Scaling Hermeneutics: Qualitative Coding with LLMs for Reflexive Content Analysis

Dunivin, Z. O. (2025)

Presented by Yanyan Luo
2/24/2026

Paper Summary

Context: Qualitative coding is central to theory development in social science, but its interpretive depth makes it time-consuming and resistant to automation. LLMs now offer the possibility of large-scale coding.

Research Question: How can researchers integrate LLMs into qualitative coding while preserving hermeneutic depth and rigor?

Contribution:

- A hybrid workflow retaining human-led codebook development with an iterative step to adapt descriptions for LLM comprehension
- Demonstration that prompt design (chain-of-thought, one code per prompt) substantially improves coding fidelity
- Case study evidence that GPT-4 achieves human-equivalent intercoder reliability on most codes in a complex humanistic task

Background

Previous automation approaches fall short:

Supervised ML requires large annotated datasets and cannot use abstract code descriptions

Unsupervised ML (e.g., LDA) rarely captures researcher-intended categories

LLMs differ fundamentally:

Interpret tasks and code descriptions specified in natural language, no training data needed

Recent studies show promising but mixed results; existing guides lack concrete demonstration or grounding in qualitative traditions (Gilardi et al., 2023; Törnberg, 2023; Chew et al., 2023)

The Hybrid Workflow

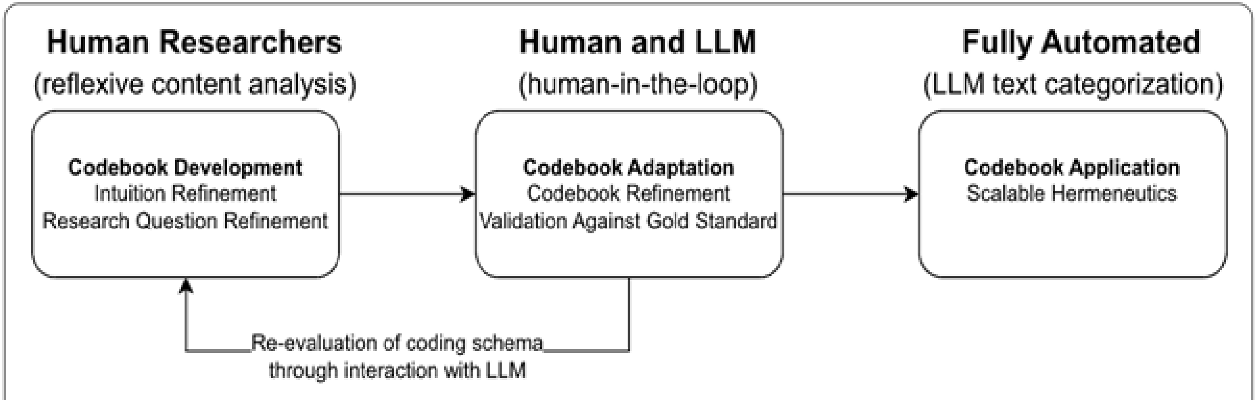
Phase 1 — Reflexive Content Analysis

- Exploratory reading, iterative refinement, intercoder reliability testing, consensus resolution

Phase 2 — Human-in-the-loop

- Adapt code descriptions for LLM comprehension through iterative testing; Validate against human-derived gold standard
- Deploy at scale only after acceptable fidelity is achieved

The LLM applies codes only. It does not participate in codebook development, theory building, or research design.



Hermeneutic Trade-offs

Preserved: Human-led codebook refinement; adapting descriptions for LLM adds a second layer of interpretive engagement

Lost: Data intimacy from manual coding; collaborative negotiation between human coders

Gained: Scalability to thousands of passages; increased statistical power; clearer code definitions through the adaptation process

Case Study – W.E.B. Du Bois in the News

New York Times articles referencing Du Bois, 1970–2023

232 passages, avg. 94 words each; 9 codes in 3 categories probing Du Bois's legacy across scholarship, activism, and public memory

Gold standard: 111 passages coded by two human coders, disagreements resolved by consensus

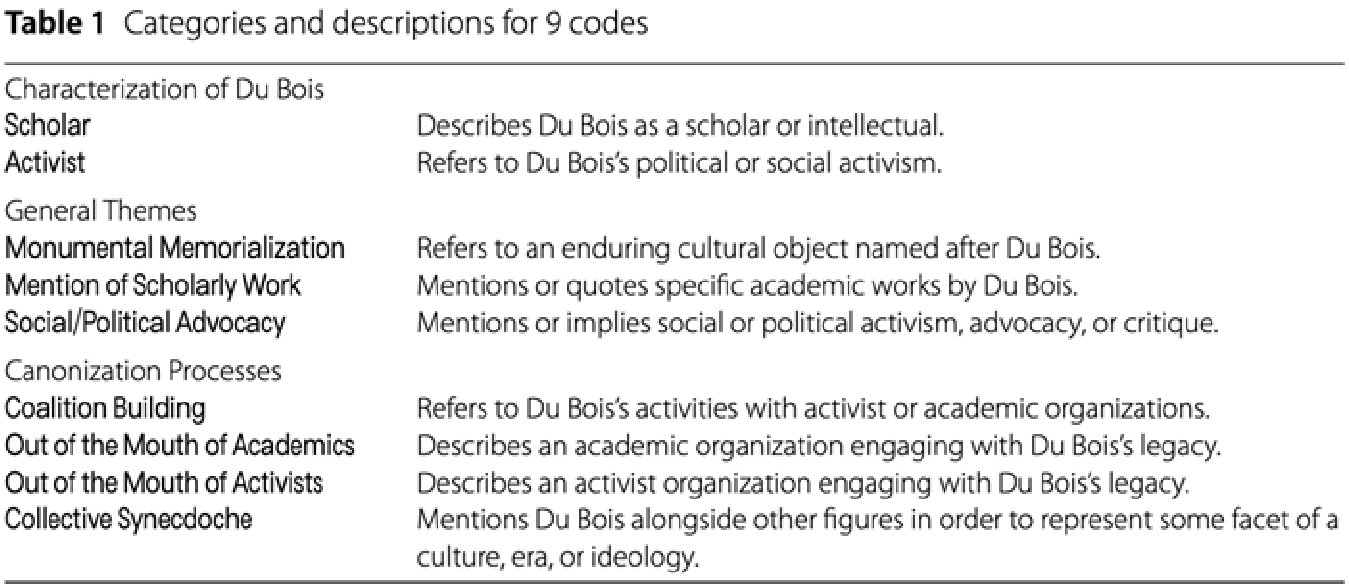


Table 1 Categories and descriptions for 9 codes

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.

What Is the LLM Used For?

Does:

- Read each passage, make binary judgment (applies / does not apply) for each code
- Provide written rationale for each judgment through **chain-of-thought** prompting (the model explains its reasoning before giving a final decision)

<u>The rationales serve two functions:</u>

- Improve coding accuracy by forcing the model to reason through the decision
- Allow researchers to diagnose why the model makes mistakes, revealing ambiguities in code descriptions that metrics alone would miss

Does not:

- Develop or modify the codebook
- Generate research questions
- Negotiate ambiguous cases

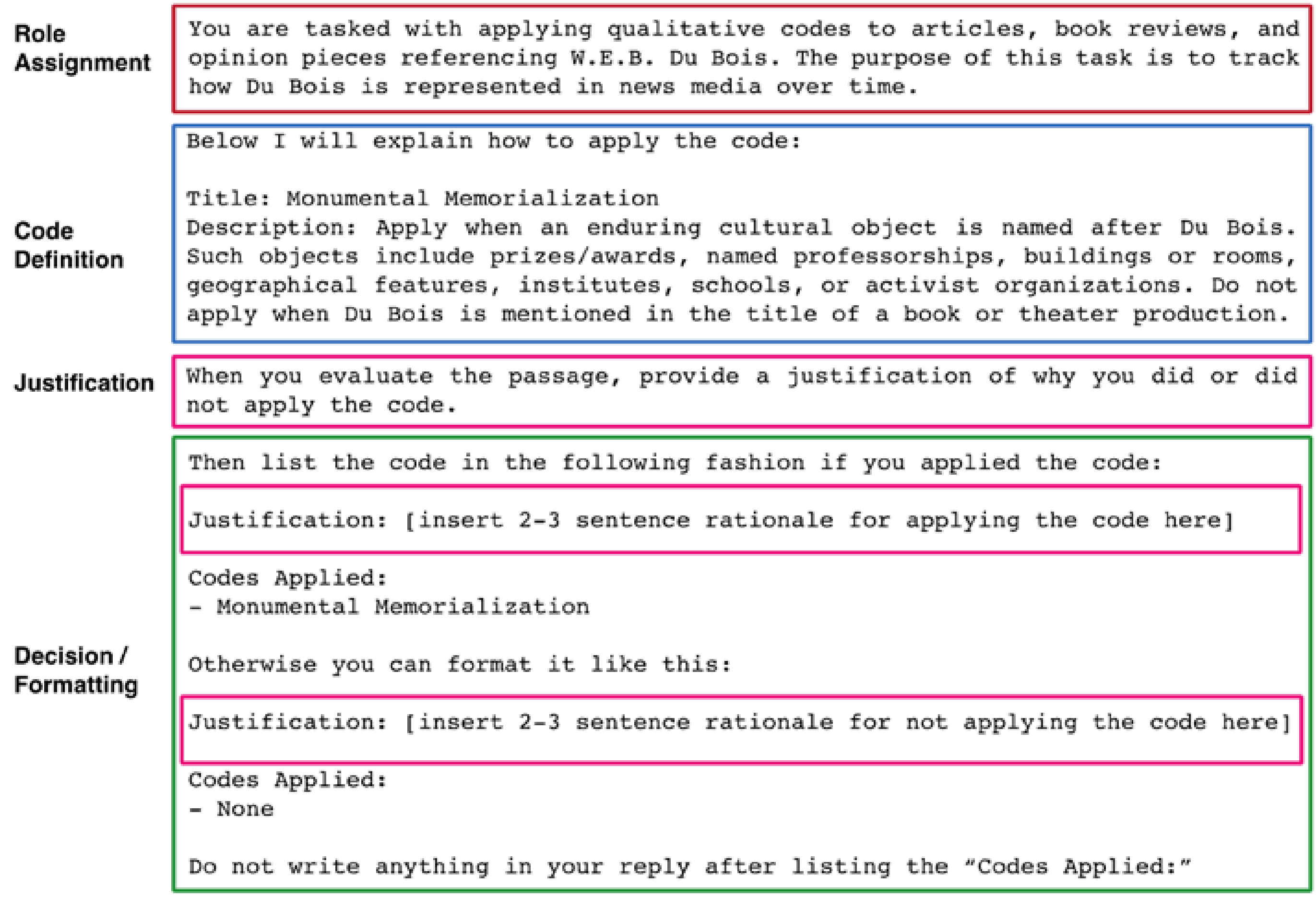
Prom

Chain-

1. Ro er
2. Co
3. Jus ng
4. De ormat

Design

tempe



Role Assignment
Code Definition
Justification
Decision / Formatting

```
You are tasked with applying qualitative codes to articles, book reviews, and
opinion pieces referencing W.E.B. Du Bois. The purpose of this task is to track
how Du Bois is represented in news media over time.
```

```
Below I will explain how to apply the code:

Title: Monumental Memorialization
Description: Apply 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. Do not
apply when Du Bois is mentioned in the title of a book or theater production.
```

```
When you evaluate the passage, provide a justification of why you did or did
not apply the code.
```

```
Then list the code in the following fashion if you applied the code:

Justification: [insert 2-3 sentence rationale for applying the code here]

Codes Applied:
- Monumental Memorialization

Otherwise you can format it like this:

Justification: [insert 2-3 sentence rationale for not applying the code here]

Codes Applied:
- None

Do not write anything in your reply after listing the "Codes Applied:"
```

Figure 2 The chain-of-thought prompt sequence

Adapting the Codebook for LLMs

Human coders bring implicit knowledge from discussions and shared context. LLMs rely entirely on what is written.

Word choice: Renaming "Academic Repute" to "Out of the Mouth of Academics" fixed systematic misinterpretation, even with the same description

Scope control: Adding "clearly implied through context" or "explicitly noted" to calibrate whether the LLM interprets narrowly or broadly

Instruction placement: Constraints placed earlier are more likely followed; positive instructions ("do") outperform prohibitions ("do not")

How Do the Authors Trust the Results?

Gold standard: 111 passages independently coded by two humans, disagreements resolved by consensus

Metrics: Cohen's κ as primary measure; AC1 can inflate scores for rare codes

Qualitative diagnosis: Reading LLM justifications to catch systematic errors that metrics miss

Overfitting safeguard: Train/test split recommended when iterating on the same data

Key Findings

Model performance (GPT-4, Per Code + CoT):

- 8 of 9 codes exceed κ = 0.6 (substantial agreement); 3 reach human-equivalent (κ = 0.79–1.00)
- The one poorly performing code (κ = 0.30) is also the hardest for human coders
- Codebook adaptation made a clear difference: e.g., Activist improved from κ = 0.48 to 0.81 after rewriting

Prompting choices matter:

- Chain-of-thought: mean κ = 0.68 with vs. 0.59 without
- Per Code vs. Full Codebook: mean κ = 0.68 vs. 0.60
- GPT-4 vs. GPT-3.5: mean κ = 0.68 vs. 0.34; the difference is qualitative, not just quantitative

Takeaways

Codebook adaptation: Make all implicit assumptions explicit; single word changes can shift performance substantially; the process deepens researcher understanding

Prompting: Chain-of-thought and per-code prompting consistently help; use "explicit"/"implicit" to calibrate scope; instruction order matters

Validation: Always validate against human gold standard; read justifications beyond metrics; re-validate when switching model versions

Appropriate use: Best for large datasets; random noise is tolerable at scale but systematic bias is not; poor-performing codes should be flagged for manual review

Thanks!