High Scores, Hidden Failures: Evaluating Codebook-Based LLM Content Analysis

Anonymous ACL submission

Abstract

Large language models (LLMs) are increasingly used for codebook-based content analysis in computational social science, where codebooks translate theoretical concepts into labeling schemes. While prior work reports strong predictive performance, it remains unclear whether these systems produce reliable and robust annotations. We conduct a large-scale empirical failure-mode analysis of LLM-based coding pipelines across three datasets and five models by systematically evaluating their stability, robustness, and conceptual validity under controlled perturbations, domain shifts, and optimization procedures. We uncover multiple failures that are largely invisible under standard metrics, showing that high aggregate scores frequently coexist with severe label-level disparities, instability across runs, domain sensitivity, and shifts in label interpretation. Importantly, these failure modes persist under widely used optimization strategies and remain largely invisible under standard metrics such as accuracy and micro-F1. Our results demonstrate that current LLM-based coding pipelines cannot be treated as stable measurement tools without targeted validation beyond aggregate performance.

1 Introduction

Large language models (LLMs) are increasingly used to annotate text data using natural language instructions, enabling scalable labeling for tasks such as dataset construction, qualitative coding, and content analysis (Törnberg, 2024; Abraham et al., 2025; Rytting et al., 2023; Xiao et al., 2023). In computational social science (CSS), these instructions are typically formalized as codebooks, structured annotation guidelines that map theoretical concepts onto predefined labels. By incorporating the codebook's labels and their definitions directly into prompts, recent studies report strong predictive performance and substantial agreement between human and LLM annotations, suggesting that LLMs could serve as scalable substitutes for human coders (Rytting et al., 2023; Ramanathan et al., 2025; Ruckdeschel, 2025; Farjam et al., 2025).

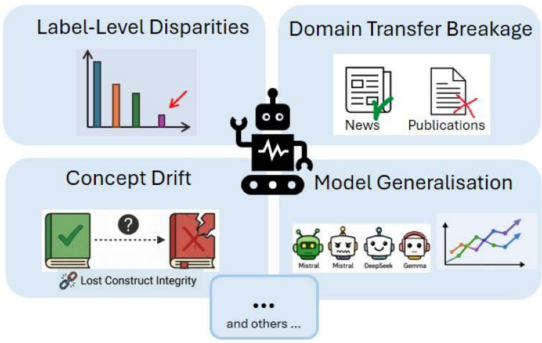

Figure 1: Illustrative examples of LLM failure modes in inference and optimization.

However, the use of LLMs for codebook-based annotation introduces two sources of potential failure. First, LLMs may fail to apply codebook instructions consistently during inference. A growing body of NLP research highlights prompt sensitivity (Shu et al., 2024; Mizrahi et al., 2024; Gu et al., 2023), where model predictions change under seemingly minor prompt variations. LLM-based coding pipelines may also exhibit broader robustness issues, including instability across repeated runs, inconsistent adherence to codebook definitions, and large disparities in performance across individual labels. Second, LLMs are increasingly used not only to apply annotation instructions but also to refine or optimize the prompts themselves. Recent approaches employ automatic prompt optimization or multi-agent prompting frameworks to iteratively modify prompts in order to improve predictive performance (Abraham et al., 2025; Zhao et al., 2025). While these methods can yield improvements on standard evaluation metrics, they also introduce additional risks. For instance, mod-

ifying prompts that encode coding schemes can implicitly alter label definitions, amplify disparities in label-level performance, or shift the conceptual meaning of the constructs being measured (see Section 5). These issues are consequential in content analysis, where annotations function as measurements used in downstream statistical analyses (e.g., frequency comparisons, regressions, hypothesis testing) and for qualitative interpretation, including conceptual and relational analysis, where measurement validity and replicability are critical.

In this paper, we operationalize the risks around faithful LLM coding by defining controlled experiments and complementary metrics. Then, we empirically demonstrate that current LLM-based codebook methods systematically fail in predictable ways. We organize our study around a set of *failure modes*, spanning two stages of practical CSS pipelines: (1) **inference failure modes**, which arise when an LLM applies a fixed codebook (see Section 4), and (2) **optimization-induced failure modes**, which arise when an LLM is used to refine a codebook prior to inference (see Section 5). Our main contributions are:

- We provide a framework for systematic failure-mode analysis for LLM content analysis, identifying predictable breakdowns at both the inference and optimization stages.
- We demonstrate across three datasets and five models, that commonly reported performance metrics misrepresent coding reliability as they mask these failures.
- We distill our findings into a minimal validation protocol (see Section 6), and release our new dataset, code and evaluation pipeline to support reproducibility.

2 Related Work

2.1 Reliability of LLM-Based Text Annotation

In CSS, LLMs are increasingly used to generate annotations for downstream analyses, such as estimating prevalence or testing hypotheses (Gligoric et al., 2025; Baumann et al., 2025). It has been shown that LLM annotations vary depending on the design of the prompt, the domain, the task, or the choice of model (Aldeen et al., 2023; Kocoń et al., 2023). Therefore, naive use can produce unreliable statistical estimates (Gligoric et al., 2025) and propagate to downstream statistical inference (Baumann et al., 2025).

Within content analysis, research has increasingly explored how LLMs can leverage codebooks in zero-shot or few-shot coding scenarios (Rytting et al., 2023; Dunivin, 2025; Farjam et al., 2025). Prior work shows that incorporating codebook definitions or examples can improve classification performance (Ruckdeschel, 2025; Xiao et al., 2023). Other studies highlight that prompt design choices and codebook components (e.g., "Negative Clarification" or "Positive Example") can influence model predictions (Halterman and Keith, 2025). Some work also reports variability in LLM coding results across repeated runs despite strong overall agreement with human annotations (Ramanathan et al., 2025), suggesting that stability remains an underexplored dimension of evaluation. However, existing studies primarily focus on aggregate performance metrics (Ruckdeschel, 2025; Xiao et al., 2023; Rytting et al., 2023; Dunivin, 2025; Farjam et al., 2025), leaving open questions about robustness, label-level disparities, and other potential failure modes when LLMs apply codebooks.

2.2 Codebook Optimization with LLMs

A complementary line of research explores how LLMs themselves can be used to optimize codebook-prompts for LLM inference. Abraham et al. (2025) propose Automatic Prompt Optimization (APO), starting with a human-labeled subset and an initial task prompt. The LLM repeatedly generates semantically equivalent variations, evaluates them on the labeled data, and retains the best-performing prompts. Additionally, Zhao et al. (2025) introduce the multi-agent framework SCALE. SCALE simulates collaborative coding processes, including discussion and evolving codebooks. In our study, we apply APO and SCALE analyzing their robustness, comparability, and impact on the quality of codebooks. While prompt optimization techniques focus on improving predictive performance, their effects on robustness, and conceptual validity remain unexplored.

3 Data and Evaluation Setup

Datasets To address robustness, we evaluate across three datasets varying in domain, size, and label count (Table 1; codebooks in Appendix A). We include the CCC dataset (Halterman and Keith, 2025) (on protests in the United States) and MINIMUM WAGE dataset (Ruckdeschel and Wiedemann, 2023) (on the impact of minimum wage policies)

Dataset	Multi-Label	#Labels	#Train	#Val	#Test
CCC	×	4	4,706	1,007	1,010
MINIMUM WAGE	✓	12	758	194	194
VALUE CREATION SCIENCE	✓	8	2,149	268	269
VALUE CREATION NEWS	✓	8	–	–	60

Table 1: Dataset statistics. ML indicates a a multi-label dataset. Data split size is in number of documents.

from prior work to ensure comparability. In addition, we introduce and release two datasets on Value Creation across distinct sources: the VALUE CREATION SCIENCE dataset, derived from scientific literature, and the VALUE CREATION NEWS, compiled from news-wire corpora. The former is the primary dataset and the latter was developed to evaluate robustness under domain shift. The related codebook captures concepts such as actor, demand, and resource, and was developed by a team of social scientists through an iterative annotation process, with refinements guided by inter-annotator agreement (Fleiss' κ). Two domain experts further provide qualitative validation for this present study. Additional details on data sources, label sets, and annotation procedures are provided in Appendix B.

Models and Evaluation Unless otherwise noted, inference experiments use Mistral-7B-Instruct-v0.3 (Jiang et al., 2023) at temperature 0.3, an open and relatively compact model accessible to a wide range of researchers. Cross-model comparisons appear in Section 4.2. To ensure comparability with the original SCALE setup, we run all three methods with GPT-4o. We report *accuracy*, *micro-F1*, *macro-F1*, *macro-Precision*, and *Worst-Label-F1* to surface disparities masked by aggregate metrics. For codebook-optimization approaches, we additionally report an expert Likert scale evaluation of the resulting codebook quality on VALUE CREATION SCIENCE.

4 Failure Modes in Inference

In the following subsections, we present the results of the inference experiments structured along their respective identified failure modes. Starting from an *Unchanged* codebook baseline (Table 2), we examine issues such as label-level disparities and overinterpretation. We further probe reliability through prompt diagnostics, evaluate context sensitivity via formatting and dataset variations, measure stochastic stability with repeated runs, test domain transfer on a new corpus, and assess model generalization across five LLMs.

4.1 Reliability of Label Application

Label-Level Disparities Standard metrics such as *accuracy* and *micro-F1* aggregate across labels and can therefore mask severe per-category failures, a problem compounded by the highly imbalanced datasets typical of CSS. Balanced performance is essential for valid downstream interpretation. For example, in a simple frequency analysis of how often a concept appears in a corpus, systematic underperformance on certain labels directly biases results. Moreover, robust generalization to datasets with different label distributions requires models to perform reliably across all categories. Our experiments expose this pattern (Table 2). In MINIMUM WAGE, the label YOUTH-AND-SECONDARY-WAGE-EARNERS (support = 5) achieves the lowest F1 (28.90), while SOCIAL-JUSTICE/INJUSTICE (support = 59) reaches 72.97. More strikingly, the gap between *worst-label-F1* and *micro-F1* is 30 points for MINIMUM WAGE, 26 points for VALUE CREATION SCIENCE, and 28 points for CCC, which are differences entirely invisible in aggregate reporting. The consistent gap between *macro-F1* and *micro-F1* across all experiments further confirms uneven performance.

Overinterpretation Ruckdeschel (2025) identify overinterpretation as a recurring issue in LLM-based coding: models infer meaning beyond what is explicitly stated. Our quantitative results confirm this tendency. In two of the three datasets, *macro-Precision* falls below *macro-F1* (Table 2), indicating that models assign labels more liberally than the gold standard warrants. At the label level, precision lags behind recall in 17 of 24 labels, pointing to systematic overprediction rather than isolated errors. For CSS workflows that rely on label frequencies or co-occurrence counts, such overcoding inflates the apparent prevalence of concepts in a corpus and can distort substantive conclusions.

Codebook-Specific Label Meanings A central assumption of codebook-based content analysis is that the model follows the provided definitions rather than defaulting to background knowledge acquired during pretraining. Ziems et al. (2024) argue that zero-shot performance is strongest when a model's internal concept overlaps with the intended construct (e.g., stance or emotion detection), but declines on tasks requiring compliance with detailed instructions. Halterman and Keith (2025) make this issue explicit with two diagnostics: *Generic Label*

Codebook Variant	MINIMUM WAGE					CCC					VALUE CREATION SCIENCE				
	Acc	mi. F1	ma. F1	ma. P	WLF1	Acc	mi. F1	ma. F1	ma. P	WLF1	Acc	mi. F1	ma. F1	ma. P	WLF1
Unchanged (§4.1, §4.1)	$18.2^{19.9}_{16.8}$	$59.1^{59.9}_{58.2}$	$54.8^{56.4}_{53.2}$	$47.5^{49.2}_{45.8}$	$28.9^{38.7}_{19.1}$	$33.0^{33.2}_{32.7}$	$34.0^{34.3}_{33.8}$	$33.4^{33.7}_{33.1}$	$45.6^{45.8}_{45.4}$	$6.4^{6.5}_{6.4}$	$1.9^{2.1}_{1.6}$	$44.9^{45.1}_{44.6}$	$39.6^{40.1}_{39.0}$	$30.5^{31.0}_{29.9}$	$18.8^{21.7}_{15.9}$
Generic Label (§4.1)	$\mathbf{23.4^{24.9}_{22.0}}$	$\mathbf{55.7^{56.2}_{55.1}}$	$\mathbf{49.5^{50.6}_{48.5}}$	$\mathbf{39.7^{41.0}_{38.4}}$	$32.2^{39.9}_{24.5}$	$\mathbf{42.9^{43.1}_{42.6}}$	$\mathbf{43.8^{44.1}_{43.5}}$	$\mathbf{27.8^{28.0}_{27.6}}$	$45.8^{46.5}_{45.1}$	$\mathbf{5.8^{6.0}_{5.7}}$	$\mathbf{1.0^{1.2}_{0.7}}$	$\mathbf{41.6^{42.1}_{41.2}}$	$\mathbf{38.7^{39.2}_{38.1}}$	$\mathbf{27.5^{27.9}_{27.1}}$	$19.3^{21.4}_{17.3}$
Swapped Label (§4.1)	$\mathbf{2.9^{3.4}_{2.4}}$	$\mathbf{37.7^{38.5}_{37.0}}$	$\mathbf{29.9^{31.2}_{28.6}}$	$\mathbf{25.2^{26.4}_{24.0}}$	$\mathbf{1.4^{4.7}_{0.0}}$	$\mathbf{12.3^{12.4}_{12.2}}$	$\mathbf{12.5^{12.7}_{12.4}}$	$\mathbf{10.6^{10.8}_{10.3}}$	$\mathbf{17.4^{17.6}_{17.1}}$	$\mathbf{2.5^{2.6}_{2.3}}$	$2.0^{2.2}_{1.8}$	$\mathbf{39.9^{40.4}_{39.4}}$	$\mathbf{35.1^{35.7}_{34.6}}$	$\mathbf{28.5^{28.9}_{28.0}}$	$20.6^{21.6}_{19.6}$
Label-Only (§4.1)	$\mathbf{6.9^{7.3}_{6.4}}$	$\mathbf{45.7^{45.9}_{45.5}}$	$\mathbf{42.1^{42.4}_{41.8}}$	$\mathbf{36.2^{36.5}_{35.9}}$	$\mathbf{14.2^{15.1}_{13.3}}$	$\mathbf{15.9^{16.1}_{15.6}}$	$\mathbf{16.8^{17.0}_{16.6}}$	$\mathbf{24.0^{24.7}_{23.7}}$	$\mathbf{46.5^{46.9}_{46.0}}$	$\mathbf{5.8^{5.8}_{5.8}}$	$\mathbf{0.6^{0.8}_{0.4}}$	$\mathbf{43.5^{43.8}_{43.2}}$	$\mathbf{38.4^{38.8}_{37.9}}$	$\mathbf{29.2^{29.6}_{28.8}}$	$\mathbf{24.2^{26.4}_{22.0}}$
Simple Codebook (§4.2.i)	$\mathbf{21.6^{22.1}_{21.1}}$	$\mathbf{61.2^{61.7}_{60.7}}$	$\mathbf{56.8^{57.8}_{55.9}}$	$48.9^{49.9}_{47.9}$	$\mathbf{39.1^{40.1}_{38.0}}$	$\mathbf{30.0^{27.2}_{26.7}}$	$\mathbf{28.7^{28.9}_{28.4}}$	$\mathbf{31.2^{31.4}_{31.0}}$	$45.6^{45.8}_{45.5}$	$\mathbf{6.1^{5.3}_{5.9}}$	$1.7^{2.0}_{1.5}$	$\mathbf{42.5^{42.8}_{42.3}}$	$\mathbf{35.9^{36.3}_{35.5}}$	$\mathbf{26.7^{27.2}_{26.2}}$	$\mathbf{10.9^{13.3}_{8.6}}$
Structured (§4.2.ii)	$\mathbf{21.0^{22.8}_{19.2}}$	$\mathbf{56.8^{57.7}_{55.8}}$	$\mathbf{50.9^{52.4}_{49.5}}$	$\mathbf{43.3^{45.0}_{41.7}}$	$25.8^{36.7}_{14.9}$	$\mathbf{49.3^{49.5}_{49.1}}$	$\mathbf{50.2^{50.3}_{50.0}}$	$\mathbf{38.6^{38.8}_{38.5}}$	$\mathbf{47.4^{47.6}_{47.2}}$	$\mathbf{7.0^{7.2}_{6.8}}$	$1.9^{2.2}_{1.5}$	$\mathbf{44.3^{44.7}_{44.0}}$	$\mathbf{38.1^{38.6}_{37.7}}$	$29.8^{30.6}_{29.0}$	$\mathbf{12.1^{14.1}_{10.0}}$
Majority Order (§4.2.iii)	$\mathbf{20.7^{21.9}_{19.5}}$	$60.2^{61.1}_{59.2}$	$54.7^{56.2}_{53.1}$	$45.7^{47.4}_{44.0}$	$36.4^{41.0}_{31.8}$	$\mathbf{29.0^{29.2}_{28.9}}$	$\mathbf{30.1^{30.3}_{30.0}}$	$\mathbf{32.0^{32.1}_{31.8}}$	$\mathbf{46.7^{47.0}_{46.4}}$	$\mathbf{6.2^{6.4}_{6.1}}$	$\mathbf{0.4^{0.6}_{0.3}}$	$\mathbf{43.6^{43.9}_{43.3}}$	$40.2^{40.6}_{39.8}$	$29.9^{30.2}_{29.6}$	$21.8^{22.5}_{21.0}$
Inverted Order (§4.2.iii)	$\mathbf{13.1^{14.0}_{12.2}}$	$\mathbf{55.9^{56.4}_{55.4}}$	$\mathbf{49.8^{50.3}_{49.4}}$	$\mathbf{42.3^{42.7}_{41.8}}$	$31.8^{32.7}_{30.9}$	$\mathbf{57.7^{57.9}_{57.5}}$	$\mathbf{58.7^{58.9}_{58.6}}$	$\mathbf{45.4^{45.5}_{45.2}}$	$\mathbf{47.0^{47.1}_{46.8}}$	$\mathbf{9.0^{9.5}_{8.5}}$	$\mathbf{0.7^{1.0}_{0.4}}$	$\mathbf{36.2^{36.6}_{35.8}}$	$\mathbf{33.3^{33.8}_{32.7}}$	$\mathbf{33.4^{34.3}_{32.5}}$	$16.3^{17.6}_{14.9}$

Table 2: LLM-Inference results with baseline Unchanged Codebook vs. other codebook variants. We report accuracy (Acc), micro-F1 (mi. F1), macro-F1 (ma. F1), macro precision (ma. P), and worst-label-F1 (WLF1). Accuracy measures the fraction of instances where every label of that instance is predicted correctly, leading to lower scores for MINIMUM WAGE and VALUE CREATION SCIENCE. For all metrics, we show their mean value and the 95% confidence interval over 10 runs. Bold values indicate statistically significant differences over baseline ($p < 0.05$) under paired t-tests. Underlined values indicate significance under paired bootstrap tests. Codebook variant numbers (e.g., §4.1) indicate where each variant is discussed.

Accuracy (replacing labels with pseudo-labels such as "Label 1") and *Swapped Label* Accuracy (permuting names and definitions). Building on this work, we introduce a third diagnostic, *Label-Only*, where inference uses only label names. The results (Table 2) reveal a consistent pattern across datasets: *Label-Only* achieves surprisingly high scores (Table 2), especially for VALUE CREATION SCIENCE, suggesting that models lean heavily on label names. In contrast, performance drops substantially under *Generic Label* and *Swapped Label*, sometimes more than in the *Label-Only* case, confirming reliance on label name over definitions.

4.2 Robustness

Context Sensitivity Prior work offers various, often conflicting guidelines for prompt design in LLM-based content analysis. We isolate three such recommendations, applying each as a single modification to the *Unchanged* codebook baseline (Table 2).

Ruckdeschel (2025) advocates for shorter, concise codebooks, while Halterman and Keith (2025) suggest using full codebooks for better overall performance. In our experiments, we use labels and their definitions as simplified codebooks, removing examples, clarifications and counter-examples (*i. Simple* codebook, Table 2). We see that simplified codebooks degrade both *micro/macro-F1* for VALUE CREATION SCIENCE and CCC datasets, but improve overall performance on MINIMUM WAGE marginally.

Halterman and Keith (2025) recommend explicit structural markers, e.g. prefixing each field with "definition:" or "example:". In our experiments, when using their structured codebook style (*ii. Structured*, Table 2), CCC shows substantial gains across nearly all metrics, while MINIMUM WAGE and VALUE CREATION SCIENCE exhibit significant decreases in all metrics. We attribute this outcome to the average length of the texts in CCC, which is by far the longest of the three. Information signaled by markers ("definition") may help the model retrieve relevant information from the full context window.

Dunivin (2025) recommends placing the majority labels at the beginning of the codebook-prompt (*iii. Majority Order*). As shown in Table 2, this ordering yields only minor changes for MINIMUM WAGE and VALUE CREATION SCIENCE, but substantially degrades performance on CCC. Strikingly, CCC achieves its best *micro/macro-F1* when the majority label order is inverted (*iii. Inverted Order*), though this harms performance on the other two datasets.

These contradictory outcomes across three independent design dimensions help to explain why different studies, most of which evaluate a single dataset, arrive at opposing best-practice recommendations.

Stochastic Instability As Ramanathan et al. (2025) note, LLM (GPT-4) performance can vary across repeated runs. We observe the same effect with open-weight models when setting temperature above zero and not fixing a random seed. Figure 2 shows the aggregated mean F1 scores over 15 such runs. The degree of instability is dataset- and label-dependent (Figure 2). For CCC, scores remain highly stable. VALUE CREATION SCIENCE shows moderate run-to-run differences that are reduced by averaging three runs, while MINIMUM WAGE exhibits stronger variation, particularly for the low-support label TURNOVER. The pattern is con-

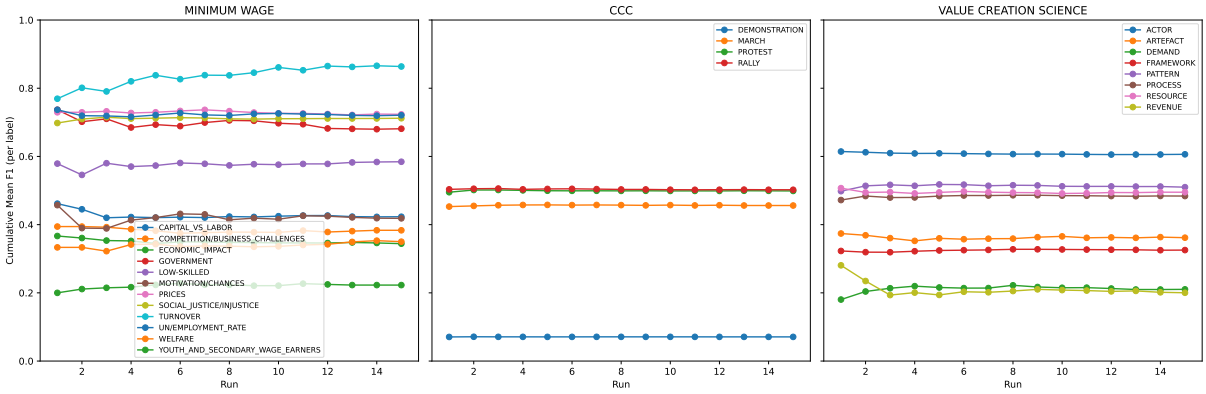

Figure 2: Aggregated mean F_1 scores over 15 runs, illustrating the stabilizing effect of averaging multiple runs for each label.

sistent with the label-level disparities documented in Section 4.1: categories with small support are most vulnerable to both systematic underperformance and stochastic noise. Our observed variance is modest compared to the greater instability reported by Ramanathan et al. (2025), likely because open-weight models allow transparent control over sampling parameters. Closed-source APIs, by contrast, may introduce opaque configuration changes between provider updates, compounding run-to-run variability with silent shifts in model behaviour.

Model Generalization We compare GPT-4o-mini, DeepSeek-V3 (DeepSeek-AI et al., 2025), Gemma-2-27B-it (Team et al., 2024), Llama-3.1-8B-Instruct, and Mistral-7B-Instruct-v0.3 (Jiang et al., 2023) using the *Unchanged* codebook on all three datasets (Figure 3). No model dominates consistently. For the VALUE CREATION SCIENCE dataset, performance levels are relatively close: Gemma achieves the highest precision, while DeepSeek leads in *micro/macro-F1*. Llama, in contrast, performs weakest overall. On the CCC dataset, Gemma's F1 scores drop drastically, up to 25 percentage points below the other models, whereas the compact Mistral performs comparably to the much larger DeepSeek and GPT models. The pattern reverses for the MINIMUM WAGE dataset, where Gemma consistently outperforms all others, while Mistral and Llama lag behind. The absence of a stable model ranking across datasets poses a systemic threat for CSS. If conclusions depend heavily on which model (or version) is chosen, downstream analyses risk being contingent on arbitrary or even opaque implementation details. Closed-source models amplify this problem, as providers update them without notice, leading

	Acc	mi. F1	macro-F1	ma. P	WLF1
VALUE CREATION SCIENCE	$1.9^{2.1}_{1.6}$	$44.9^{45.1}_{44.6}$	$39.6^{40.1}_{39.0}$	$30.5^{31.0}_{29.9}$	$18.8^{21.7}_{15.9}$
VALUE CREATION NEWS	$\underline{\mathbf{0.0^{0.1}_{0.0}}}$	$\underline{\mathbf{62.2^{62.8}_{61.7}}}$	$\underline{\mathbf{59.6^{60.2}_{58.9}}}$	$\underline{\mathbf{54.1^{54.7}_{53.5}}}$	$\underline{\mathbf{32.9^{33.5}_{32.2}}}$

Table 3: Metrics over ten runs for VALUE CREATION SCIENCE and VALUE CREATION NEWS, shown with their mean value and the 95% confidence interval over 10 runs. Bold values indicate statistically significant differences under paired t-tests ($p < 0.05$). Underlined values indicate significance under paired bootstrap tests.

to silent shifts in performance and prompt sensitivity. What appears to be a stable, replicable pipeline today may yield different results, undermining the credibility of published findings.

Domain Transfer Failure A critical assumption in codebook-based content analysis is that coding frameworks remain valid across new data sources within the same theoretical domain. We test this assumption by applying the VALUE CREATION SCIENCE codebook, developed on academic publications, to news articles covering the same business and economics topics. Both corpora share the underlying constructs but differ in style, vocabulary, and communicative conventions. The results expose a paradox (Table 3). On one side, the per-label performance improves substantially in the news domain: *micro-F1* rises from 44.89 to 62.22 and *macro-F1* from 39.57 to 59.55, likely because news articles use simpler language and more direct formulations (qualitative examples in Appendix C). On the other side, this confidence does not extend to a holistic understanding of the coding task: the model fails to predict the full correct label set for any single instance, resulting in an overall *accuracy* of 0.0. Such domain sensitivity poses significant risks for comparative analyses and theory testing

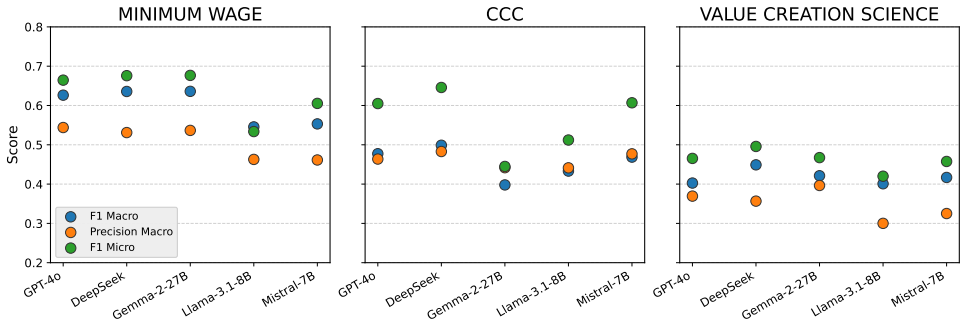

Figure 3: An overview of model performance for the different datasets.

in CSS. A codebook that performs well on one text type may yield systematically different results on another, even when both sources address the same substantive questions. These inconsistencies can lead to spurious empirical patterns or failed replications.

5 Failure Modes in Optimization

Failure modes in LLM-based coding do not arise only during inference, additional risks emerge when LLMs are used to optimize codebooks that are later applied during inference. To study these effects, we first implement targeted minimal optimizations, introducing small structured changes to the original codebook to test their impact across datasets (see Table 4). These perturbations are generated zero-shot in a single LLM rollout. Second, we evaluate real-world approaches, in which the complete codebook is optimized. Specifically, we apply (i) a zero-shot full-codebook improvement (*FULL*), prompting an LLM to refine the entire codebook; (ii) *APO* by Abraham et al. (2025); and (iii) the multi-agent framework *SCALE* (Zhao et al., 2025), both iterative prompt refinement methods. Using these setups, we analyse risks such as concept drift, bias amplification, and the practical feasibility of LLM-based codebook optimization.

5.1 Amplified Bias and Sensitivity

Increasing Label Bias The label-level disparities observed during inference (Section 4.1) are amplified when LLMs are used to optimize codebooks. Small modifications to definitions or examples can disproportionately shift performance across labels (see Table 4). When evaluation relies solely on aggregate metrics, such as *micro-F1*, there is a danger of endorsing adaptations that improve majority classes while harming minority ones. A concrete case illustrates the danger: adding positive and negative examples improves *micro/macro-F1*, *accuracy*, and *macro-Precision* on both CCC and VALUE CREATION SCIENCE, yet simultaneously drives *worst-label-F1* down to 5.67 and 11.32 respectively. Judged by aggregate metrics alone, these optimizations would be endorsed as beneficial. In practice, they systematically suppress minority categories, precisely the labels most vulnerable to begin with. The problem is compounded by automatic methods such as APO, which optimize for *micro-F1* by design and therefore have no mechanism to prevent minority-label degradation.

Dataset Dependency for Optimization The dataset sensitivity documented for inference (Section 4.2) intensifies when LLMs are used to optimize codebooks, since each refinement step compounds the interaction between prompt wording and data characteristics. The absence of universal optimization rules motivates automated methods such as APO and SCALE, yet these introduce systematic risks of their own.

Our experiments show that even minimal adjustments, like adding examples, inserting clarifications, or reformulating definitions produce inconsistent effects across datasets (Table 4). What improves performance in one setting often degrades it in another, casting doubt on the existence of generalizable strategies for LLM-based refinements. This underscores the need for dataset-specific optimization methods that adapt prompts automatically rather than relying on fixed heuristics.

Among real-world optimization methods, the simple FULL approach proves most stable, with *macro-F1* decreasing by at most 5 points across datasets (Table 4). APO is more volatile: it performs well on CCC, but for VALUE CREATION SCIENCE it removed one label entirely, leading to weaker results. SCALE also exhibits high variability, improving CCC performance but reducing

Optimization	MINIMUM WAGE					CCC					VALUE CREATION SCIENCE				
	Acc	mi. F1	ma. F1	ma. P	WLF1	Acc	mi. F1	ma. F1	ma. P	WLF1	Acc	mi. F1	ma. F1	ma. P	WLF1
Unchanged Codebook	17.5	59.8	54.4	47.1	25.0	38.3	40.9	38.1	44.4	7.1	0.7	44.5	39.4	29.8	18.2
Label Names (§ 5.1, § 5.1)	26.8	56.0	51.0	44.8	22.2	45.4	47.2	40.1	46.3	7.4	3.4	42.4	35.3	29.3	8.5
Label Definition (§ 5.1, § 5.1)	19.1	60.0	55.7	46.1	28.6	52.9	55.0	44.4	45.0	13.0	1.9	45.5	39.8	30.8	14.8
Pos Clarification (§ 5.1, § 5.1)	21.1	58.1	53.7	46.6	25.0	35.4	37.3	35.7	44.8	7.5	2.6	44.5	39.0	29.4	12.3
Pos + Neg Clarification (§ 5.1, § 5.1)	24.7	61.1	57.5	53.9	40.0	45.5	47.2	40.1	46.3	7.4	2.2	44.8	40.4	30.6	23.5
Pos Examples (§ 5.1, § 5.1)	13.4	58.5	55.8	47.9	38.7	31.3	31.6	32.4	52.1	5.1	2.6	45.8	40.0	31.3	15.1
Pos + Neg Examples (§ 5.1, § 5.1)	15.0	59.3	55.4	49.9	25.0	44.2	44.9	39.4	50.7	5.7	3.0	45.5	38.9	30.2	11.3
Full (§ 5.1, § 5.2)	18.6	58.9	52.7	44.5	33.3	30.9	32.2	33.4	47.3	6.4	3.4	43.3	38.5	28.8	13.2
APO (§ 5.1, § 5.2)	16.5	38.8	33.8	26.9	22.2	57.0	60.1	45.2	45.4	9.5	0.0	40.2	34.3	22.5	0.0
SCALE (§ 5.1, § 5.2)	17.0	39.7	33.6	22.9	9.3	49.5	59.9	44.0	42.2	5.6	0.0	38.1	36.7	23.7	26.2

Table 4: Results for different codebook adaptations across datasets. Each dataset reports accuracy (Acc), micro F1 (mi. F1), macro F1 (ma. F1), marco precision (ma. P), and worst label F1 (WLF1). For the structured experiment Mistral-7B-Instruct-v0.3 was used. GPT-4 was used for FULL, APO and SCALE experiments.

Method	Definition for the REVENUE label
ORIGINAL	The benefit or reward an actor receives for contributing to value creation. Can be tangible or intangible. *Examples:* wages, profits, royalties, increases in social capital.
FULL	The reward or advantage received by an actor for participating in value creation, either tangible or intangible. *Examples:* wages, profits, royalties, increases in social capital.
APO	The benefit or compensation received by an actor for their involvement in value creation. This can be either tangible or intangible. *Examples:* wages, profits, royalties, increases in social capital.
SCALE	The benefit or reward an actor receives for contributing to value creation. Can be tangible or intangible. *Examples:* wages, profits, royalties, increases in social capital, financial implications of operational changes, and benefits from educational initiatives. *Clarification:* Revenue can be seen as the return on investment for actors involved in the value creation process.

Table 5: Optimized versions of the label REVENUE across three approaches.

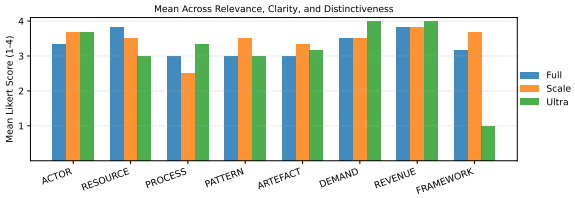

Figure 4: Overview of mean Likert scores across dimensions for each optimization method.

Dataset	Full	APO	SCALE
MINIMUM WAGE (§ 5.3)	1	113,200	856
CCC (§ 5.3)	1	113,200	688
VALUE CREATION SCIENCE (§ 5.3)	1	113,200	1,044

Table 6: Number of API calls required for each optimization method across datasets (§5.3).

most metrics for the other two datasets. For VALUE CREATION SCIENCE, repeated SCALE runs produced inconsistent outcomes ranging from leaving all labels unchanged to completely rewriting the codebook, resulting in labels that bore little resemblance to the originals. This run-level instability means that even the choice of which optimization run to deploy becomes a consequential, yet essentially arbitrary, decision.

5.2 Conceptual Validity of Optimized Codebooks

Optimization alters label definitions and occasionally label names, raising the question of whether the resulting codebooks still capture the theoretical constructs encoded in the original design. To evaluate this, two domain experts assessed each label description in the VALUE CREATION SCIENCE dataset using a 1–4 Likert scale (Lynn, 1986), a method commonly used to evaluate natural language generation outputs (Amidei et al., 2019). Due to low rating variance, inter-annotator agreement measured by Cohen's κ is relatively low ($\kappa = 0.34$), although the observed agreement between annotators is high: 51.6% exact agreement and 87.3% of ratings differing by at most one scale point.[1] Three dimensions were evaluated: *relevance*, *clarity*, *distinctiveness* (see Appendix E).

[1]Cohen's κ adjusts observed agreement by the agreement expected by chance. When rating variance is low (i.e., both annotators tend to use the same rating categories very frequently), the expected chance agreement becomes high. This reduces the resulting κ value even if the annotators largely agree in practice. In our case, both annotators consistently assigned high scores, and one rater frequently used the maximum score (4), which increases expected chance agreement and lowers κ.

Relevance For relevance, the expert evaluated whether each label still reflected the intended concept of the original codebook. The highest mean score was achieved by SCALE (3.56), followed by FULL (3.44) and APO (3.19) (Figure 5). The APO approach received consistently high ratings, except for the label *FRAMEWORK*, which was removed during optimization. Despite this, overall performance in Table 4 remains strong given that one label could not be detected.

Clarity Since codebooks are intended for downstream analysis, it is essential that optimized label descriptions remain interpretable and clear to social scientists. Both SCALE outperforms FULL and APO with mean clarity scores of 3.44 compared to 3.25 and 3.06. This gap may stem from the richer definitions and examples that help differentiate labels (see Table 5).

Distinctiveness For codebooks to be analytically useful, labels must remain clearly distinct from one another. Again, FULL and SCALE yield higher mean Likert scores APO.

For *clarity* and *relevance*, SCALE produces definitions experts judge superior to FULL (detailed scores in Appendix E. Yet FULL often achieves higher F1 scores (Table 4), revealing a disconnect: predictive gains can arise from simplified definitions that better align with model behavior while reducing conceptual interpretability. This tension highlights that quantitative metrics alone are insufficient for CSS applications where codebooks encode theoretical constructs.

5.3 Efficiency and Accessibility

Beyond performance, optimization approaches must also be feasible for researchers in practice. We therefore compare the efficiency of different methods by counting the number of API calls required for each approach (see Table 6). This provides an approximate measure of computational cost. FULL requires a single call per dataset. SCALE uses 688–1,044 calls depending on the dataset. APO requires 113,200 calls uniformly, over two orders of magnitude more than SCALE and five orders more than FULL, yet does not consistently outperform either alternative (Table 4). The cost-benefit profile of iterative optimization is thus far from guaranteed. Beyond cost, accessibility also matters: minimal prompt adjustments are within reach of any social scientist, whereas APO and SCALE demand non-trivial engineering skills.

6 Minimal Validation Protocol

To mitigate failure modes from Sections 4 and 5, we recommend reporting:

Minimal Validation Protocol

Tail Risk: Report *macro-F1*, *worst-label-F1*, per label scores and label supports alongside micro-F1/accuracy; include *macro-Precision* to detect overcoding.
Meaning Adherence: Test *Label-Only*, *Generic Label*, and *Swapped Label* to detect reliance on label-name semantics.
Stability: Repeat inference ≥ 3 times (more for opaque APIs) and report mean ± 95% CI; focus on low-support labels.
Sensitivity: Apply at least one formatting or label-order perturbation to reveal prompt-specific effects.
Transfer: Evaluate on an additional dataset from the same domain to detect shifts under domain or style changes.
Optimization Safeguards: When refining codebooks, include worst-label-F1 to protect minority labels, verify constructs with experts, and report computational cost.

7 Conclusion

LLMs enable large-scale content analysis in CSS but remain fragile and context-dependent. Our experiments demonstrate that high aggregate scores routinely coexist with severe label-level disparities, reliance on label-name semantics over codebook definitions, instability across runs, sensitivity to superficial formatting choices, and fragile domain transfer. When LLMs are additionally used to optimize codebooks, these problems do not diminish, they compound: optimization amplifies minority-label bias, produces dataset-contingent effects, and can silently alter the theoretical constructs a codebook is designed to measure. A recurring theme across all failure modes is that standard metrics are insufficient to detect them. To address this, we proposed a minimal validation protocol (Section 6) that operationalizes our findings into concrete checks, intended to be lightweight enough for routine adoption in CSS research.

Limitations

Our analysis is limited to a small set of datasets, codebooks, and model configurations. While these reflect common CSS research settings, results may not generalize across domains, languages, or annotation frameworks. Nonetheless, our evaluation framework is applicable to any dataset, codebook, or LLM, and we hope future work will extend it to larger, multilingual, and domain-specific contexts. Finally, the human evaluation of optimized codebooks relies on only two experts, which may limit intersubjective validity.

Ethical Considerations

This work studies the reliability of LLM-based text annotation and highlights potential risks when such annotations are used in downstream social science analyses. By identifying failure modes in inference and codebook optimization, our goal is to promote more transparent and responsible use of LLMs as measurement tools in research workflows.

All datasets used in this study are either publicly available or released by the authors with appropriate documentation. We do not introduce new personal or sensitive data, and our experiments focus on methodological evaluation rather than producing judgments about individuals or groups.

LLMs were used to assist with grammar, spelling, and refinement of the manuscript text. All technical content, analyses, and conclusions were developed and verified by the authors.

References

Louis Abraham, Charles Arnal, and Antoine Marie. 2025. Prompt selection matters: Enhancing text annotations for social sciences with large language models. *Preprint*, arXiv:2407.10645.

Mohammed Aldeen, Joshua Luo, Ashley Lian, Venus Zheng, Allen Hong, Preethika Yetukuri, and Long Cheng. 2023. Chatgpt vs. human annotators: A comprehensive analysis of chatgpt for text annotation. In *2023 International Conference on Machine Learning and Applications (ICMLA)*, pages 602–609.

Jacopo Amidei, Paul Piwek, and Alistair Willis. 2019. The use of rating and Likert scales in natural language generation human evaluation tasks: A review and some recommendations. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 397–402, Tokyo, Japan. Association for Computational Linguistics.

Joachim Baumann, Paul Röttger, Aleksandra Urman, Albert Wendsjö, Flor Miriam Plaza del Arco, Johannes B. Gruber, and Dirk Hovy. 2025. Large language model hacking: Quantifying the hidden risks of using llms for text annotation. *Preprint*, arXiv:2509.08825.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2025. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Zackary Okun Dunivin. 2025. Scaling hermeneutics: a guide to qualitative coding with llms for reflexive content analysis. *EPJ Data Science*, 14(1):28.

Mike Farjam, Hendrik Meyer, and Meike Lohkamp. 2025. A practical guide and case study on how to instruct llms for automated coding during content analysis. *Social Science Computer Review*.

Kristina Gligoric, Tijana Zrnic, Cinoo Lee, Emmanuel Candes, and Dan Jurafsky. 2025. Can unconfident LLM annotations be used for confident conclusions? In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3514–3533, Albuquerque, New Mexico. Association for Computational Linguistics.

Jiasheng Gu, Hongyu Zhao, Hanzi Xu, Liangyu Nie, Hongyuan Mei, and Wenpeng Yin. 2023. Robustness of learning from task instructions. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13935–13948, Toronto, Canada. Association for Computational Linguistics.

Andrew Halterman and Katherine A. Keith. 2025. Codebook llms: Evaluating llms as measurement tools for political science concepts. *Preprint*, arXiv:2407.10747.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Jan Kocoń, Igor Cichecki, Oliwier Kaszyca, Mateusz Kochanek, Dominika Szydło, Joanna Baran, Julita Bielaniewicz, Marcin Gruza, Arkadiusz Janz, Kamil Kanclerz, Anna Kocoń, Bartłomiej Koptyra, Wiktoria Mieleszczenko-Kowszewicz, Piotr Miłkowski, Marcin Oleksy, Maciej Piasecki, Łukasz Radliński, Konrad Wojtasik, Stanisław Woźniak, and Przemysław Kazienko. 2023. Chatgpt: Jack of all trades, master of none. *Information Fusion*, 99:101861.

Mary R. Lynn. 1986. Determination and quantification of content validity. *Nursing research*, 35 6:382–5.

Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. 2024. State of what art? a call for multi-prompt llm evaluation. *Transactions of the Association for Computational Linguistics*, 12:933–949.

Sriram Ramanathan, Lisa-Angelique Lim, Nazanin Rezazadeh Mottaghi, and Simon Buckingham Shum. 2025. When the prompt becomes the codebook: Grounded prompt engineering (groproe) and its application to belonging analytics. In *Proceedings of the 15th International Learning Analytics and Knowledge Conference*, LAK '25, page 713–725, New York, NY, USA. Association for Computing Machinery.

Mattes Ruckdeschel. 2025. Just read the codebook! make use of quality codebooks in zero-shot classification of multilabel frame datasets. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6317–6337, Abu Dhabi, UAE. Association for Computational Linguistics.

Mattes Ruckdeschel and Gregor Wiedemann. 2023. Argument aspect corpus.

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

Bangzhao Shu, Lechen Zhang, Minje Choi, Lavinia Dunagan, Lajanugen Logeswaran, Moontae Lee, Dallas Card, and David Jurgens. 2024. You don't need a personality test to know these models are unreliable: Assessing the reliability of large language models on psychometric instruments. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5263–5281, Mexico City, Mexico. Association for Computational Linguistics.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, and 179 others. 2024. Gemma 2: Improving open language models at a practical size. *Preprint*, arXiv:2408.00118.

Petter Törnberg. 2024. Best practices for text annotation with large language models. *Sociologica*, 18(2):67–85.

Ziang Xiao, Xingdi Yuan, Q. Vera Liao, Rania Abdelghani, and Pierre-Yves Oudeyer. 2023. Supporting qualitative analysis with large language models: Combining codebook with gpt-3 for deductive coding. In *Companion Proceedings of the 28th International Conference on Intelligent User Interfaces*, IUI '23 Companion, page 75–78, New York, NY, USA. Association for Computing Machinery.

Chengshuai Zhao, Zhen Tan, Chau-Wai Wong, Xinyan Zhao, Tianlong Chen, and Huan Liu. 2025. SCALE: Towards collaborative content analysis in social science with large language model agents and human intervention. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8473–8503, Vienna, Austria. Association for Computational Linguistics.

Caleb Ziems, William Held, Omar Shaikh, Jiaao Chen, Zhehao Zhang, and Diyi Yang. 2024. Can large language models transform computational social science? *Computational Linguistics*, 50(1):237–291.

A Codebooks

We retained the original codebooks as closely as possible and reformatted them into a standardized prompt format for annotation.

VALUE CREATION Codebook Prompt

The value creation codebook prompt:

You are an economic analyst examining the socio-technical structure of value creation systems. You have been handed the following codebook as instructions to code text units for their aspects of value creation.

Aspect Annotation in short text units.

The task is to find the aspect(s) present in a text unit. An aspect is defined as an element of the value creation system being discussed. It is possible that more than one aspect applies to a text unit; please mark all applicable aspects.

Below, the aspect categories are further explained. Each category has a label name and details defining or illustrating them.

[ACTOR]

- A person or group providing a resource to help create value. Can be individuals or organized/informal groups (e.g., communities, companies).
- Examples: manufacturers, investors, employees, innovators, user communities.

[RESOURCE]

- Something an actor uses to create value. Can be materials, objects processed, tools, information, or skills.
- Examples: machinery, raw materials, financial capital, data, technology, expertise.

[PROCESS]

- A series of actions where actors use resources to create value. Includes primary activities (making and delivering products) and supporting activities.

- Examples: manufacturing, procurement, distribution, maintenance, servicing.

[PATTERN]

- A regular and repeated way value creation is performed. Encompasses rules, norms, routines, and practices shaping actor roles or constraining actions.
- Examples: organizational structures, communication protocols, property rights, quality standards.

[ARTEFACT]

- The output of the value creation process. Can be physical goods, intangible services, or product-service systems.
- Examples: consumer products, digital services, experiences, entertainment offerings.

[DEMAND]

- A stated or expected need that an artefact fulfills. Needs can originate from individuals, groups, or society as a whole.
- Examples: consumer demand for essentials, societal challenges, personal preferences.

[REVENUE]

- The benefit or reward an actor receives for contributing to value creation. Can be tangible or intangible.
- Examples: wages, profits, royalties, increases in social capital.

[FRAMEWORK]

- External factors influencing how value is created. Includes environmental, political, legal, and social factors.
- Examples: regulatory regimes, market conditions, intellectual property laws, educational infrastructure.

In some cases, more than one label applies. Do not mention labels that do not apply.

 # Task

Assign one or more of the labels from the codebook to the following text:

MINIMUM WAGE Codebook Prompt

You are a well-trained social scientist with a speciality for coding data for scientific research. You have been handed the following markdown codebook as instructions to code example sentences for their aspects.

Aspect Annotation in short argumentative text units.

The task is to find the aspect of an argumentative sentence or short text unit. An aspect in this task is defined as a sub-topic of discourse in the broader topic of the debate about minimum wage. It is possible, that more than one aspect is present in a sentence, please mark all applicable aspects.

Below, the aspect categories are further explained. Each category has a label name and details defining or illustrating them.

[MOTIVATION/CHANCES]

- Incentives and opportunities for workers to start a job/career, futher educate themselves and move up in a company.
- Influence of minimum wage on education in general or for particular groups.
- Eagerness and aspiration of workers.
- Keywords: motivation, incentives to education, upward mobility, worker morale.
- Examples:
  - "If we raise the minimum wage, low skilled workers might become complacent and do not want to move up"
  - "Paying employee better will make them more motivated to work for your company."

[COMPETITION/BUSINESS CHALLENGES]

- Effects of minimum wage on business competition and profitability.
- Challenges for small businesses.
- Keywords: labour costs, payability, SME competition.

[PRICES]

- Effects on consumer prices and inflation.
- Keywords: inflation, living costs, housing, daycare.
- Example: "Higher prices cancel positive effects of minimum wage."

[SOCIAL JUSTICE/INJUSTICE]

- Fairness, inequality, poverty, and living wage.
- Keywords: dignity, standard of living, poverty line.
- Examples:
  - "Workers should be able to afford paying their bills"
  - "Wages have to fairly compensate workers"

[WELFARE]

- Relationship between minimum wage and welfare spending.
- Keywords: welfare, subsidies, tax credits.
- Example: "Full-time workers should not have to rely on welfare"

[ECONOMIC IMPACT]

- Effects on the overall economy and consumer spending.
- Keywords: ripple effect, economic growth.

[TURNOVER]

- Effects on employee turnover and training costs.
- Keywords: job stability, hiring costs.

[CAPITAL VS LABOR]

- Power disparity between workers and owners.
- Role of unions and exploitation.
- Keywords: collective bargaining, working class, sweatshops.

[GOVERNMENT]

- State intervention and wage legislation.
- Keywords: regulation, free market.
- Example: "State vs. free market."

[UN/EMPLOYMENT RATE]

- Effects on employment, unemployment, automation, outsourcing.
- Keywords: job loss, labor supply.

[LOW-SKILLED]

- Effects on low-skilled and entry-level workers.
- Distinction from youth workers.
- Keywords: unskilled, entry-level.

[YOUTH AND SECONDARY WAGE EARNERS]

- Part-time workers, students, teenagers.
- Secondary income earners.

In some cases, more than one label applies. Do not mention labels that do not apply.

 # Task

Assign one or more of the labels from the codebook to the following text:

CCC Protest Dataset Prompt

Instruction: You're an expert political scientist categorizing news stories about protests into categories. Carefully read the definitions below, read the story, and write the single Label that best match the story.

[PROTEST]

- A crowd gathering to express disagreement with, or disapproval or anger or frustration toward, a specific individual or organization that is at or near the crowd's gathering point.
- The crowd must be near the target of the protest or in negative reaction to a recent or current event.

[RALLY]

- A crowd gathering to demonstrate and reinforce their mutual support for a set of political demands or claims.
- Typically includes speeches and may support press conferences.

[DEMONSTRATION]

- A crowd gathering to demonstrate their support to an external audience.
- Usually involves signs, banners, flags, or pickets.

[MARCH]

- A crowd moving on foot from one location to another.
- Does not include motorized or bicycle movement.

 # Task (single-label multi-class)

Do not write anything else except the Label in all caps.

Output Reminder: Write the name of the Label that fits best, with no other text. For example, "Label: PROTEST", "Label: RALLY", etc.

Text

B Details on Dataset Creation

This appendix describes the construction of the datasetsVALUE CREATION SCIENCE and VALUE CREATION NEWS. The datasets were developed as part of the *Wertschöpfungsradar* (Value Creation Radar, WSR) research project.

B.1 Original Dataset

B.1.1 Annotation Scheme

The annotation scheme is grounded in an abstract model of value creation systems, which distinguishes eight interdependent categories that characterise the essential elements of value creation processes. Each category is defined relationally—that is, through its connections to other elements in the model. The categories and their definitions are as follows:

- **Actor** (0): an individual or a collective that contributes a resource at its disposal to a process of creating an artefact.
- **Resource** (1): a capability of an actor that is used in the process of creating an artefact.
- **Process** (2): an activity for the creation of an artefact to which actors contribute their resources.
- **Pattern** (3): an organising principle through which a process integrates the resources to create an artefact.
- **Artefact** (4): a good that is created by integrating resources according to a pattern and that serves a demand.
- **Demand** (5): an articulated or anticipated need that an artefact serves.
- **Revenue** (6): a return an actor receives for contributing to the process of creating an artefact.
- **Framework** (7): a context factor that influences how value creation takes place.

Annotators were instructed to label noun phrases in English-language abstracts of scientific articles about value creation. Labels capture both positive and negative instances of each category (e.g., financial losses are labelled REVENUE; material costs are labelled RESOURCE).

B.1.2 Annotation Process

The annotation procedure was developed iteratively across two phases: a pilot study and a main annotation campaign.

Pilot study. An internal pilot was conducted in June–July 2023, in which 8 annotators labelled 20 abstracts in parallel using the DOCCANO annotation tool.[2] The pilot yielded a Fleiss' κ of 0.31, indicating fair for multi-label span annotation, but insufficient agreement. A qualitative analysis of 75 annotator comments and a co-occurrence analysis of category overlaps revealed three main sources of disagreement: (i) ambiguous category boundaries, particularly between PATTERN and PROCESS, REVENUE and DEMAND, and ACTOR and FRAMEWORK; (ii) missing formal coding rules regarding grammatical forms, compound phrases, and abbreviations; and (iii) insufficient attention to the value-creation context when annotating.

Guideline revision. Based on the pilot findings, the annotation guidelines were substantially revised. Key improvements included: sharpened category definitions supplemented with illustrative positive and negative examples drawn from the pilot data; the introduction of a step-by-step annotation procedure recommending that annotators first identify ACTOR, PROCESS, and ARTEFACT before turning to the remaining categories; explicit coding rules restricting annotations to complete noun phrases, prohibiting the splitting of compound expressions, and requiring annotators to consider the relationships between labelled entities as specified by the value creation model. A one-page reference sheet summarising the definitions, explanatory notes, and examples was provided as a quick-reference document.

Annotator training. Before the main campaign, annotators participated in a training session consisting of a presentation of the revised guidelines followed by a joint practice labelling exercise. An interactive training quiz was also distributed, which walked annotators through a sample text and tested their understanding of noun phrase boundaries, context-dependent category assignment, and the distinction between entities and concepts.

Main annotation campaign. The main campaign began in September 2023. Annotators labelled abstracts using DOCCANO with the eight predefined categories. The final dataset comprises 921 annotated abstracts. The collected annotated spans were proofed by a domain expert in order to improve label quality and remove any ambiguous data points.All involved social scientists consented to the use and release of this dataset.

B.2 VALUE CREATION SCIENCE

From these 921 annotated abstracts, we selected the sentences which contained annotated noun-phrases and transformed the data into a multi-label dataset. That is, the whole sentence gets the labels of the identified spans that are contained in it. After that, we selected only sentences with at least one label and which carried enough context for the labels they carried. This helps to minimize context loss in the dataset transformation.

B.3 VALUE CREATION NEWS

For the VALUE CREATION NEWS dataset, we collected 60 sentences from business-related news feeds and had the three most reliable annotators from the pilot study to label those.

C Qualitative Examples of Domain Differences

Below we provide example sentences from the VALUE CREATION NEWS dataset. Compared to scientific abstracts in the VALUE CREATION SCIENCE dataset, these texts typically use simpler language and more direct formulations.

1. **Example 1:** "GameStop has significantly reduced its investments in cryptocurrencies, NFTs, and Web3 applications, now holding no crypto tokens and shifting focus back to core business areas."

2. **Example 2:** "The telegraph quickly became popular for casual communication, and Ezra Cornell led its construction through southern New York."

3. **Example 3:** "Modern companies must adopt data-driven strategies, leveraging technologies like IoT, Big Data, and AI to remain competitive and optimize decision-making."

[2] https://github.com/doccano/doccano

D Statistical Significance Testing

In the following, we present the procedure for assessing (i) run-to-run stability and (ii) instance-level uncertainty when comparing each prompt prototype against the baseline.

D.1 Setup

For each dataset, we compare each prompt prototype against a fixed baseline prompt. We perform R=10 independent inference runs per (dataset, system) pair and store all test-set predictions. We compute accuracy, macro-F1, micro-F1, macro-precision, and micro-precision per run. Multi-label accuracy is subset accuracy (exact match). We set `zero_division=0` for F1 and precision.

D.2 Run-Level Stability

We assess robustness to inference stochasticity via paired comparisons within each run index. For each run $r \in \{1, \ldots, R\}$, we compute metric values $M_{\text{base}}^{(r)}$ and $M_{\text{proto}}^{(r)}$ on the full test set, then form deltas $\delta_r = M_{\text{proto}}^{(r)} - M_{\text{base}}^{(r)}$. We apply a two-sided one-sample t-test on $\{\delta_r\}_{r=1}^{R}$ to test H_0: $\mathbb{E}[\delta] = 0$, and report per-system mean$\pm$std, $\bar{\delta}$, the t-statistic, p-value, and t-based 95% confidence intervals with $R-1$ degrees of freedom. As a non-parametric sensitivity check, we also compute a two-sided Wilcoxon signed-rank test.

D.3 Instance-Level Uncertainty

We quantify finite-sample uncertainty via a paired bootstrap that preserves baseline–prototype pairing on the same sampled instances (B=20,000; seed=42; 95% confidence level).

The observed run-averaged improvement is $\Delta_{\text{obs}} = \frac{1}{R}\sum_r (M_{\text{proto}}^{(r)} - M_{\text{base}}^{(r)})$. For each replicate b, we resample N instance indices with replacement and apply this index vector to y, $\hat{y}_{\text{base}}^{(r)}$, and $\hat{y}_{\text{proto}}^{(r)}$ for every run r, yielding a run-averaged bootstrap delta $\Delta_b = \frac{1}{R}\sum_r (M_{\text{proto},b}^{(r)} - M_{\text{base},b}^{(r)})$. We report the 95% percentile CI $[\Delta_{2.5}, \Delta_{97.5}]$ and a two-sided bootstrap p-value:

$$p_{\text{boot}} = 2\min\big(\Pr(\Delta_b \le 0),\ \Pr(\Delta_b \ge 0)\big).$$

D.4 Out-of-Domain Comparison

For the dataset-shift analysis, the two test sets contain different instances and are not paired at the instance level. We compute per-run metrics on each test set and apply a paired-delta t-test across runs. We obtain a 95% CI and two-sided p-value via an unpaired bootstrap over runs (same B and seed as above). For worst-label analysis, we compute per-label F1 per run, identify the label with the lowest mean F1 per test set, and compare these distributions using Welch's t-test and an unpaired run-level bootstrap.

E Detailed Likert Scores

Figure 4 shows the Likert scores per label for each dimension.

Expert Evaluation Criteria Likert Scale

- **1** – Not relevant / clear / distinctive
- **2** – Somewhat relevant / clear / distinctive
- **3** – Quite relevant / clear / distinctive
- **4** – Highly relevant / clear / distinctive

Evaluation Dimensions

- **Relevance:** Does the code or label still capture the intended concept from the original codebook?
- **Clarity:** How easily can the code/label and its definition be understood by a researcher applying it?
- **Distinctiveness:** Is the code/label sufficiently distinct from other codes in the set, or could it overlap or be confused with another?

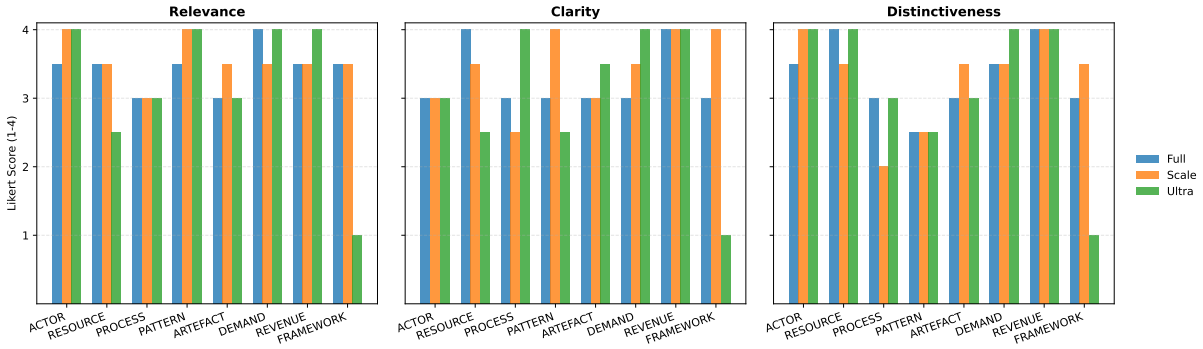

Figure 5: Overview of relevance, clarity, and distinctiveness across optimization methods.