv:2306.03314
39. Button KS, Ioannidis JP, Mokrysz C, Nosek BA, Flint J, Robinson ES, Munafò MR (2013) Power failure: why small sample size undermines the reliability of neuroscience. Nat Rev Neurosci 14(5):365–376. https://doi.org/10.1038/nrn3475
40. Baumer EP, Mimno D, Guha S, Quan E, Gay GK (2017) Comparing grounded theory and topic modeling: extreme divergence or unlikely convergence? J Assoc Inf Sci Technol 68(6):1397–1410. https://doi.org/10.1002/asi.23786
41. Miles MB, Huberman AM, Saldaña J, et al (2020) Qualitative data analysis: a methods sourcebook, 4th edn. SAGE, Thousand Oaks. https://doi.org/10.1177/239700221402800402
42. White J, Fu Q, Hays S, Sandborn M, Olea C, Gilbert H, Elnashar A, Spencer-Smith J, Schmidt DC (2023) A prompt pattern catalog to enhance prompt engineering with ChatGPT. https://doi.org/10.5121/csit.2024.140606. arXiv preprint arXiv:2302.11382
43. Wei J, Wang X, Schuurmans D, Bosma M, Xia F, Chi E, Le QV, Zhou D, et al (2022) Chain-of-thought prompting elicits reasoning in large language models. Adv Neural Inf Process Syst 35:24824–24837. https://doi.org/10.18653/v1/2024.emnlp-main.55
44. Bsharat SM, Myrzakhan A, Shen Z (2023) Principled instructions are all you need for questioning LLaMA-1/2, GPT-3.5/4. https://doi.org/10.26434/chemrxiv-2023-fw8n4-v2. arXiv preprint arXiv:2312.16171
45. Atreja S, Ashkinaze J, Li L, Mendelsohn J, Hemphill L (2024) Prompt design matters for computational social science tasks but in unpredictable ways. https://doi.org/10.1017/cbo9780511810503.011. arXiv preprint arXiv:2406.11980
46. Madaan A, Yazdanbakhsh A (2022) Text and patterns: for effective chain of thought, it takes two to tango. https://doi.org/10.18848/1447-9508/cgp/v06l03/42384. arXiv preprint arXiv:2209.07686
47. Ye X, Durrett G (2022) The unreliability of explanations in few-shot prompting for textual reasoning. Adv Neural Inf Process Syst 35:30378–30392. https://doi.org/10.18653/v1/2023.findings-acl.668
48. Turpin M, Michael J, Perez E, Bowman S (2023) Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. Adv Neural Inf Process Syst 36. https://doi.org/10.18653/v1/2024.acllong.73
49. Zhang Z, Zhang A, Li M, Smola A (2022) Automatic chain of thought prompting in large language models. https://doi.org/10.18653/v1/2024.acl-long.73. arXiv preprint arXiv:2210.03493
50. Feng G, Zhang B, Gu Y, Ye H, He D, Wang L (2024) Towards revealing the mystery behind chain of thought: a theoretical perspective. Adv Neural Inf Process Syst 36. https://doi.org/10.1016/j.fri.2020.200396
51. Yao S, Yu D, Zhao J, Shafran I, Griffiths T, Cao Y, Narasimhan K (2024) Tree of thoughts: deliberate problem solving with large language models. Adv Neural Inf Process Syst 36. https://doi.org/10.1109/issrew60843.2023.00040
52. Chu Z, Chen J, Chen Q, Yu W, He T, Wang H, Peng W, Liu M, Qin B, Liu T (2023) A survey of chain of thought reasoning: advances, frontiers and future. https://doi.org/10.18653/v1/2024.acllong.65. arXiv preprint arXiv:2309.15402
53. Levy M, Jacoby A, Goldberg Y (2024) Same task, more tokens: the impact of input length on the reasoning performance of large language models. https://doi.org/10.18653/v1/2024.acllong.818. arXiv preprint arXiv:2402.14848
54. Lin Z, Trivedi S, Sun J (2023) Generating with confidence: uncertainty quantification for black-box large language models. https://doi.org/10.1109/smc54092.2024.10831438. arXiv preprint arXiv:2305.19187
55. Zhou K, Jurafsky D, Hashimoto T (2023) Navigating the grey area: expressions of overconfidence and uncertainty in language models. https://doi.org/10.18653/v1/2023.emnlp-main.335. arXiv preprint arXiv:2302.13439
56. Chen J, Yoon J (2024) Introducing ASPIRE for selective prediction in LLMs. https://doi.org/10.18653/v1/2023.findingsemnlp.345. https://blog.research.google/2024/01/introducing-aspire-for-selective.html?m=1. Accessed: 20 Jan 2024
57. Paul D, Ismayilzada M, Peyrard M, Borges B, Bosselut A, West R, Faltings B (2023) REFINER: reasoning feedback on intermediate representations. https://doi.org/10.1016/b978-0-12088478-0.00005-0. arXiv preprint arXiv:2304.01904
58. Palmer A, Smith NA, Spirling A (2024) Using proprietary language models in academic research requires explicit justification. Nat Comput Sci 4(1):2–3. https://doi.org/10.1038/s43588-023-00585-1
59. Shinn N, Labash B, Gopinath A (2023) Reflexion: an autonomous agent with dynamic memory and self-reflection. https://doi.org/10.21240/mpaed/diss.ck/2022.02.28.x. arXiv preprint arXiv:2303.11366
60. Yao S, Zhao J, Yu D, Du N, Shafran I, Narasimhan K, Cao Y (2022) ReAct: synergizing reasoning and acting in language models. https://doi.org/10.18653/v1/2023.acl-long.830. arXiv preprint arXiv:2210.03629
61. Xu Z, Shi S, Hu B, Yu J, Li D, Zhang M, Wu Y (2023) Towards reasoning in large language models via multi-agent peer review collaboration. https://doi.org/10.18653/v1/2023.emnlp-main.13. arXiv preprint arXiv:2311.08152
62. Gu A, Dao T (2023) Mamba: linear-time sequence modeling with selective state spaces. https://doi.org/10.21437/interspeech.2024-1274. arXiv preprint arXiv:2312.00752


## Publisher's note