COSI-230B: Natural Language Annotation for Machine Learning

Lecture 13: Quality Control

Jin Zhao

Brandeis University

Spring 2026

Today's Agenda

1. The promise and peril of QC metrics
2. Limits of inter-annotator agreement
3. Gold data pitfalls
4. Annotator modeling (conceptual overview)
5. Metrics vs. error inspection
6. LLMs as QC tools
7. Building a QC pipeline

Theme: Metrics can mask failure — learning to see past the numbers.

Why Quality Control Matters

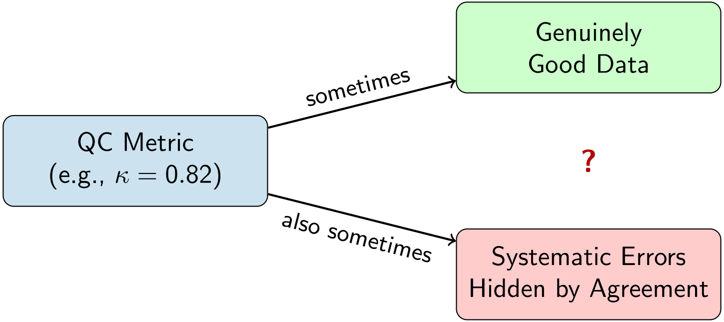
The core problem:

- You build a dataset. You report high agreement. You publish.
- Two years later, someone discovers systematic errors in 15% of your data.
- Every model trained on it inherited those errors.

The Uncomfortable Truth

High IAA does not mean your data is correct.
It means your annotators *agree* — possibly on the wrong answer.

The QC Illusion



The number alone does not tell you which path you are on.

Limits of Inter-Annotator Agreement

IAA is necessary but not sufficient.

Five ways IAA can mislead you:

1. **Shared bias** — annotators trained together may replicate the same errors
2. **Easy-item inflation** — agreement is high because most items are trivial
3. **Category collapse** — annotators avoid hard categories, boosting agreement
4. **Prevalence effects** — skewed distributions inflate chance-corrected metrics
5. **Guideline memorization** — annotators learn to “pass the test” rather than annotate well

Shared Bias: When Everyone Is Wrong Together

Scenario: Sentiment annotation for product reviews.

Review Text	Ann. A	Ann. B	Actual
"Not bad at all, really"	Negative	Negative	Positive
"Could have been worse"	Negative	Negative	Positive
"I didn't hate it"	Negative	Negative	Neutral
"Surprisingly adequate"	Negative	Negative	Positive

Result

$\kappa = 1.0$ — Perfect agreement. Completely wrong on negation and hedging.

Easy-Item Inflation

Scenario: NER annotation on 1,000 sentences.

Distribution:

- 850 sentences: no entities (trivial)
- 100 sentences: clear entities (easy)
- 50 sentences: ambiguous entities (hard)

Agreement breakdown:

- No-entity: $\kappa = 0.98$
- Clear entities: $\kappa = 0.90$
- Ambiguous: $\kappa = 0.35$

Reported vs. Real

Overall $\kappa = 0.91$ — looks great!
But 50 items that actually matter for model robustness have $\kappa = 0.35$.

Category Collapse

When annotators silently abandon hard distinctions

Category	Guidelines Expect	Actual Usage
Positive	30%	42%
Negative	30%	41%
Mixed	20%	3%
Neutral	20%	14%

- “Mixed” is hard to apply consistently, so annotators default to Positive/Negative
- Agreement goes *up* because fewer categories = fewer chances to disagree
- But the schema is no longer measuring what you designed it to measure

Gold Data: Pitfalls and Better Practices

Common pitfalls:

1. **Circular creation** — project lead writes guidelines *and* gold; evaluates annotators against their own interpretation
2. **Stale gold** — guidelines evolve but gold items remain from Round 1
3. **Unrepresentative gold** — hand-picked for clarity, missing hard cases
4. **Memorized gold** — repeated items become recognized, not evaluated

Anti-Pattern: Static & Easy Gold

Same 20 items for 6 months. Annotators memorize them. Gold accuracy = 99%. Real accuracy = unknown.

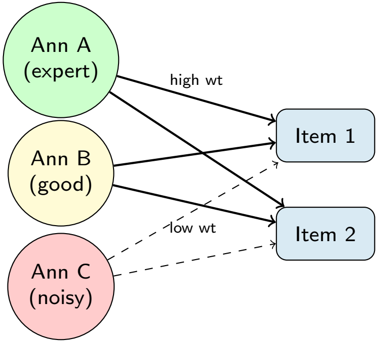
Better Practice

Rotating gold: Refresh quarterly; do not reuse items.

Stratified gold: Mirror the difficulty distribution of real data; include known edge cases.

Annotator Modeling: Beyond Majority Vote

The idea: Not all annotators are equal, and not all disagreements are noise.



Three dimensions to model:

1. **Reliability** — consistency with expert/consensus labels
2. **Bias** — systematic tendencies (e.g., always labels borderline as positive); predictable and correctable
3. **Item difficulty** — genuinely ambiguous items produce expected disagreement

Practical Takeaway

Even conceptually, this changes how you interpret disagreement.

Majority Vote vs. Annotator Modeling

Majority Vote	Annotator Modeling
All annotators weighted equally	Weights reflect reliability
Disagreement = noise to eliminate	Disagreement = signal to interpret
One "correct" label per item	Can preserve legitimate ambiguity
Simple, transparent	More complex, requires tuning
Ignores annotator-specific patterns	Can detect and correct for bias

When to consider annotator modeling:

- Large annotator pools (crowdsourcing)

What Error Inspection Catches That Metrics Miss

Metrics summarize. Inspection reveals.

What metrics tell you:

- Overall agreement level
- Per-category F1 or κ
- Whether you "pass" a threshold

What inspection tells you:

- *Which* items cause disagreement
- Whether errors are random or systematic
- If certain annotators struggle with specific categories
- If guidelines are ambiguous on particular constructions

Rule of Thumb

For every QC metric you compute, look at at least 50 of the items behind it.

Discussion: When Metrics Mislead

Consider this scenario:

You compute $\kappa = 0.78$ on a sentiment annotation task with 3 annotators and 500 items. The metric passes your threshold ($\kappa \geq 0.70$).

Questions to Consider

1. What specific patterns could be hiding behind that 0.78?
2. If you could only inspect 50 items, which 50 would you choose and why?
3. How would you determine whether disagreements reflect guideline gaps vs. genuine ambiguity?
4. What would change your confidence that the data is actually good?

Key insight: The κ may "pass" even with systematic problems (e.g., all annotators mislabel sarcasm the same way). Disagreement often clusters around specific *types* of items.

LLMs as QC Tools: The New Frontier

Can LLMs help with quality control?

Three promising applications:

1. **Detecting inconsistencies**
   - Flag items where the label seems inconsistent with similar items
2. **Flagging edge cases**
   - Identify items that are likely to cause disagreement
3. **Generating QC reports**
   - Summarize patterns in disagreements across the dataset

But Also: Limitations

LLM-based QC inherits the training biases of the LLM itself. We will discuss this.

LLM QC in Practice: Inconsistency Detection & Edge Cases

Inconsistency detection:

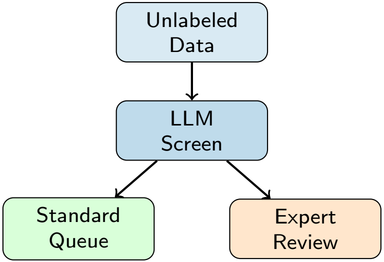
Example

"Here are 5 reviews labeled Negative. Review #3 seems different. Is the label consistent?"

#3: "I wasn't expecting much, but it actually works great." — Label: Negative

- Cluster items by label; ask LLM to find outliers
- Use LLM confidence as proxy for item difficulty

Edge case flagging (pre-annotation):



- Route hard items to experts
- Catch guideline gaps *before* annotation

Not a replacement for human review — a triage tool to focus human attention.

The Bias Problem: Why LLM QC Is Not Neutral

LLM-based QC inherits the biases of the LLM's training data.

Documented risks:

- LLMs may flag valid labels as "inconsistent" if they conflict with the LLM's own biases
- Cultural or dialectal variation may be marked as errors
- Subjective categories reflect the LLM's training distribution, not your schema

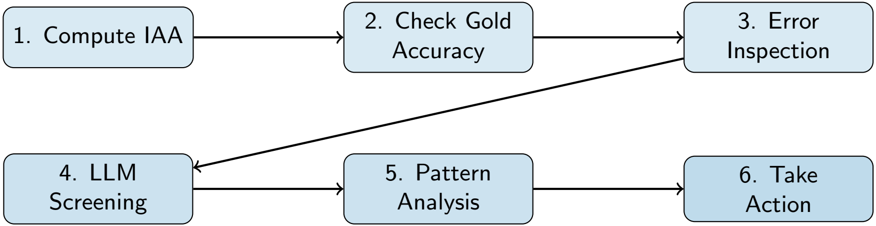
Example:

- Task: Toxicity annotation
- Human labels African American English (AAE) text as non-toxic
- LLM flags this as "inconsistent" because AAE patterns are over-represented in toxic-labeled training data

Principle

LLM QC should **surface** items for human review, never **override** human labels.

Putting It Together: A QC Pipeline



Revise guidelines? Retrain? Remove annotator?

Key principles:

- No single metric is sufficient — use multiple QC signals
- Always include manual inspection alongside automated metrics
- QC must be ongoing, not a one-time check
- Document what you checked *and what you did not check*

What QC Cannot Tell You

Even the best QC pipeline has blind spots.

1. **Whether your schema is right**
   - QC measures consistency with your schema, not whether the schema captures reality
2. **Whether your data represents the target domain**
   - Perfect annotations on unrepresentative data = useless annotations
3. **Whether annotators understood vs. memorized**
   - High gold accuracy can reflect memorization, not comprehension
4. **What failure looks like downstream**
   - An annotation error that does not affect model performance may not matter
   - An annotation error that flips a model decision on a critical case matters a lot

Defining Failure Before You Start

A good QC plan specifies what failure would look like.

QC Dimension	What Failure Looks Like
Inter-annotator agreement	$\kappa < 0.60$ on any single category
Gold accuracy	Any annotator below 80% on gold items
Category distribution	Any category used $<$ 50% of expected rate
Annotation speed	Items completed in $<$ 5 seconds (likely not reading)
Systematic patterns	Same annotator disagrees with consensus $>$ 30% on a specific category

Pre-registering failure criteria prevents post-hoc rationalization.

Common QC Mistakes

Mistake 1: Report Only Overall κ

Per-category κ may reveal that one category drags down — or inflates — the overall number.

Mistake 2: QC Only at the End

Finding problems after 10,000 annotations means re-doing 10,000 annotations. Check early and often.

Mistake 3: Blame the Annotator First

Low agreement often signals bad guidelines, not bad annotators. Fix the system before fixing the people.

Mistake 4: Use QC to Gatekeep, Not to Learn

QC should be diagnostic — what do disagreements *teach* you about your task? — not just pass/fail.

Key Takeaways

1. **High agreement $\neq$ correct data.** Shared bias, easy-item inflation, and category collapse can all inflate metrics.
2. **Gold data has pitfalls.** Static gold, easy gold, and gold created by guideline authors all undermine QC.
3. **Annotator modeling treats disagreement as signal, not noise.** Even conceptually, this changes how you interpret your data.
4. **Manual inspection is irreplaceable.** Always look at the items behind the numbers.
5. **LLMs can assist QC but inherit training bias.** Use them to surface issues, not to override human judgment.
6. **Define failure before you start.** Pre-registered QC criteria prevent self-deception.

Questions?

Questions?

Office Hours: Wednesdays 1–3pm, Volen 109

jinzhao@brandeis.edu