

QualAlign: Benchmarking Automated Qualitative Coding Against Human Schemas

Taggert Smith*, Chuong Nguyen*, Qisen Yang, Oishani Bandopadhyay,
Yaru Su, Nadia Polikarpova, Xinyu Pi

University of California, San Diego

{tds002, chn021, qsyang, obandopadhyay, yas037, npolikarpova, xpi}@ucsd.edu

Abstract

Large language models are increasingly used for qualitative analysis, but existing inductive coding systems are difficult to compare because they use different datasets and system-specific evaluation metrics. We introduce **QualAlign**, a human-grounded benchmark for automated qualitative coding spanning 8 public datasets, each with research questions, inductively-derived codebooks, and code-excerpt assignments produced by expert researchers. The benchmark measures alignment with human expert coding via these artifacts using lexical, semantic, and assignment-based scores. Under a shared setup, we compare four recent systems for inductive codebook generation: LLoM, ThematicLM, HICode, and LOGOS. We find that while human and LLM-produced codebooks show moderate semantic resemblance given the same datasets and research questions, lexical agreement scores remains low, indicating different vocabulary used to describe similar concepts. Moreover, code-excerpt assignments diverge dramatically: given semantically similar conceptual labels, LLMs still partition the data differently than human researchers. These patterns suggest that current systems may be useful for proposing candidate codes or surfacing broad conceptual patterns for analyst review and revision—similar to prior topic modeling methods; however, they cannot mimic human researchers’ excerpt-level coding decisions. Based on these findings, we caution that LLM-generated codebooks that appear semantically similar to human work cannot necessarily be taken to represent a similar qualitative interpretation of the data, and thus should not replace the interpretive work of human coders.

Code: <https://github.com/chuongnguyen26/HumanAlignment-Dataset>

1 Introduction

Qualitative analysis often involves studying open-ended text by organizing excerpts from text corpora into labels called *codes* and broader groupings. A common form of this process is *inductive coding*, where analysts do not start with a fixed codebook but instead develop codes from the data as they read and compare excerpts (Braun & Clarke, 2006; 2014; 2023). Because this process is labor-intensive, recent large language model (LLM) based systems have begun to automate it, from human-in-the-loop tools to fully automated pipelines (Gilardi et al., 2023; Dai et al., 2023; Gao et al., 2024; Schroeder et al., 2025a). Recent systems also go further by inducing qualitative codebooks directly from raw corpora (Lam et al., 2024; Qiao et al., 2025; Zhong et al., 2025; Pi et al., 2025). Evaluation schemes vary across these prior works, and the question remains: when an automated system performs the same inductive coding task as human experts, how should its outputs be compared to human-produced qualitative artifacts?

Given a corpus and research question, inductive qualitative coding typically produces two analytic artifacts: it assigns codes to text excerpts creating a record of *code-excerpt assignments*, then collects codes into a *codebook* with brief definitions or inclusion criteria, wherein codes

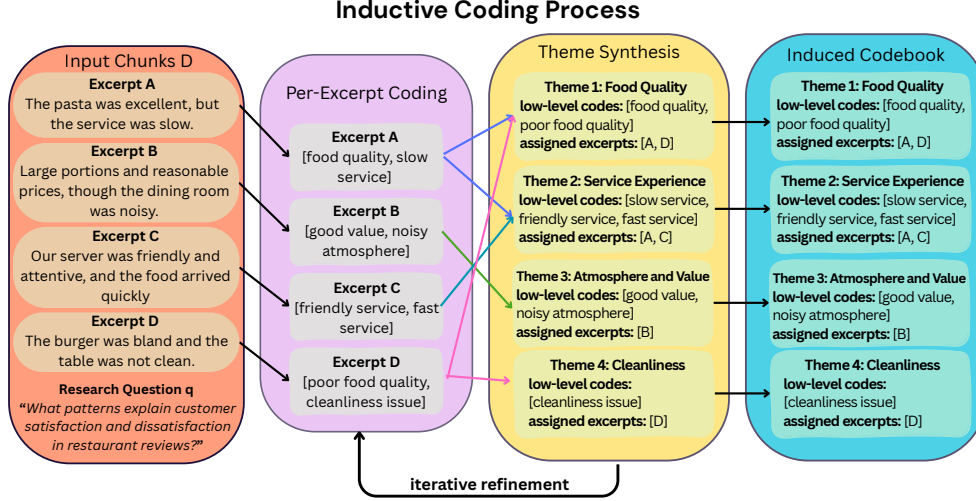


Figure 1: General inductive coding pipeline. Given input excerpts and a research question, an automated inductive coding framework carries out the middle stages of the process: it proposes excerpt-level codes, synthesizes them across the corpus, and constructs an induced codebook or schema. This figure illustrates the general task performed by automated inductive coding systems rather than the QualAlign evaluation pipeline itself.

may also be organized into higher-level patterns, often called themes. An excerpt may receive multiple codes, and the same code may appear across many excerpts. A natural way to compare human and automated analyses, then, is to compare the human reference and model-generated artifacts for the same corpus and questions, each consisting of a codebook plus code-excerpt assignments.

Unlike deductive coding, where the codebook is fixed in advance, inductive systems must decide what codes should exist, how broad or narrow they should be, and how they should be assigned across excerpts (Fereday & Muir-Cochrane, 2006). A system can therefore produce outputs that look plausible while still differing substantially from human analysis in codebook structure, coding granularity, and excerpt-level use. Existing automated methods are typically evaluated with system-specific criteria and often without an independent human reference, making cross-method comparison difficult. Specifically, recent systems such as HICode and LOGOS are introduced together with their own evaluation datasets and metrics, so their reported gains are tied to different data and scoring setups rather than to one shared human-grounded benchmark. As a result, the field still lacks a shared human-grounded benchmark for judging how well automated inductive coding systems recreate expert qualitative codebooks (Braun & Clarke, 2019; 2021; Nelson, 2020).

We introduce **QualAlign**, a human-grounded benchmark for automated inductive coding. It spans 8 public datasets, with excerpt-level expert annotations released in a shared format. We use it to compare four recent systems, LLoM (Lam et al., 2024), ThematicLM (Qiao et al., 2025), HICode (Zhong et al., 2025), and LOGOS (Pi et al., 2025), against the same expert reference artifacts under one common setup. We test whether current frameworks align with human qualitative codebooks and code-excerpt assignments when evaluated on shared data and metrics rather than through separate system-specific pipelines.

Across the benchmark, agreement with human coding remains low. Even under the more favorable matching view, the best system reaches only Krippendorff’s $\alpha = 0.192$ and Jaccard similarity = 0.302, while semantic similarity is substantially higher (0.58–0.60 cosine scores and at most 0.724 under embedding-based matching). Our case studies illustrate how systems can recover topical or semantic proximity while still missing the human ground truth through granularity mismatch, valid paraphrase with low lexical overlap, or genuine interpretive drift. For example, in one case study a model-generated code reaches

cosine similarity = 0.837 with the human reference yet receives an LLM-judge score of only 0.0 because it reframes the excerpt as a neutral description rather than preserving the human coder’s framing. These results suggest that current systems may helpfully propose candidate codes or surface broad semantic patterns for analyst review, but they are not yet reliable enough for final excerpt-level coding, nor can they obviate the work of human interpretation.

2 Related Work

LLMs for Qualitative Analysis. LLM-based qualitative analysis spans two settings. In *deductive* coding, the codebook is fixed in advance; in *inductive* coding, the system must derive the codebook from the data itself. Existing works on deductive coding have shown the power of LLM on text annotation (Hou et al., 2024; Hila & Hauser, 2025a). Exploration on inductive coding differs in how much control remains with the researcher: some systems are explicitly human-in-the-loop, such as PaTAT and CollabCoder (Gebreegziabher et al., 2023; Gao et al., 2024), while others move toward fully automated pipelines (De Paoli, 2024b; Khan et al., 2024). Our benchmark focuses on the most demanding end of this design space: systems such as LLoM, ThematicLM, HICode, and LOGOS that induce qualitative codebooks directly from raw corpora (Lam et al., 2024; Qiao et al., 2025; Zhong et al., 2025; Pi et al., 2025). Across the literature, a recurring pattern is that LLMs often recover broad themes or semantically related codes while struggling to reproduce finer-grained human distinctions (Dai et al., 2023; De Paoli, 2024a; Deiner et al., 2024; Parfenova et al., 2025).

Need for a Shared Human-Grounded Benchmark. Existing systems are typically tested on different datasets and judged with system-specific criteria, which makes it difficult to compare their alignment with expert qualitative work. Prior NLP literature offers lexical (Papineni et al., 2002; Lin, 2004), and semantic (Zhang et al., 2019; Reimers & Gurevych, 2019) metrics that can be applied to compare LLM and human codebooks, as well as set- or partition-based (Jaccard, 1912; Hubert & Arabie, 1985; Meilă, 2003; Vinh et al., 2010; Fowlkes & Mallows, 1983), and chance-corrected agreement-based metrics (Cohen, 1960; Krippendorff, 2018; Gwet, 2008; Scott, 1955) that can be applied once human code-excerpt assignments are available. Related work suggests that these views do not always agree, highlighting the need for a diverse suite of evaluation metrics (Schroeder et al., 2025b; Li et al., 2025; Goodall et al., 2026). QualAlign leverages a diverse and representative set of these metrics to address the lack of a shared human-grounded benchmark for comparing automated inductive coding systems against the same reference data under a common setup.

3 Methodology

3.1 Terminology

We use the following qualitative-analysis terminology throughout. A *code* is a short label that captures a main idea of a particular text excerpt or passage from a larger corpus; for example, a passage about managing a task list might receive codes such as *importance of prioritization* or *ability to create subtasks*. A *codebook* is the set of codes used in one analysis, often including brief definitions, inclusion criteria, or examples. In thematic analysis, granular codes are often organized under broader codes, i.e. “themes.” For the sake of consistent comparison across heterogeneous studies in our dataset, we use the minimal definition of a codebook—a list of codes organized hierarchically in one to three levels of granularity. We use both codebook and *schema* interchangeably in this work. *Excerpt-level assignments* record which codes are attached to which text excerpts, and a single excerpt may receive multiple codes. *Excerpts* in this paper vary in exact form depending on dataset: in an analysis of social media posts, one entire post’s text may be considered an excerpt, while in an interview transcript analysis, it may be the specific phrase or passage highlighted by the original analysts. For each dataset, we select the smallest meaningful unit to which codes are assigned to define excerpt boundaries. *Research question(s)* refer to the set of one or more research questions stated by the original authors of the source publication

for each dataset. These question sets are accordingly given to each system to guide their coding. It should be noted that two publications subsetting their code-excerpt assignments such that codes were further assigned to specific individual research questions, effectively presenting multiple separable codebooks for the same dataset. In the main results, we ignore the assignment of codes to specific research questions and evaluate one artifact pair per dataset to avoid these two cases biasing the overall results. In Appendix A.4.1, we generate separate, question-conditioned codebooks and assignments for each individual research question as a robustness check.

3.2 Task Formulation

For each dataset D and system, QualAlign compares two artifact pairs: the human reference $(C_D^{\text{human}}, A_D^{\text{human}})$ and the model output $(C_D^{\text{model}}, A_D^{\text{model}})$ where C_D is the a codebook and A_D is the corresponding code-excerpt assignment structure. For an excerpt x_i , then $A_D(x_i)$ denotes the set of codes attached it. The set notation reflects the fact that qualitative coding is often multi-label.

3.3 Code Matching

Before comparing a human codebook with a model codebook, we must decide which human and model codes should be treated as matches for evaluation. Here, a match means that the two codes may be similar enough in meaning to be compared as potentially valid alternative expressions of the same underlying analytic idea, even if they differ in wording.

Because inductive codebooks are generated independently, human and model code labels rarely match exactly in wording, even when they may describe the same concept (e.g., "*calm feeling*" vs. "*peaceful mood*"). We therefore align codes using two complementary similarity measures rather than exact label matching:

1. **Cosine similarity** between code-label embeddings (qwen3-embed-0.6b), capturing similarity in *semantic meaning* between the labels themselves.
2. **Jaccard similarity** over each code’s assigned-excerpt set, capturing similarity in *how the code was applied*. Put another way, it measures similarity in the *extensional definition* of each code, in place of explicit definitions and inclusion criteria.

Under each measure, every model code is mapped independently to its closest human code (*best-match*). Since automated systems may over-generate codes relative to humans, multiple codes from one codebook may map to a single code from the other. A worked example appears in Appendix A.4.3, alongside a less charitable, one-to-one matching algorithm as a robustness check.

We present results from both *Cosine* and *Jaccard*-based matching in this paper, aligning with our positioning of this benchmark as a diagnostic tool, but we also offer a combined cost function in Appendix A.4.4 for future users who may wish to find a single optimal mapping that balances these two axes of similarity.

3.4 Evaluation Metrics

We report four complementary views of alignment, each represented by a representative set of computational metrics. Appendix A.3.1 contains a larger set of secondary metrics that robustify and reinforce the same family-level patterns.

3.4.1 Comparing Codebooks

Lexical Similarity. These metrics ask whether matched codes share similar surface forms. BLEU and ROUGE-L measure overlap in wording (Papineni et al., 2002; Lin, 2004). These metrics are useful for quantifying differences in vocabulary and phrasing between human and LLM-generated labels.

Semantic Similarity. These metrics ask whether matched codes are close in general meaning, even if their exact wording differs. We include two embedding-based views. BERTScore F1 measures token-level contextual semantic similarity through contextual token matching (Zhang et al., 2019). Cosine similarity over qwen3-embed-0.6b code embeddings measures semantic proximity between whole code labels (Reimers & Gurevych, 2019).

Model-Based Interpretive Judgment. This metric asks whether a large language model judges two aligned codes as expressing a similar qualitative interpretation, capturing nuances of paraphrase, abstraction, and framing. In the main text, we report judge scores from qwen3.5-35b under a numeric rubric prompt. We treat this metric as a conservative signal rather than a validated measure of qualitative similarity, since it can capture framing differences missed by embeddings but remains model-dependent. Appendix A.4.2 reports judge-substitution robustness.

3.4.2 Comparing Code-Excerpt Assignments

Assignment Metrics. These metrics measure how often corresponding human and model codes are attached to the same excerpts. Krippendorff’s α is a common measure of inter-rater reliability in codebook-style qualitative analysis because it subtracts out the level of agreement expected from random assignment, reflecting agreement corrected for chance Krippendorff (2018). Further, rather than scoring two excerpts’ code sets as simply matching or not, MASI distance grades their relationship—identical, subset, partial overlap, or disjoint—and scales the score accordingly Passonneau (2006). Unless otherwise noted, “Krippendorff’s α ” in the main text refers to this MASI-based variant designed for multi-label settings; A.3.1 reports traditional/binary variants as supplementary views. The Jaccard index is a measure of set overlap, formulated as the intersection over union of assigned excerpts for each code in a matched pair Jaccard (1912). Its monotonic transformation, F1, is often used in classification tasks. These two specifically are chosen to be readable to audiences from both qualitative analysis and machine learning backgrounds.

MASI grades the relationship between two code sets—identical, subset, overlapping, or disjoint—rather than scoring them as simply match/no-match, so partial overlap between excerpt sets is credited rather than penalized as full disagreement. This distance is then chance-corrected in the usual way to produce α . Unless noted otherwise, “Krippendorff’s α ” refers to this MASI-based variant throughout the main text; traditional/binary variants appear in A.3.1. As an uncorrected, direct view, we also report the Jaccard index: intersection over union of the two assigned-excerpt sets for each matched code pair Jaccard (1912). Unlike α , it applies no adjustment for chance agreement, making it more sensitive to the raw size and composition of each code’s excerpt set. Its monotonic transform, F1, offers a familiar counterpart for classification-oriented readers.

We report these metric families separately because they capture different aspects of qualitative alignment. Lexical overlap, semantic proximity, assignment fidelity, and interpretive correspondence can diverge substantially in inductive coding, and the benchmark is designed to make those differences visible rather than collapse them into a single score.

3.5 Experimental Setup

3.5.1 Datasets

We evaluate four automated inductive coding systems, LLoM (Lam et al., 2024), ThematicLM (Qiao et al., 2025), HICode (Zhong et al., 2025), and LOGOS (Pi et al., 2025), on eight human-annotated datasets spanning health, software engineering, information science, and everyday technology use: ADHD Reddit¹ Smith & Smithwick (2026), Data Sharing (Donaldson & Koepke, 2022), Mental-Health (Spek & Vegt, 2025), Patient Deterioration (Beane et al.), Scrum Quality (Alami & Krancher, 2022), Search Systems (Hickner, 2022), Shuffle Music (Sanfilippo et al., 2020), Stroke Wearables (Lee et al., 2022); details in A.2).

¹Currently available via our public benchmark repository. When the official proceedings and camera-ready DOI for the source study have been released, the citation will be updated and it will be linked there as well.

3.5.2 Shared Re-coder and Isolating Codebook vs Coder Effects

The four systems do not expose code-excerpt assignments in a directly comparable format, so QualAlign adds a shared deductive re-coding stage: an LLM re-coder assigns each system’s induced codes back onto text chunks (without ever seeing the human codebook) to produce standardized assignments for evaluation. Applying the same re-coder to the human codebook itself yields three assignment mappings per dataset:

1. the original expert (human) assignments,
2. re-coded assignments using the **human** codebook,
3. re-coded assignments using each **system’s** codebook.

The comparison of these artifacts yields multiple relevant assignment scores.

Comparing (2) vs. (3) isolates the effect of codebook differences alone (holding the re-coder fixed) and is what we report as our main assignment-based results in Table 1, hereafter referred to as the “codebook effect.” This allows the benchmarking of automated codebook-only inductive systems, including those we foreground in this paper (LLoM, HiCode, LOGOS, ThematicLM), at the cost of missing the differences between human and LLM assignment decisions.

Comparing (1) vs. (3) reflects combined codebook-related and assignment-related divergence from the human reference, and is thus useful for evaluating the current state of fully automated, “end-to-end” qualitative coding, and any future systems which—like human teams—produce both an induced codebook and code-excerpt assignments.

Comparing (1) vs. (2) isolates the re-coder’s own performance independent of any inductive system. We use the resultant “ceiling” to contextualize the benchmark’s “end-to-end” scores and thus better compare the effects of the specific codebook-only systems evaluated here. Further, this comparison allows QualAlign to serve as a benchmark for automated deductive coding methods.

This re-coding stage is a necessary benchmark component, not part of the original systems, and is designed to be swapped out as better automated deductive coders become available, or as inductive systems begin to natively expose code-excerpt assignments.

Automated deductive coding is itself an open problem (Hou et al., 2024; Hila & Hauser, 2025b; Schroeder et al., 2025b), and our work is among the first to assess it across a diverse set of naturalistic qualitative corpora. We use—to our knowledge—the state of the art in automated deductive coding methods to the extent which it can be applied to our setting. We also measure the noise introduced by the recoder directly as a “self-consistency score,” presented alongside the main results (Figure 2) as a theoretical ceiling value for the “codebook-effect” scores. We find that this step contributes some but not overwhelming noise, as consistency between re-coder runs is similar to that expected between human analysts Krippendorff (2018).

3.5.3 Evaluation Protocol

For each system-dataset pair:

1. The system generates a codebook from the raw text and research question(s).
2. The shared re-coder assigns those codes back to excerpts.
3. Human and model codes are aligned via the matching procedure in §3.3.
4. Codebook-level metrics (BLEU, ROUGE-L, BERTScore F1, cosine, LLM judge) and assignment-level metrics (Krippendorff’s α , Jaccard) are computed over aligned pairs.
5. Scores are averaged across research questions within each dataset, then aggregated by system, to avoid datasets with more separable questions dominating the results.

Models. We use gpt-oss-120b for codebook generation and re-coding, and qwen3-embed-0.6b for embedding-based code alignment and cosine similarity.

Method	Kripp. α	Jac. Sim.	Cosine	BERT F1	LLM Judge	BLEU	ROUGE-L
HICode	0.061	0.101	0.724	0.211	0.545	0.050	0.261
LLOOM	0.047	0.081	0.721	0.169	0.458	0.034	0.210
LOGOS	-0.041	0.073	0.722	0.126	0.523	0.017	0.146
ThematicLM	0.062	0.098	0.703	0.129	0.420	0.022	0.163

Method	Kripp. α	Jac. Sim.	Cosine	BERT F1	LLM Judge	BLEU	ROUGE-L
HICode	0.162	0.253	0.603	0.123	0.451	0.020	0.107
LLOOM	0.167	0.302	0.587	0.077	0.400	0.012	0.076
LOGOS	0.109	0.234	0.592	0.056	0.454	0.008	0.075
ThematicLM	0.192	0.278	0.592	0.069	0.386	0.012	0.085

Table 1: Top: embedding-based matching. Bottom: Jaccard-based matching. Values are mean agreement scores; bold indicates best-performing method per metric.

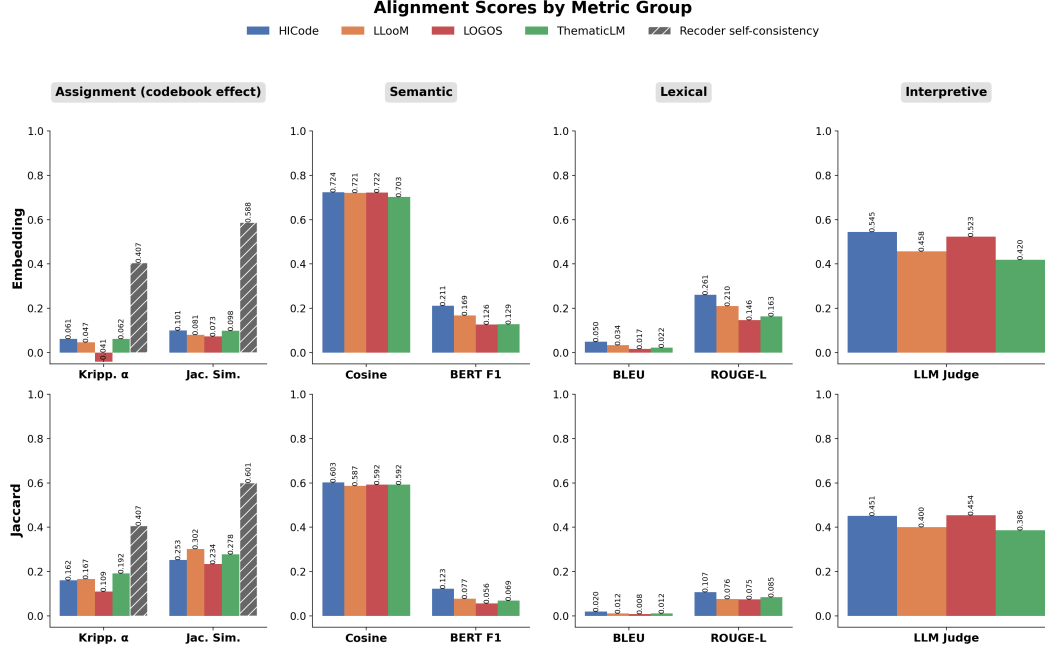


Figure 2: Alignment scores by metric group across methods. Top: embedding-based matching. Bottom: Jaccard-based matching. Values are mean scores across all datasets; higher is better. Assignment scores are presented with a recoder self-consistency baseline from comparing two runs of the same shared LLM coder given the same codebook and prompts.

4 Results

4.1 No System Mimics Human Coding Decisions

Figure 2 reports average performance across all datasets. Alignment with human coding is low overall: all systems achieve weak excerpt-level assignment performance under both a chance-corrected reliability view and a direct set-overlap view, with Krippendorff’s α peaking at 0.192 and Jaccard peaking at 0.302. 53% of excerpts have zero code overlap between the human ground truth assignments and end-to-end LLM assignments. BLEU and ROUGE-L remain low across methods, indicating limited lexical overlap even when two codes are topically related, suggesting that exact vocabulary-based alignment between human and LLM codebooks remains a distant target. The semantic metrics are higher, indicating broad topical alignment, while LLM-as-a-judge is more conservative, suggesting that embedding similarity alone does not guarantee qualitatively similar interpretation. These metric groups together reinforce that despite broad semantic resemblance, current systems fail to replicate expert-level coding, partitioning, organizing, interpreting, and articulating the data differently. This contrast illustrates that while automated systems

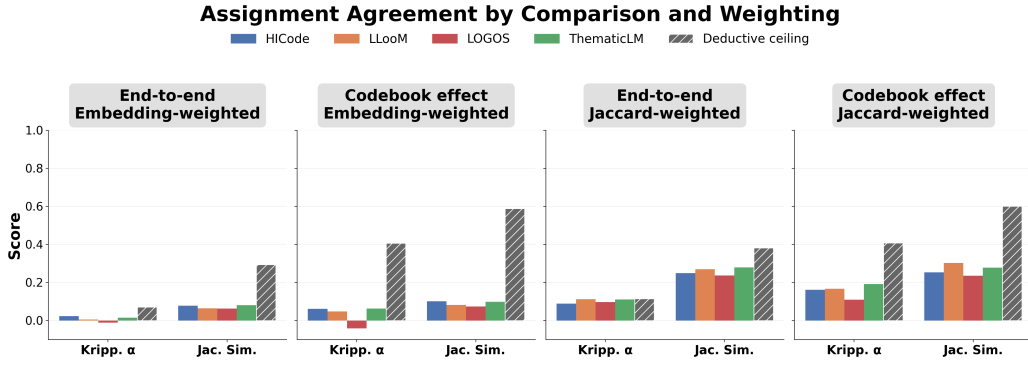


Figure 3: End-to-end assignment performance compared to codebook-effect assignment performance. End-to-end assignment is presented alongside the practical ceiling value achieved by an oracle LLM coder for context. See Appendix A.3.4 for an alternate view of end-to-end assignment scores as a ratio baselined against the deductive ceiling.

may produce semantically similar labels, humans and LLMs assign these seemingly similar labels to markedly disparate data.

88% of LLM-induced codes get paired to a different human code depending on whether matching is done using embedding or Jaccard-based cost, indicating misalignment between the semantic meaning of labels and the implicit, extensional definition implied by their usage. This dramatic tradeoff between matching paradigms further evidences that while automated systems may produce somewhat similar labels (per the embedding-weighted matching) and somewhat similar data clusters in aggregate (per the Jaccard-weighted matching), the assignment of data to labels diverges significantly from human teams. LLMs and humans do not assign similar labels to similar data, and do not use similar lexical vocabulary for similar concepts. Implicitly, the qualitative interpretation of datapoints—represented by the assigned labels—does not align between automated systems and expert coders.

This pattern is robust to granularity diagnostics (Appendix A.2.3), deductive-ceiling analysis (Appendix A.3.4), LLM judge substitution (Appendix A.4.2), and matching-sensitivity analysis (Appendix A.4.3); see the appendix for details.

No system is consistently strong across lexical, semantic, interpretive, and excerpt-level assignment views at the same time, and per-metric winners are not necessarily robust to factors such as changing the LLM-as-judge model (see Appendix A.4.2). We therefore treat these rankings as secondary to the broader result that current systems still fall well short of strong human alignment.

4.2 Inductive Versus Deductive Assignment Error

Upon rerunning the same benchmark pipeline with the human reference codebook supplied during the recoding stage, excerpt-level agreement remains limited across datasets: Krippendorff’s α ranges from 0.01 to 0.12 (mean 0.07) under embedding-based matching and from 0.05 to 0.26 (mean 0.11) under Jaccard-based matching, while Jaccard similarity ranges from 0.14 to 0.44 (mean 0.29) under embedding-based matching and from 0.20 to 0.58 (mean 0.38) under Jaccard-based matching.

In other words, current end-to-end automated performance is limited due to both difference between LLM-induced and human codebooks as well as LLM difficulties in excerpt-level deductive coding, even when the human codebook is known. We present this as a significant finding in its own right—the current automated deductive coding methods struggle when applied to real qualitative datasets at scale.

However, the poor assignment-based scores are not entirely attributable to the failure of the deductive recoder, as the pattern worsens under a shared coder where the only variable is the induced versus ground truth codebook, per Figure 3.

The isolated induced-codebook-related assignment error may be partially explained by the uniformly low lexical overlap scores, given the sensitivity of LLM responses to specific

language choices Hou et al. (2024). However, the low deductive ceiling clearly implicates broader interpretive divergence between automated and human coders.

5 Case Studies

These examples show why broad semantic similarity between human and LLM codebooks does not guarantee reliable excerpt-level coding. In each case, the model output looks plausible, but still diverges from the human reference in a way that matters for qualitative analysis.

Case 1. Near-identical labels can encode different coding behavior

Raw excerpt: "We also include [a] clear declaration of the CC0 license. So a machine could also identify that there are no restrictions in this data."

Human code: Data Licenses (26 excerpts)

Induced code: Data Licensing (3 excerpts)

Matching method: Best-match (embedding / cosine)

Metrics: Lexical = 1.000, Cosine = 0.934, LLM = 1.0, Jaccard = 0.074 (2 shared excerpts)

Case 2. Granularity mismatch reduces assignment scores

Raw excerpt: "Reminds me of being young. I used to enjoy this song when it was in the charts, when I was much younger. I created a playlist full of songs that reminded me of my youth... I listen to [it] for nostalgia."

Human code: Reminisce–Nostalgia

Induced codes on this excerpt: Nostalgia; Nostalgic memory recall; Personal memory association; Nostalgic personal connection; Nostalgia-driven listening motivation; Nostalgia-driven listening; Nostalgic personal history

Featured matched pair: Reminisce–Nostalgia ↔ Nostalgic memory recall

Matching method: Best-match (embedding / cosine)

Metrics: Lexical = 0.000, Cosine = 0.866, LLM = 1.0, Jaccard = 0.091 (12 shared excerpts)

Case 3. Semantic proximity masking interpretive drift

Method and dataset: ThematicLM, *adhd_reddit*

Raw excerpt: "overall nice idea but i think creating a task in such a system might take annoyingly long you need to set up a side quest a main quest and the focus context for every single task no one with adhd would keep doing that ... it forces you to really prioritize"

Human ground-truth code: tedious task management apps

Predicted code: Single urgent priority task management

Matching method: Best-match (embedding / cosine)

Metrics: BERT F1 = 0.439, ROUGE-L = 0.444, Cosine = 0.837, Jaccard = 0.250, LLM = 0.0

Case 1 - Near-identical labels can encode different coding behavior: A label-only evaluation would suggest that the system faithfully recovers the human category: the labels share identical lexical content, high embedding similarity, and a maximal LLM-judge score. Their extensional behavior tells a different story—the human code covers 26 excerpts and the induced code covers only three, with just two excerpts in common. A model can therefore reproduce the name and apparent meaning of a human code without reproducing the boundary that determines when the code is applied, which is why label-level and assignment-level agreement must be reported separately.

Case 2 - Granularity mismatch reduces assignment scores: Each induced code above is a plausible expression of nostalgia on its own, but the system fragments one human category into a family of near-synonyms. Excerpts that the human coder collects under a single code are consequently divided across several induced codes, a granularity mismatch that directly lowers the assignment overlap of every individual matched pair even though the broad concept is clearly recovered. This case gives an excerpt-level mechanism for the systematic codebook-cardinality differences reported in Appendix A.2.3. **Case 3 - Semantic proximity masking interpretive drift:** Both labels occupy the same task-management semantic neighborhood, which keeps cosine similarity high. The human code, however, foregrounds the burden of a tedious application workflow, while the model reframes the excerpt as a neutral description of managing one urgent priority. The predicted code preserves topical vocabulary but loses the excerpt's negative evaluation, and the LLM-judge score of 0.0 captures that interpretive shift far more clearly than cosine similarity alone.

These cases clarify the aggregate patterns in Table 1 and show why human-LM schema alignment must be evaluated from multiple perspectives at once. Agreement-based metrics are sensitive to codebook granularity and fragmentation, while embedding-based similarity can remain speciously high even when the LLM interpretation does not align with the human coding frame. Across all three cases, systems recover the high-level semantic structure of human codebooks without reliably reproducing the human-like category boundaries, assignment behavior, or interpretive framing that qualitative coding depends on.

6 Discussion

Across the benchmark, excerpt-level assignment alignment remains low even when broader semantic similarity is substantially higher. This gap suggests that current automated inductive coding systems can often recover topical relatedness or plausible codebook semantics without reproducing the excerpt-level distinctions that matter in human qualitative analysis. The case studies and granularity diagnostics show that this is not a single failure mode: semantically related labels may still differ in abstraction level, plausible paraphrases may look weak under lexical metrics, and topically similar codes may still shift the underlying interpretation.

The deductive baseline reinforces the same point. Even when the human reference codebook is supplied, agreement remains limited, which suggests that disagreement is not solely a consequence of open-ended codebook induction. More broadly, inductive coding allows several analytic degrees of freedom, including deciding what codes should exist, what level of abstraction they should occupy, how related ideas should be split or merged, and how codes should be applied across excerpts. Granularity mismatch and interpretive drift are two prominent examples, but they do not exhaust the failure modes observed here. As the results and case studies show, current systems exhibit multiple disagreements between semantic resemblance and coding behavior, fundamentally challenging the interpretability and explorability of model outputs by human analysts.

The granularity mismatch pattern merits special note as an example of QualAlign metrics highlighting opportunities for design improvement. A codebook that is substantially finer-grained than a human reference, as most systems tend to produce, is not simply a harder target to match under our metrics; it is not usable as a draft for an analyst to revise, since working through many near-synonymous codes (Case 2’s seven nostalgia-adjacent labels for one human category) reintroduces the consolidation work that automated coding is meant to reduce, even though each individual code is a locally defensible paraphrase. Nor is such a codebook useful as a digestible finished product for a downstream reader. Compression in the other direction, which ThematicLM tends toward (per ??), is also problematic: collapsing several human categories into one broader label discards an interpretive distinction a human team might have chosen to keep, without surfacing that choice for review. We suggest that future systems should make granularity adjustable by a downstream analyst.

7 Conclusion

QualAlign provides a shared human-grounded benchmark for comparing automated inductive coding systems against expert qualitative artifacts across eight datasets. Under this benchmark, current systems can support analysts by generating draft codebooks and surfacing broad conceptual patterns, but they are not yet reliable enough for final excerpt-level coding or for replacing expert analysts.

A promising next step is not to eliminate the flexibility of inductive coding, but to better structure it. Future systems should preserve room for qualitative exploration while introducing more stable boundaries for excerpt-level alignment, comparison, and reuse, and should make code granularity, assignment behavior, and interpretive framing more visible and controllable for researchers.

References

- Adam Alami and Oliver Krancher. How scrum adds value to achieving software quality? *Empirical Software Engineering*, 27(7):165, 2022.
- A Beane, W Wijesiriwardana, C Pell, et al. De-identified interview transcripts and analysis. figshare. dataset. 2022.
- Virginia Braun and Victoria Clarke. Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2):77–101, 2006. doi: 10.1191/1478088706qp063oa. URL <https://doi.org/10.1191/1478088706qp063oa>.
- Virginia Braun and Victoria Clarke. What can “thematic analysis” offer health and wellbeing researchers?, 2014.
- Virginia Braun and Victoria Clarke. Reflecting on reflexive thematic analysis. *Qualitative Research in Sport, Exercise and Health*, 11(4):589–597, 2019. doi: 10.1080/2159676X.2019.1628806. URL <https://doi.org/10.1080/2159676X.2019.1628806>.
- Virginia Braun and Victoria Clarke. One size fits all? what counts as quality practice in (reflexive) thematic analysis? *Qualitative research in psychology*, 18(3):328–352, 2021.
- Virginia Braun and Victoria Clarke. Toward good practice in thematic analysis: Avoiding common problems and be(com)ing a knowing researcher. *International Journal of Transgender Health*, 24(1):1–6, 2023. doi: 10.1080/26895269.2022.2129597. URL <https://doi.org/10.1080/26895269.2022.2129597>.
- Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960.
- Shih-Chieh Dai, Aiping Xiong, and Lun-Wei Ku. Llm-in-the-loop: Leveraging large language model for thematic analysis. In *Findings of the association for computational linguistics: EMNLP 2023*, pp. 9993–10001, 2023.
- Stefano De Paoli. Further explorations on the use of large language models for thematic analysis: Open-ended prompts, better terminologies and thematic maps. In *Forum Qualitative Sozialforschung*, volume 25, pp. 5. Institut fur Klinische Sychologie and Gemeindesychologie, 2024a.
- Stefano De Paoli. Performing an inductive thematic analysis of semi-structured interviews with a large language model: An exploration and provocation on the limits of the approach. *Social Science Computer Review*, 42(4):997–1019, 2024b. doi: 10.1177/08944393231220483. URL <https://doi.org/10.1177/08944393231220483>.
- Michael S Deiner, Vlad Honcharov, Jiawei Li, Tim K Mackey, Travis C Porco, and Urmimala Sarkar. Large language models can enable inductive thematic analysis of a social media corpus in a single prompt: human validation study. *JMIR infodemiology*, 4(1):e59641, 2024.
- Devan Ray Donaldson and Joshua Wolfgang Koepke. A focus groups study on data sharing and research data management. *Scientific Data*, 9(1):345, 2022.
- Jennifer Fereday and Eimear Muir-Cochrane. Demonstrating rigor using thematic analysis: A hybrid approach of inductive and deductive coding and theme development. *International journal of qualitative methods*, 5(1):80–92, 2006.
- Edward B Fowlkes and Colin L Mallows. A method for comparing two hierarchical clusterings. *Journal of the American statistical association*, 78(383):553–569, 1983.
- Jie Gao, Yuchen Guo, Gionnieve Lim, Tianqin Zhang, Zheng Zhang, Toby Jia-Jun Li, and Simon Tangi Perrault. CollabCoder: A lower-barrier, rigorous workflow for inductive collaborative qualitative analysis with large language models. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, 2024. doi: 10.1145/3613904.3642002. URL <https://doi.org/10.1145/3613904.3642002>.

- Simret Araya Gebreegziabher, Zheng Zhang, Xiaohang Tang, Yihao Meng, Elena L. Glassman, and Toby Jia-Jun Li. Patat: Human-ai collaborative qualitative coding with explainable interactive rule synthesis. *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023. URL <https://api.semanticscholar.org/CorpusID:258216675>.
- Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. ChatGPT outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30):e2305016120, 2023. doi: 10.1073/pnas.2305016120. URL <https://doi.org/10.1073/pnas.2305016120>.
- Leonardo S Goodall, Dor Shilton, Daniel A Mullins, and Harvey Whitehouse. Large language models struggle with ethnographic text annotation. *arXiv preprint arXiv:2601.12099*, 2026.
- Kilem Li Gwet. Computing inter-rater reliability and its variance in the presence of high agreement. *British Journal of Mathematical and Statistical Psychology*, 61(1):29–48, 2008.
- Andy Hickner. Data for “How do Search Systems Impact Systematic Searching? A Qualitative Study”, 2022. URL <https://doi.org/10.5064/F68AAMX3>.
- Angjelin Hila and Elliott Hauser. Assessing the reliability of large language models for deductive qualitative coding: A comparative intervention study with chatgpt. *Proceedings of the Association for Information Science and Technology*, 62(1):275–285, 2025a.
- Angjelin Hila and Elliott Hauser. Assessing the reliability of large language models for deductive qualitative coding: A comparative study of chatgpt interventions, 2025b. URL <https://arxiv.org/abs/2507.14384>.
- Chenyu Hou, Gaoxia Zhu, Juan Zheng, Lishan Zhang, Xiaoshan Huang, Tianlong Zhong, Shan Li, Hanxiang Du, and Chin Lee Ker. Prompt-based and fine-tuned GPT models for context-dependent and -independent deductive coding in social annotation. In *Proceedings of the 14th Learning Analytics and Knowledge Conference, LAK '24*, pp. 518–528, New York, NY, USA, 2024. Association for Computing Machinery. doi: 10.1145/3636555.3636910. URL <https://doi.org/10.1145/3636555.3636910>.
- Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.
- Paul Jaccard. The distribution of the flora in the alpine zone. 1. *New phytologist*, 11(2):37–50, 1912.
- Awais Hameed Khan, Hiruni Kegalle, Rhea D’Silva, Ned Watt, Daniel Whelan-Shamy, Lida Ghahremanlou, and Liam Magee. Automating thematic analysis: How LLMs analyse controversial topics. *arXiv preprint arXiv:2405.06919*, 2024. doi: 10.48550/arXiv.2405.06919. URL <https://arxiv.org/abs/2405.06919>.
- Klaus Krippendorff. *Content analysis: An introduction to its methodology*. Sage publications, 2018.
- Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.
- Michelle S. Lam, Janice Teoh, James A. Landay, Jeffrey Heer, and Michael S. Bernstein. Concept induction: Analyzing unstructured text with high-level concepts using LLoM. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, 2024. doi: 10.1145/3613904.3642830. URL <https://doi.org/10.1145/3613904.3642830>.
- Sunghoon Ivan Lee, Eun Kyoung Choe, and Hee-Tae Jung. Data for: Envisioning the Use of In-Situ Arm Movement Data in Stroke Rehabilitation: Stroke Survivors’ and Occupational Therapists’ Perspectives, 2022. URL <https://doi.org/10.5064/F6L42GPJ>.

- Zongxia Li, Lorena Calvo-Bartolomé, Alexander Miserlis Hoyle, Paiheng Xu, Daniel Kofi Stephens, Juan Francisco Fung, Alden Dima, and Jordan Lee Boyd-Graber. Large language models struggle to describe the haystack without human help: A social science-inspired evaluation of topic models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7583–7604, 2025.
- Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pp. 74–81, 2004.
- Yvonna S. Lincoln and Egon G. Guba. *Naturalistic Inquiry*. Sage Publications, Beverly Hills, CA, 1985.
- Marina Meilă. Comparing clusterings by the variation of information. In *Learning Theory and Kernel Machines: 16th Annual Conference on Learning Theory and 7th Kernel Workshop, COLT/Kernel 2003, Washington, DC, USA, August 24-27, 2003. Proceedings*, pp. 173–187. Springer, 2003.
- Laura K. Nelson. Computational grounded theory: A methodological framework. *Sociological Methods & Research*, 49(1):3–42, 2020. doi: 10.1177/0049124117729703. URL <https://doi.org/10.1177/0049124117729703>.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.
- Angelina Parfenova, Andreas Marfurt, Jürgen Pfeffer, and Alexander Denzler. Text annotation via inductive coding: Comparing human experts to llms in qualitative data analysis. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 6456–6469, 2025.
- Rebecca Passonneau. Measuring agreement on set-valued items (MASI) for semantic and pragmatic annotation. In Nicoletta Calzolari, Khalid Choukri, Aldo Gangemi, Bente Maegaard, Joseph Mariani, Jan Odijk, and Daniel Tapias (eds.), *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC’06)*, Genoa, Italy, May 2006. European Language Resources Association (ELRA). URL <https://aclanthology.org/L06-1392/>.
- Xinyu Pi, Qisen Yang, and Chuong Nguyen. LOGOS: LLM-driven end-to-end grounded theory development and schema induction for qualitative research. *arXiv preprint arXiv:2509.24294*, 2025. URL <https://arxiv.org/abs/2509.24294>.
- Tingrui Qiao, Caroline Walker, Chris W. Cunningham, and Yun Sing Koh. Thematic-LM: A LLM-based multi-agent system for large-scale thematic analysis. In *Proceedings of the ACM on Web Conference 2025*, pp. 649–658. Association for Computing Machinery, 2025. doi: 10.1145/3696410.3714595. URL <https://doi.org/10.1145/3696410.3714595>.
- Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019.
- Katie Rose M Sanfilippo, Neta Spiro, Miguel Molina-Solana, and Alexandra Lamont. Do the shuffle: Exploring reasons for music listening through shuffled play. *PLoS One*, 15(2): e0228457, 2020.
- Hope Schroeder, Marianne Aubin Le Quéré, Casey Randazzo, David Mimno, and Sarita Schoenebeck. Large language models in qualitative research: Uses, tensions, and intentions. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, 2025a. doi: 10.1145/3706598.3713120. URL <https://doi.org/10.1145/3706598.3713120>.

- Hope Schroeder, Deb Roy, and Jad Kabbara. Just put a human in the loop? investigating LLM-assisted annotation for subjective tasks. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 25771–25795, Vienna, Austria, July 2025b. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.1323. URL <https://aclanthology.org/2025.findings-acl.1323/>.
- William A Scott. Reliability of content analysis: The case of nominal scale coding. *Public opinion quarterly*, pp. 321–325, 1955.
- Taggert Smith and Andy Smithwick. Technology concerns of adhd subreddits, 2026. URL <https://github.com/chuongnguyen26/HumanAlignment-Dataset>.
- Deanne Spek and Niko Vegt. Mapping Narratives on Mental Health: Data from a Scoping Review, Interview Study and Podcast Analysis, 2025. URL <https://doi.org/10.5064/F6E10MLI>.
- Sarah J. Tracy. Qualitative quality: Eight “big-tent” criteria for excellent qualitative research. *Qualitative Inquiry*, 16(10):837–851, 2010. doi: 10.1177/1077800410383121. URL <https://doi.org/10.1177/1077800410383121>.
- Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11(95):2837–2854, 2010. URL <http://jmlr.org/papers/v11/vinh10a.html>.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.
- Mian Zhong, Pristina Wang, and Anjalie Field. HICode: Hierarchical inductive coding with LLMs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 31060–31078. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.1580. URL <https://aclanthology.org/2025.emnlp-main.1580/>.

A Appendix

A.1 Limitations and Future Work

A.1.1 Limitations

QualAlign evaluates agreement with expert-produced artifacts; it does not assume that a qualitative corpus has a single uniquely correct interpretation. Human codebooks are useful reference points for quality, but disagreement can also reflect defensible differences in framing, abstraction, or theoretical commitments. Scores should therefore be read as measures of *artifact alignment* and consequent *human interpretability*, not as a complete measure of qualitative validity. The benchmark likewise does not measure novelty, downstream analytic usefulness, reflexivity, or transferability—all significant quality criteria for qualitative analyses Tracy (2010); Lincoln & Guba (1985).

The benchmark is modest in scale, though the largest of its kind that we are aware of: eight datasets, four systems, and domains concentrated in health, software work, information studies, and everyday technology use. Dataset provenance and annotation practice are necessarily heterogeneous. Excerpt boundaries range from complete short responses to coded quotations or conversational interview turns, and the reference codebooks contain either two or three hierarchical levels. This heterogeneity reflects the reality of real-world qualitative practice, but it limits causal claims about why one dataset or method is easier, hence our focus on the overall results relevant to automated qualitative systems rather than crowning an individual winner. A next-generation benchmark would benefit from adding more languages, disciplines, analytic traditions, longer documents, and prospectively collected datasets with independently replicated inductive human coding.

Our common deductive re-coder makes outputs from systems with different native interfaces comparable, but it is also a source of measurement error. Even when supplied with the human codebook, it falls well short of perfect recovery of expert assignments. The end-to-end assignment results therefore measure a pipeline containing both induction and re-coding. We expose this limitation with the deductive ceiling and the secondary ceiling-adjusted ratios in Appendix A.3.4; future work should evaluate stronger deductive coders, repeated draws, model ensembles, and human adjudication of difficult passages. In the meantime, we view this low ceiling as itself a significant finding for the state of automated deductive coding: given a complex qualitative label set, LLM coders do not reliably reproduce human code-excerpt assignments.

Code matching is another consequential modeling decision. Our simple algorithm is permissive and accommodates differences in granularity, while Hungarian alignment imposes a one-to-one correspondence. Embedding and assignment-overlap costs answer different questions, and the combined cost in Appendix A.4.4 is only one possible compromise. Future benchmarks could endeavor to estimate partial, many-to-many, or hierarchy-aware transport mappings rather than requiring a single correspondence heuristic.

Finally, semantic embeddings and an LLM judge are model-dependent. High label similarity can coexist with different operational use, and judge substitution changes some within-family rankings (Appendix A.4.2). We consequently report metric groups separately and avoid presenting a single overall leaderboard score.

A.1.2 Benchmark Usage Note

QualAlign is artifact-based: any system that produces a frozen codebook and (optionally) excerpt-level assignments can be scored through the same pipeline, and any expert qualitative analysis which publishes its research questions, codebook, and code-excerpt assignments can be added as a human reference. Although QualAlign does not require systems to produce code-excerpt assignments thanks to the built-in re-coder, we encourage future systems to attend to code-excerpt assignments as a first-class artifact alongside the codebook itself, as without a record of coding decisions, generated labels become difficult to explore, scrutinize, and rely on. The choice of re-coder versus native assignments should be kept consistent across systems being scored if cross-system comparison is the goal. Semi-

automated and human-in-the-loop systems can also be evaluated with QualAlign given a frozen, post-review codebook, and we encourage the comparison of these approaches in the future.

A.2 Dataset Details

A.2.1 Provenance

The benchmark reuses expert-coded artifacts released with eight prior qualitative studies. We preserve the smallest meaningful unit to which the source analysts attached a code and retain the source hierarchy where available. Dataset names used throughout this paper are short descriptive aliases. ADHD Reddit contains posts and comments about adults’ use of technology; Data Sharing contains scientist interviews about data practices (Donaldson & Koepke, 2022); Mental Health collects coded material concerning mental-health narratives (Spek & Vegt, 2025); Patient Deterioration contains healthcare-worker responses about recognizing ward deterioration (Beane et al.); Scrum Quality contains practitioner interviews about quality in Scrum teams (Alami & Krancher, 2022); Search Systems contains evidence-synthesis searcher accounts (Hickner, 2022); Shuffle Music contains free-text responses about music listening (Sanfilippo et al., 2020); and Stroke Wearables contains interviews concerning arm-use sensors in rehabilitation (Lee et al., 2022).

QualAlign compares one pooled artifact pair per dataset in the main analysis. This prevents Data Sharing and Scrum Quality—whose released codes are explicitly attributable to five and three separable questions, respectively—from receiving extra weight. Appendix A.4.1 restores those question-specific partitions as a robustness check.

A.2.2 Research Questions

Dataset	Research questions supplied to the systems
ADHD Reddit	RQ1: How do adults with ADHD currently use and appropriate existing technologies? RQ2: What unmet needs, frustrations, and preferences do they express regarding current technology use? RQ3: What non-technical practices or coping strategies might technology support in the future?
Data Sharing	RQ1: What data do scientists produce and use? RQ2: How do scientists manage data and data-management plans? RQ3: How do scientists perceive libraries and librarians as aids to managing and sharing data? RQ4: How aware are scientists of the FAIR principles, and how do they perceive them? RQ5: What do scientists need or want from data repositories?
Mental Health	RQ1: What master narratives about mental health emerge across recent interviews, podcasts, and literature? RQ2: How are emphases and perspectives within these narratives changing over time?
Patient Deterioration	RQ1: How do healthcare workers recognize deterioration of ward patients in resource-limited settings? RQ2: What organizational, cultural, and decision-making factors influence recognition?
Scrum Quality	RQ1: How do Scrum practitioners define software quality? RQ2: How do Scrum values, principles, and prescribed events or activities advance software quality? RQ3: Why do some Scrum teams fail to achieve higher software quality?
Search Systems	RQ1: How do platform and interface limitations shape systematic-searching workflows? RQ2: How and why do searchers select platforms or databases for evidence-synthesis projects?
Shuffle Music	RQ1: Why do people download and listen to music? RQ2: Which aspects of listening are prioritized when people discuss a track on their device?
Stroke Wearables	RQ1: How do stroke survivors and occupational therapists perceive the usefulness, concerns, and opportunities of wearable arm-use sensors? RQ2: What design considerations and data features would support integration into rehabilitation planning and follow-up?

Table 3: Research questions associated with the benchmark datasets.

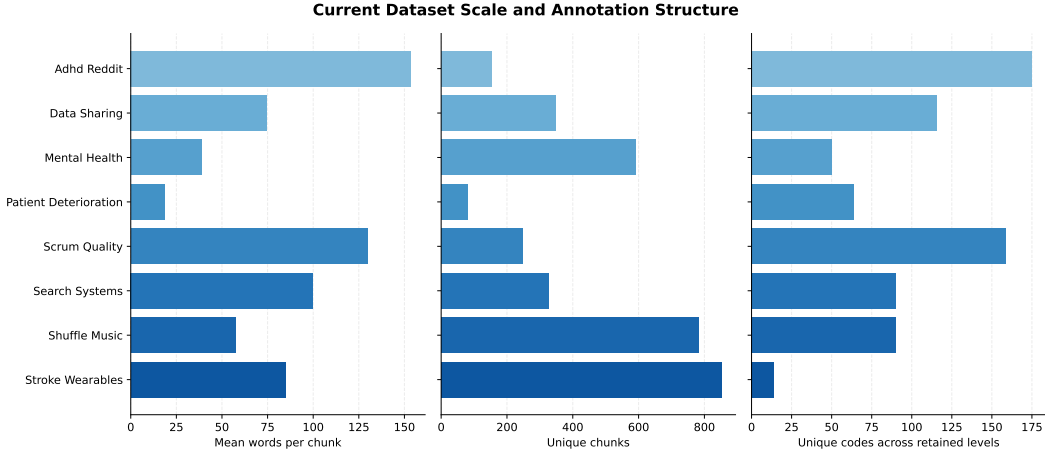


Figure 4: Dataset statistics regenerated directly from the current canonical human-reference mapping files. Panels show mean words per unique chunk, number of unique chunks, and distinct codes pooled across retained hierarchy levels.

A.2.3 Statistics and Semantic Properties

Dataset	RQs	Chunks	Codes	Words/chunk	Chunk div.	Doc. div.
ADHD Reddit	3	154	175	153.67	0.431	0.431
Data Sharing	5	349	116	74.65	0.405	0.028
Mental Health	2	591	50	38.69	0.433	0.215
Patient Deterioration	2	81	64	18.60	0.329	0.329
Scrum Quality	3	249	159	129.70	0.326	0.104
Search Systems	2	327	90	100.01	0.349	0.052
Shuffle Music	2	784	90	57.53	0.322	0.322
Stroke Wearables	2	852	14	84.81	0.367	0.055

Table 4: Current benchmark statistics. “Codes” counts distinct labels pooled across retained hierarchy levels. Semantic diversity is mean cosine distance from the corresponding embedding centroid.

We embed excerpts with qwen3-embed-0.6b. Chunk diversity is the mean cosine distance of chunk embeddings from the dataset centroid. Document diversity first averages all chunk embeddings belonging to a source document and then applies the same calculation. The two views separate within-document topical variation from variation between documents. Mental Health has the highest chunk diversity (0.433), whereas ADHD Reddit has the highest document diversity (0.431). Data Sharing has broad variation among turns but low between-document diversity because its 349 turns come from only five long interviews.

Figure 5 gives the requested granularity comparison. Every method differs from the human codebook in cardinality on every evaluated dataset slice in the original question-level analysis. HICode and LLoM predominantly fragment the reference into more labels, ThematicLM predominantly compresses it, and LOGOS is intermediate. Cardinality is only a proxy—two equally sized codebooks can still differ structurally—but the direction and magnitude explain why a one-to-one matcher can be punitive.

A.3 Supplementary Metrics and Diagnostic Views

A.3.1 Supplementary Metric Groups

The main text retains a deliberately small, nonredundant set. Two groups appear only in the appendix. The *interpretive* group contains LLM-as-a-judge alone: it asks whether matched labels express the same qualitative concept under a rubric, rather than treating the judge as another embedding similarity. The *structural* group contains tree edit distance (TED) alone: it compares organization of the codebook hierarchy, rather than wording or excerpt

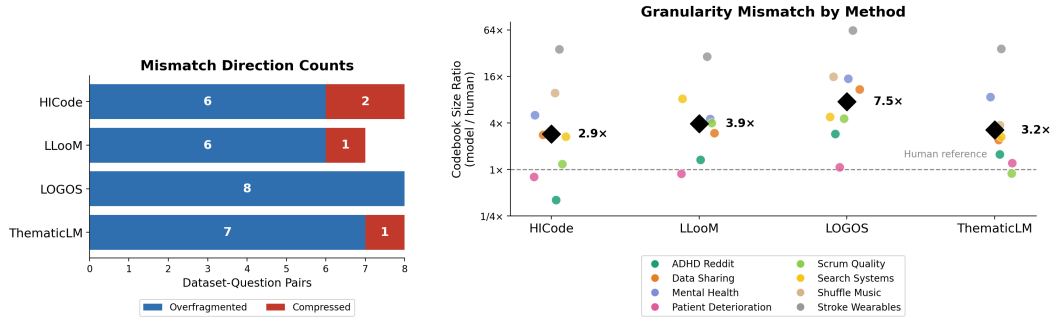


Figure 5: Granularity comparison. Left: counts of overfragmented and compressed method–dataset–question codebooks. Right: generated-to-reference codebook-size ratios, with diamonds marking medians.

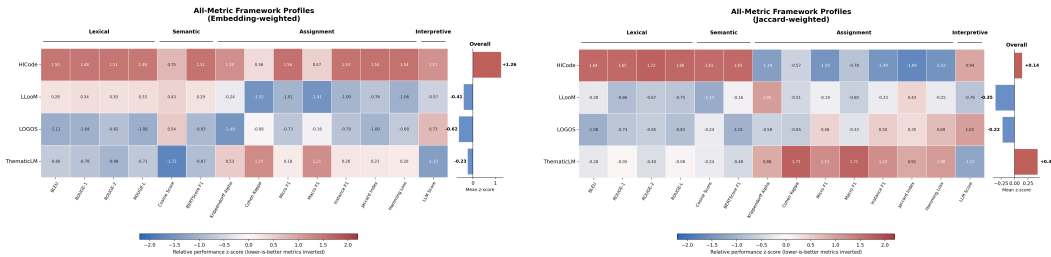


Figure 6: All available metrics by framework. Left: embedding-weighted matching. Right: Jaccard-weighted matching.

assignments. TED is reported as a distance (lower is better); its normalized similarity display is explicitly identified when used in figures.

The assignment group contains several related but nonidentical views: instance-, micro-, and macro-F1; Jaccard index over the aligned multilabel outputs; mean pairwise Jaccard similarity over matched label coverage sets; nominal multiclass and binary Krippendorff’s α ; Cohen’s κ ; Hamming loss; and partition measures where applicable. Instance-F1 weights excerpts equally, micro-F1 pools label decisions, and macro-F1 weights labels equally. The two Jaccard variants consequently should not be conflated: one compares each excerpt’s aligned label set and the other averages overlap between matched codes’ excerpt sets. Binary α treats each aligned code–excerpt decision as present/absent; nominal α operates on the benchmark’s collapsed aligned assignments. Their shared grouping means only that all evaluate *where* codes were applied.

A.3.2 Clustering and Simple LLM Baselines

These baselines ask what complexity buys in terms of performance. The non-LLM floor clusters excerpt text directly using TF-IDF k -means, Sentence-BERT k -means, and BERTopic (UMAP followed by HDBSCAN). For each dataset and hierarchy level, k -means receives the true number of human codes at that level; we use three random seeds. Cluster IDs are scored natively against expert assignments with Jaccard-based best-match and Hungarian alignment. This is an intentionally favorable cardinality-controlled partition floor, but it is *not* on the same scale as the main system comparison: it compares native cluster assignments with original human assignments and bypasses the shared LLM re-coder. BERTopic determines its own effective cluster count and treats noise points according to the released implementation.

The simple LLM comparator uses the same model and corpus access as the benchmark systems but requests a flat codebook in a single prompt, without iterative synthesis, retrieval, agent coordination, or framework-specific hierarchy induction. It is then passed through

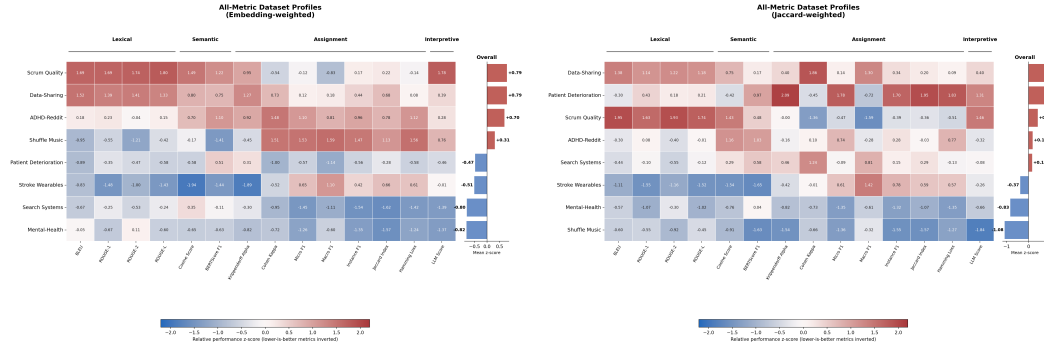


Figure 7: All available metrics by dataset. Left: embedding-weighted matching. Right: Jaccard-weighted matching.

Baseline	Level	Weight	Instance Jaccard	Pair Jaccard	Kripp. α
TF-IDF k -means	Theme	Jaccard	.391	.229	.108
		Emb	.256	.158	-.118
	Leaf	Jaccard	.191	.277	.088
		Emb	.057	.072	-.009
Sentence-BERT k -means	Theme	Jaccard	.414	.257	.181
		Emb	.231	.148	-.100
	Leaf	Jaccard	.197	.289	.108
		Emb	.066	.078	.007
BERTopic/HDBSCAN	Theme	Jaccard	.389	.277	.084
		Emb	.214	.173	-.131
	Leaf	Jaccard	.129	.287	.021
		Emb	.044	.100	-.041
Four systems (mean)	Theme	Jaccard	.593	.071	.360
		Emb	.316	.027	-.005
	Leaf	Jaccard	.348	.250	.229
		Emb	.121	.071	.043

Table 5: Native-assignment non-LLM clustering floor under best-match alignment, Jaccard-weighted and embedding-weighted, shown against the four induction systems’ scores at matching hierarchy levels for reference. Clustering Jaccard rows average eight datasets and three seeds (24 runs); clustering Emb rows use one representative seed (eight runs), since embedding-matched scoring requires the additional cluster-naming step. System rows average HICode, LLoM, LOGOS, and ThematicLM re-scored at each level through the standard shared-re-coder pipeline (§A.3.3)—a different, recoder-mediated comparison from the clustering floor’s native assignments, so the two blocks calibrate rather than directly rank against each other. The k -means variants are given the reference code count at the evaluated level; BERTopic/HDBSCAN selects its own effective cluster count.

the same shared deductive re-coder and evaluation stack as the four systems. Thus the clustering floor tests whether assignment structure is recoverable from geometry alone, while the simple-prompt baseline tests whether framework orchestration improves on direct LLM induction. We use these as calibration points, not additional competitors: complexity is valuable only to the extent that a framework improves semantic, structural, or assignment alignment beyond these simpler paths. Because clusters are unnamed, lexical and judge metrics are not reported for the clustering floor; to obtain an embedding-matched view as well, we additionally name every cluster from one representative seed with a single batched LLM call per run (one call names every cluster produced by that run) and embed the resulting labels locally, letting us recompute best-match and Hungarian alignment under embedding cost alongside the Jaccard-weighted view.

Sentence-BERT k -means is strongest on the coarse theme view under Jaccard weighting (instance Jaccard .414, $\alpha = .181$), demonstrating that some broad organization is available from embedding geometry without a qualitative-analysis pipeline. At the leaf level, however, its instance Jaccard falls to .197 and α to .108. BERTopic reaches a comparable instance

Jaccard to the fixed- k methods at the theme level (.389) but the lowest theme-level α of the three (.084), consistent with density-based clusters not aligning as cleanly with a coarse, small- k partition as the fixed-cardinality methods; its effective cluster count and outlier rate also vary by seed, since HDBSCAN is adaptive rather than fixed- k . Complexity therefore buys most when the target is a named, fine-grained, and hierarchically organized codebook rather than a coarse partition. The floor is not directly rank-comparable with the four systems because it bypasses re-coding and, for k -means, receives oracle cardinality—but placing the two side by side is still informative about *where* that gap actually lives.

The comparison against the four systems is metric-dependent rather than uniform. On instance Jaccard and Krippendorff’s α , the systems are clearly ahead of the clustering floor at both levels and under both matching weights: theme instance Jaccard reaches .593 for the systems against .389–.414 for the floor, and theme α reaches .360 against .084–.181; the gap narrows but persists at the leaf level (.348 vs. .129–.197 instance Jaccard; .229 vs. .021–.108 α). On pair Jaccard, however, the floor is competitive to clearly ahead: it edges out the systems at the leaf level (.277–.289 vs. .250) and dramatically exceeds them at the theme level (.229–.277 vs. only .071). We attribute this to oracle cardinality rather than superior partitioning: the systems induce many more codes than there are human themes, so best-match assigns several narrow system codes to each broad theme; the *union* of those narrow codes’ excerpts still covers the theme well (supporting high instance Jaccard and α), but each individual matched *pair* has a small system code competing against a large human theme, depressing the per-pair overlap that pair Jaccard averages over. The clustering floor’s k -means variants, handed the true theme count directly, produce one cluster per theme with no such many-to-one mismatch, so their pair-level overlap stays high even though their aggregate multi-label coverage and chance-corrected agreement remain well below the systems’. The same ranking holds under embedding-weighted matching, reinforcing that this is a cardinality effect rather than an artifact of one matching rule. Embedding-weighted matching also exposes an asymmetry in label quality: the clustering floor’s α turns negative under embedding weighting almost everywhere (all combinations except Sentence-BERT/leaf), while the four systems retain small positive or near-zero α at both levels. One-shot LLM cluster naming from raw excerpt samples evidently paraphrases into human-like codebook vocabulary less reliably than the systems’ induction pipelines, which are prompted and iterated specifically toward producing legible, analyst-facing labels.

A.3.3 Hierarchical Analysis

Pooling all hierarchy levels can make a method appear misaligned merely because it expresses a related concept at a different depth. We therefore repeat end-to-end scoring after restricting both the human and generated artifacts to top/theme, intermediate (where available), and leaf/code levels. Four datasets expose three substantive levels and four expose two.

For each level, we filter label columns before matching, retain the same excerpt universe, and recompute both best-match and Hungarian configurations. We also recompute the human-codebook deductive ceiling at each level, rather than dividing by a pooled ceiling. This analysis distinguishes two hypotheses: improvement at coarse levels indicates that systems recover broad themes but miss fine distinctions; uniformly weak scores indicate different excerpt partitioning even after granularity is controlled. Level-specific scores should not be compared solely by magnitude because label prevalence and the number of available codes change with depth; within-level method-to-ceiling gaps are the relevant comparison.

Best-match Krippendorff’s α rises monotonically from the pooled default ($\alpha = .102$) through leaf ($\alpha = .229$), mid ($\alpha = .291$), to top ($\alpha = .359$); instance Jaccard follows the same ordering. This is consistent with the hypothesis that a meaningful share of the benchmark’s reported disagreement is a granularity mismatch rather than a wholesale failure to recover topical structure. The deductive ceiling exhibits the same ordering at every level (Jaccard α : pooled .067, leaf .161, mid .235, top .288), so part of the coarse-level gain is attributable to the shared re-coder rather than codebook induction alone; end-to-end scores remain below the ceiling at every level, so neither stage alone explains the pattern. Embedding-weighted matching

Level	Weight	Instance Jaccard	Pair Jaccard	Kripp. α
Top	Jaccard	.594	.071	.359
	Emb	.317	.027	-.005
Mid	Jaccard	.464	.129	.291
	Emb	.198	.052	.043
Leaf	Jaccard	.349	.250	.229
	Emb	.122	.071	.043
Pooled default (§4)	Jaccard	.259	.258	.102
	Emb	.101	.071	.008

Table 6: End-to-end assignment agreement under best-match alignment by hierarchy level, averaged across the four systems and every dataset exposing that level (top/leaf: all eight datasets; mid: the four datasets with a genuine third level). The pooled row repeats the same union-of-available-levels default used elsewhere in this benchmark, shown here as a reference point rather than a verified identical recomputation.

tells a starker story: α is at or below zero at the top level (−.005) and only reaches .043 at mid and leaf, even though instance Jaccard is comparatively high at top (.317)—coarse embedding-matched pairs recover broad topical overlap without chance-corrected reliability, echoing the semantic/assignment gap reported in §4 at every granularity, not only in the pooled view.

To separate a genuine coarse-level advantage from the mechanical fact that fewer, broader label categories are easier to match by chance, we additionally computed a permutation null at each level: for each (method, dataset, level, matching, weight) cell we shuffled which chunk received which human label at that level, holding the label frequency distribution fixed, and rescored, repeating 15–50 times per cell. Under best-match Jaccard weighting, observed Krippendorff’s α clears its own null band by a wide margin at every level (top: .359 vs. null mean .111, 95% range [.072, .149]; mid: .291 vs. .070, range [.055, .085]; leaf: .229 vs. .117, range [.105, .129]), so the coarse-level improvement is not simply an artifact of label-space shrinkage: systems recover above-chance structure at every level tested. The margin over the null narrows from top (.248) to leaf (.112), so roughly half of the raw top-versus-leaf gap in α tracks a genuine increase in above-chance alignment at coarser granularity, and the remainder tracks a chance-agreement floor that itself rises as the label space shrinks. The null band is not monotonic across levels (leaf’s null mean, .117, exceeds mid’s, .070), which we attribute to level-specific differences in label cardinality and marginal frequency across datasets rather than a systematic level effect.

A.3.4 Ceiling-Adjusted End-to-End Agreement Ratios

This is an alternate view that attempts to isolate the contribution of the induced schema to the , subtracting out the noise introduced by the shared re-coder. Let $s_{m,d,k}$ denote an end-to-end score for method m , dataset d , and eligible higher-is-better assignment metric k , and let $c_{d,k}$ be the corresponding deductive score obtained when the re-coder is supplied the human codebook. We report

$$R_{m,d,k} = \min\left(\frac{s_{m,d,k}}{c_{d,k}}, 1\right). \quad (1)$$

For nonnegative lower-is-better losses, the orientation is reversed, $R = \min(c/s, 1)$. A ratio of one means that the full system reaches or exceeds the observed deductive pipeline ceiling; it does not mean human-level or perfect performance.

This ratio is relegated to an auxiliary view because division magnifies noise when c is small, has no coherent interpretation when a chance-corrected ceiling is zero or negative, and can make a tiny absolute difference look large. The main text therefore presents raw scores alongside the deductive baseline, which is easier to audit, and the codebook-effect scores, which have a clearer causal interpretation. Ratios are limited to bounded nonnegative extensional measures; signed α and κ remain unadjusted.

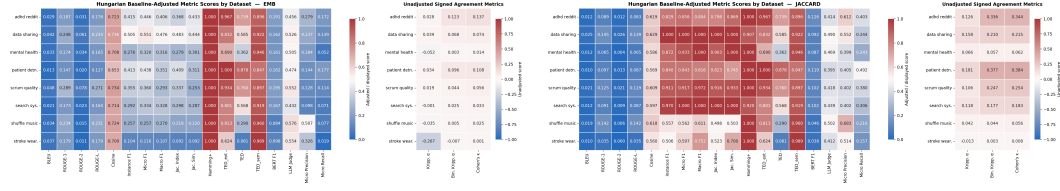


Figure 8: Secondary ceiling-adjusted ratios from the canonical pooled lineage under Hungarian matching. Left: embedding cost. Right: Jaccard cost. Eligible extensional measures are divided by dataset-specific deductive ceilings; signed chance-corrected coefficients remain unadjusted.

Dataset	Method	Kripp. α		Pair Jaccard		Cosine	
		Q-cond.	Pooled	Q-cond.	Pooled	Q-cond.	Pooled
Data Sharing	HICode	.044	.065	.128	.075	.769	.776
	LLoM	.094	.037	.130	.065	.724	.741
	ThematicLM	.003	.040	.076	.090	.736	.738
Scrum Quality	HICode	.048	.053	.126	.088	.801	.793
	LLoM	.070	.043	.151	.072	.761	.788
	ThematicLM	-.041	.019	.081	.070	.736	.753

Table 7: Question-conditioned versus pooled results under embedding-based best-match alignment. Question-conditioned values average five Data Sharing or three Scrum Quality question runs from the latest archive. LOGOS is omitted because a like-for-like current pooled artifact is unavailable. Conditioning sometimes improves direct assignment overlap, especially for LLoM, but effects on chance-corrected agreement and semantic similarity are mixed.

A.4 Robustness Checks and Alternate Matching Strategies

A.4.1 Question Conditioning

The main analysis pools questions within a dataset. As a robustness check, we use the most recent archived question-conditioned run (July 1, 2026) for the two datasets whose released codes are explicitly assigned to separable questions: Data Sharing (five questions) and Scrum Quality (three). Each system receives only the focal question and is evaluated against codes assigned to that question; scores are then averaged across questions within a dataset so the datasets retain equal weight. Table 7 compares these question-conditioned averages with the current pooled run for configurations available in both lineages.

The robustness result is consequently not that conditioning uniformly raises scores. It reduces irrelevant-code competition and often raises pairwise assignment overlap, but smaller question-specific codebooks and changing prevalence can lower chance-corrected statistics. Most importantly, conditioning does not close the gap between moderate label semantics and weak assignment agreement.

A.4.2 Judge Substitution and Exact Prompt

We substitute gpt-oss-120b and Qwen/Qwen3.5-35B-A3B while holding matched pairs and the rubric fixed. Scores correlate at $r = .567$ with mean absolute difference .061. The top-two ordering changes, supporting our decision not to treat judge scores as model-invariant rankings.

The exact prompt is reproduced below; pairs are appended as PAIR_i, Text 1: ..., and Text 2: Inference uses batches of 25 and requires one numeric line per pair.

You are evaluating semantic alignment between qualitative code labels.
For each pair, rate how closely the two labels express the same underlying concept, regardless of wording.
Scoring rubric:

λ	Instance Jaccard	Micro-F1	Pair Jaccard	Pair cosine
.0	.246	.366	.305	.599
.1	.246	.366	.305	.614
.2	.244	.363	.303	.623
.3	.238	.354	.299	.636
.4	.227	.341	.293	.647
.5	.217	.327	.282	.660
.6	.199	.302	.264	.674
.7	.171	.263	.239	.688
.8	.143	.224	.203	.700
.9	.109	.175	.148	.710
1.0	.075	.123	.091	.713

Table 8: Current combined-cost Hungarian sweep. λ is the cosine weight and $1 - \lambda$ the excerpt-set Jaccard weight. Values are raw, dataset-balanced means.

- 1.0 = same concept or clear paraphrase
 - 0.8 = highly similar meaning, with only minor difference in emphasis
 - 0.5 = partial overlap; related but not equivalent
 - 0.2 = same broad topic but substantially different concept
 - 0.0 = unrelated meanings
- Important instructions:
 Compare each pair independently.
 Return exactly one line per pair in the format PAIR_<index>=<score>.
 Do not include any explanation or extra text.

A.4.3 Hungarian Matching

The default matching algorithm (*best-match*) maps every generated code independently to its most similar human code, allowing several generated codes to share a target. *Hungarian matching* instead solves a rectangular one-to-one linear assignment problem (Kuhn, 1955) to minimize a cost function across the entire set of code pairs. For generated codes g_i and human codes h_j , embedding-cost matching uses

$$S_{ij}^{\text{emb}} = \max(0, \cos(e(g_i), e(h_j))), \quad C_{ij} = 1 - S_{ij}. \quad (2)$$

The solver returns $\min(n, m)$ pairs and leaves surplus codes unmatched. No dummy rows are required by the implementation. Jaccard-cost matching substitutes $S_{ij}^{\text{jac}} = |A_i \cap B_j| / |A_i \cup B_j|$, where A_i and B_j are assigned-excerpt sets.

Hungarian matching is stricter because a broad reference code cannot absorb many generated fragments. It is useful for assessing global codebook correspondence, but unmatched codes are excluded from pair-only label metrics and must be interpreted alongside cardinality diagnostics. Best-match is therefore the main assignment view; Hungarian results test whether the conclusion depends on the more charitable many-to-one mapping.

A.4.4 Combined Cost Function

Embedding matching can pair semantically similar labels used on different excerpts; Jaccard matching can pair differently named labels used on the same excerpts. A combined similarity is more charitable to pairs that are strong overall without being the single best pair under either diagnostic:

$$S_{ij}^{\text{comb}}(\lambda) = \lambda S_{ij}^{\text{emb}} + (1 - \lambda) S_{ij}^{\text{jac}}, \quad C_{ij}^{\text{comb}} = 1 - S_{ij}^{\text{comb}}. \quad (3)$$

We reran the complete grid $\lambda \in \{0, .1, \dots, 1\}$ on the 32 available method–dataset artifacts. Table 8 averages methods within each dataset and then weights the eight datasets equally. We see the expected tradeoff: increasing the cosine weight improves the semantic quality of selected pairs while reducing their extensional overlap.

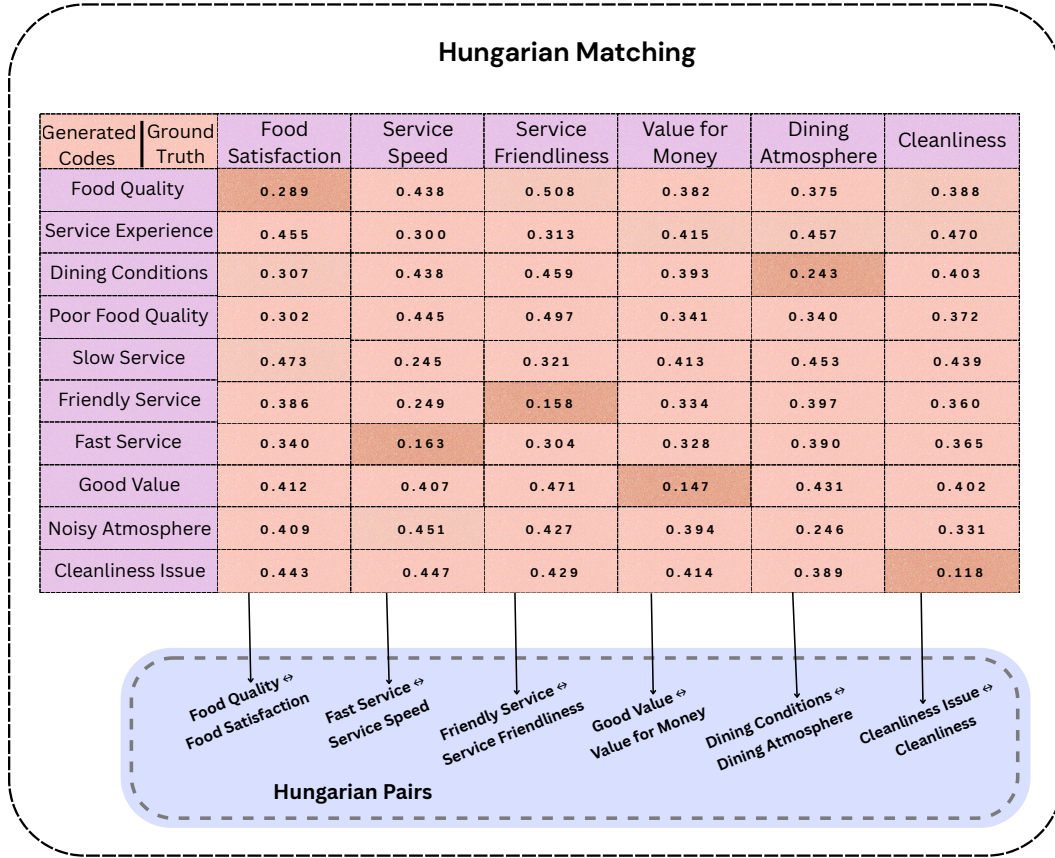


Figure 9: Rectangular Hungarian matching. Highlighted cells form the globally optimal one-to-one assignment; surplus codes on the larger side remain unmatched.

The blend is therefore potentially more charitable, but “best” depends on which correspondence is being privileged. Selecting λ using the same semantic or assignment outcomes later reported would bake the desired conclusion into matching. We retain the two endpoints in the main text because they remain more diagnostic and reproducible, and present the grid only as a sensitivity analysis and an offering for future benchmark users to tune the matching according to their priorities and subjective quality criteria.

A.5 AI Usage Disclosure

LLMs were used to assist with related-literature discovery and manuscript editing for grammar and presentation. They were also used to support code writing for analysis, table preparation, and figure preparation.