Self-reflection in Automated Qualitative Coding

Improving Text Annotation through Secondary LLM Critique

Zachary Dunivan[1], Mobina Noori[2], Seth Frey[2], Curtis Atkinson[3]

[1] U Stuttgart

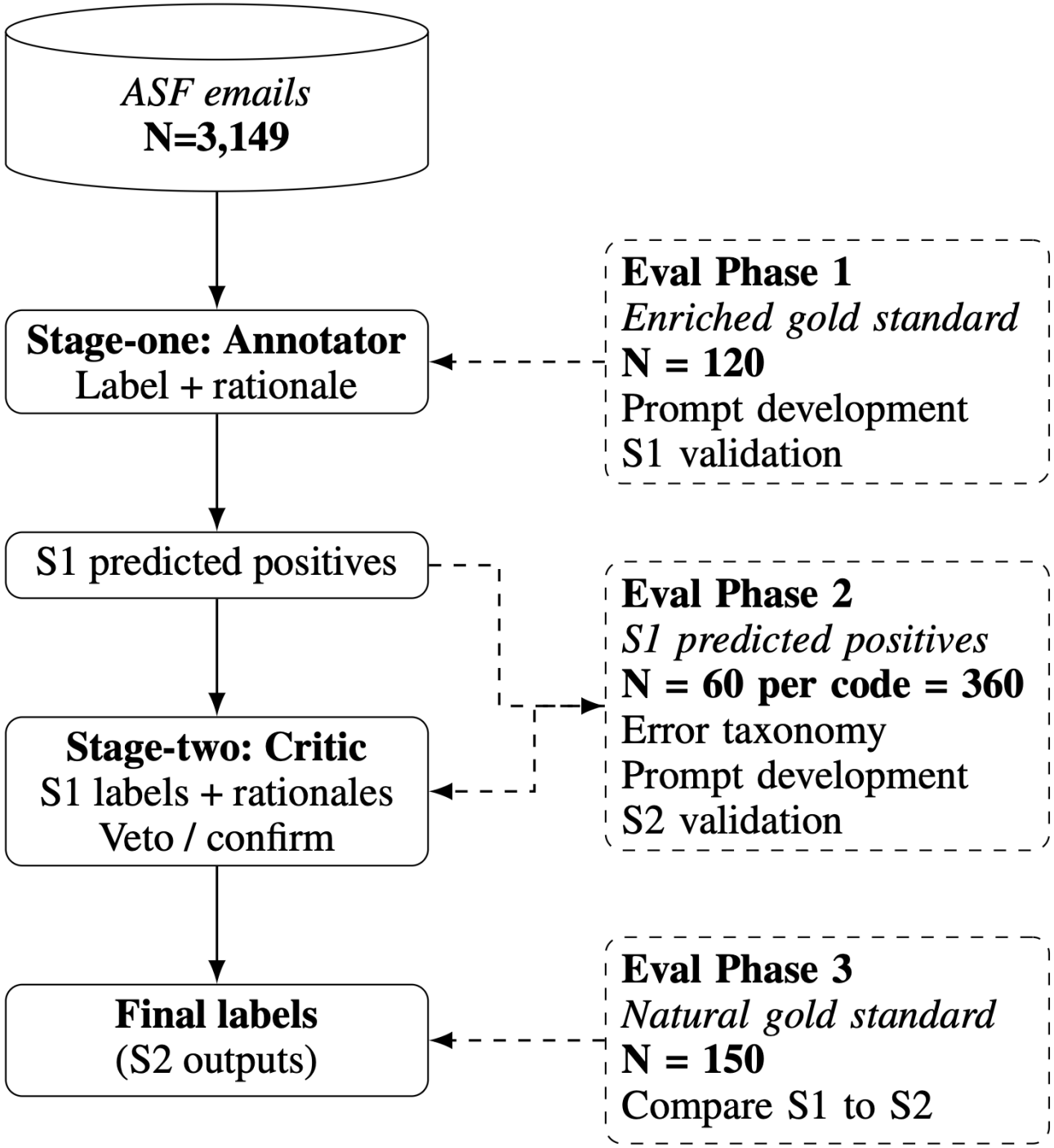
[2] UC Davis

[3] U Washington

Motivation and Problem Statement

- Problem:
  - Unreliable LLM-assigned codes in the annotation wild-west (Tornberg, 2024)
  - Models are eager to please, leading to high number of FPs
- Gap:
  - How to address structured errors made by a calibrated annotation model for substantive use
- Hook:
  - Proposes a method to make nuanced qualitative codes more usable at scale
  - Contributes to an emerging hermeneutic turn in computational text analysis

Two-Stage Annotation Pipeline



Methodology

- Step 1: Codebook development and first-stage coding
- Step 2: Error analysis: developing a classification schema for FPs
- Step 3: Prompting strategy for secondary LLM critique
- Step 4: Full pipeline evaluation design

Codebook development and first-stage coding

- RQ: How senior members of the Apache Software Foundation evaluate projects success?
- Thematic analysis of 16 semi-structured interviews with ASF members
- Six qualitative codes emerged
- Adapted codebook for GPT-4o analysis (Dunivan, 2025):
  - Tightening definitions and scope
  - Adding boundary clauses and negative examples
- Two experts independently labeled a gold-standard set of 120 emails

Error analysis

- One author audited a random sample of 60 stage-one positive passages per code and assigned an error class when invalid.
- Multiple rounds of inductive coding yielded two recurrent error classes:
  - Meta-discussion (use vs mention problem)
  - Misinterpretation (missing exclusion criteria)

Prompting for secondary LLM critique

- Sufficiency rule: Critic overturns the positive only if no cited justification remains valid after re-evaluation
- Three-layered prompt:
  - Task framing
  - Decision Policy (Error taxonomy, Sufficiency rule)
  - Input/output contract
- Evaluate performance on a new gold-standard set of 150 passages randomly sampled from the final corpus

Results

	Detected Positive Rate			Cohen's κ			F1		
Theme	**Gold**	**S1**	**S2**	**S1**	**S2**	**Δ**	**S1**	**S2**	**Δ**
Community Vitality	0.21	0.25	0.25	0.90	0.89	−0.01	0.92	0.91	−0.01
Corporate Involvement	0.09	0.09	0.09	0.93	0.93	0.00	0.94	0.94	0.00
Cultural Alignment	0.11	0.16	0.11	0.80	0.87	0.07	0.83	0.88	0.05
Mentor Engagement	0.04	0.05	0.03	0.53	0.78	0.26	0.55	0.79	0.25
Policy Compliance	0.29	0.33	0.30	0.76	0.82	0.06	0.83	0.87	0.04
Technical and Market	0.05	0.04	0.03	0.50	0.68	0.19	0.52	0.69	0.18

Eval Phase 3: Classification Performance on 150 Randomly Sampled Passages

Discussion Points

- Specific division of labor between the two stages of the pipeline:
  - Stage 1: Annotator model optimizes for recall
  - Stage 2: Critic model optimizes for precision
- Bounded autonomy: Self-reflection instrumented and directed by the research team
- Hermeneutic value: LLMs helped the authors identify ambiguities in the codebook
- Compute efficiency: Critic only runs on positives

Thoughts on the Paper

- Multi-stage evaluation process revealed important insights
- No acknowledgement of limitations
- Critic relies on human-crafted error taxonomy, same base model
- Error taxonomy may vary with the task, generalizability not demonstrated
- Does not focus on coming up with initial codes
- Manual taxonomy construction a bottleneck
- Gains concentrated in weak codes