

COSI-230B: Natural Language Annotation for Machine Learning

Lecture 18: LLMs: Roles, Risks, and Limits

Jin Zhao

Brandeis University

Spring 2026

Today's Agenda

Part I: LLMs as Annotators

- Three roles: annotator, assistant, adversary
- When LLM annotation is useful
- When it is dangerous
- Case studies across NLP tasks

Part II: Hybrid Pipelines

- Human-in-the-loop design
- Quality control strategies
- Cost–quality tradeoffs

Part III: The LLM Angle

- Prompt sensitivity
- Model collapse risks
- Feedback loops in self-training

Part IV: Ethics & Best Practices

- Systematic differences: human vs. LLM
- Ethical considerations
- Best practices summary

The Big Picture: Three Roles for LLMs



Guiding Principle

Each role carries distinct benefits **and** distinct risks. The key is knowing *when* and *how* to deploy each one.

When LLM Annotation Is Useful

High-Value Scenarios:

- ① **Bootstrap low-resource labels** — initial data for new languages/domains
- ② **Scale well-defined tasks** — sentiment, topic, language ID
- ③ **Silver-standard data** — pre-filter before human review
- ④ **Rapid prototyping** — test schemas before expensive campaigns

Task Characteristics Favoring LLMs:

- Clear, objective guidelines
- Binary or few-class decisions
- Large volume needed quickly
- Low ambiguity in label definitions

Task	H-H κ	LLM-H κ
Binary Sentiment	0.82	0.78
Toxicity Detection	0.75	0.71
Topic Classification	0.88	0.84

Example

GPT-4 achieves ~85–90% agreement with humans on binary sentiment (Gilardi et al., 2023). Cost: \$0.002 vs. \$0.50 per label.

When LLM Annotation Is Dangerous

High-Risk Scenarios

- ① **Subjective or culturally situated tasks** — hate speech, sarcasm, pragmatic inference
- ② **Expert-level annotation** — medical NER, legal reasoning, linguistic syntax trees
- ③ **Evaluation of LLMs themselves** — circular reasoning when the judge is the defendant
- ④ **Minority viewpoints** — LLMs reflect majority training distributions

The Circularity Problem

If you use GPT-4 to label data, then train a model on it, then evaluate with GPT-4 labels — you are measuring **agreement with GPT-4**, not task performance.

Case Study: Dangers in Practice

Hate Speech Detection

- LLMs systematically under-flag African American Vernacular English (AAVE)
- Simultaneously over-flag content mentioning marginalized groups
- Cultural context is *not* well captured by prompts alone

Pragmatic Inference

- “Nice job!” — sincere or sarcastic?
- LLMs default to literal interpretations
- Miss discourse-level cues

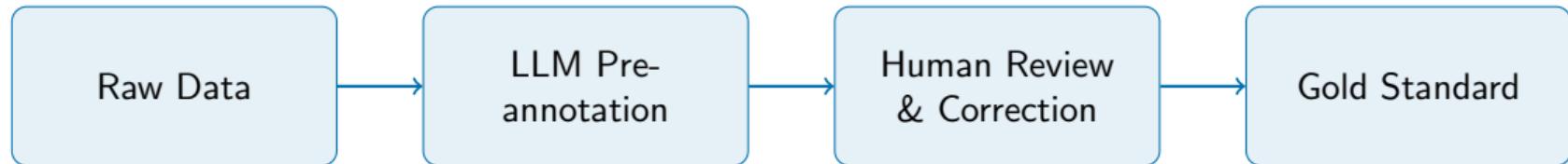
Clinical NLP

- Medication NER: LLMs confuse brand/generic names
- Negation scope: “no evidence of malignancy” ≠ positive finding
- Abbreviations are domain-specific (“pt” = patient vs. physical therapy)

Key Takeaway

High-stakes, context-dependent, or expert tasks demand **human annotation** with LLM assistance at most.

LLMs as Assistants in the Annotation Pipeline



Benefits of LLM-Assisted Annotation:

- **Speed:** Humans correct pre-labels 2–3x faster than annotating from scratch
- **Consistency:** LLM pre-labels reduce annotator drift over long sessions
- **Schema validation:** LLMs can flag ambiguous cases for expert review

Critical Design Choice

The human must be empowered to **override**, not merely **accept**. Anchoring bias is real — pre-labels can reduce disagreement by suppressing valid minority opinions.

LLMs as Adversaries: Stress-Testing NLP Systems

Adversarial roles for LLMs:

- ① **Generate adversarial examples** — paraphrases that flip model predictions
- ② **Create counterfactual data** — minimal edits that change the gold label
- ③ **Red-team other LLMs** — probe for failure modes, biases, hallucinations
- ④ **Augment evaluation suites** — expand coverage of edge cases

Adversarial Paraphrase

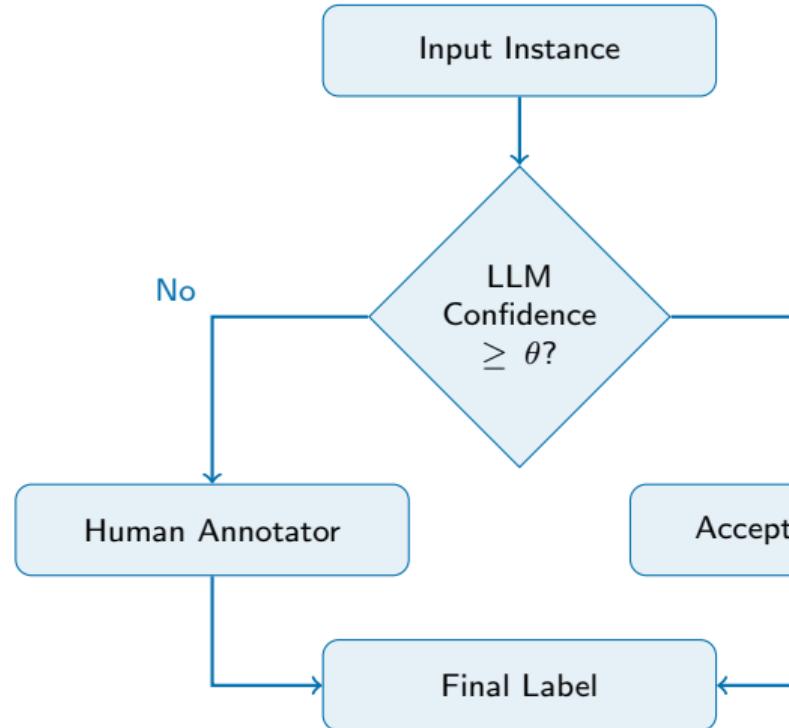
Original: “The movie was not bad.” → Positive
LLM Paraphrase: “The movie wasn’t terrible.”
→ Model says Negative?

Counterfactual NER

Original: “Apple released iOS 18.” → ORG
Counterfactual: “Apple fell from the tree.” → No entity

Hybrid Human-LLM Pipelines: Design Principles

- 1 Define the human's role clearly**
 - Creator, reviewer, adjudicator, or auditor?
- 2 Calibrate LLM confidence**
 - Route low-confidence items to humans
 - Use probability thresholds, not just argmax
- 3 Monitor distributional shift**
 - Compare LLM label distributions against known priors
- 4 Maintain human-only evaluation sets**
 - Never contaminate test data with LLM labels
- 5 Iterate the schema**
 - Use disagreements to refine guidelines



Cost–Quality Tradeoffs in Annotation

Strategy	Cost/Label	Quality	Speed
Expert human only	\$1.00–5.00	Highest	Slow
Crowd workers	\$0.10–0.50	Variable	Medium
LLM only (zero-shot)	\$0.001–0.01	Moderate	Fast
LLM + human review	\$0.20–1.00	High	Medium
LLM pre-label + expert correct	\$0.50–2.00	Very High	Medium

The Right Question

Not “Can the LLM do it?” but “**What is the cost of the LLM being wrong?**”

- Low-stakes prototype? LLM-only may be fine.
- Published benchmark? Human gold standard is essential.
- Clinical deployment? Expert review is non-negotiable.

Prompt Sensitivity: The Problem

Definition

Prompt sensitivity: Small changes in wording, formatting, or instruction framing can cause large shifts in LLM outputs — even when the task remains identical.

Examples of Prompt Variations for Sentiment:

Prompt	Positive %
"Classify the sentiment: positive or negative."	62%
"Is this review positive or negative?"	58%
"Rate the sentiment as POS or NEG."	71%
"Would you say this review is favorable?"	67%
"Label: 1 = positive, 0 = negative"	55%

Table: *

Same 500 reviews, different prompts, different label distributions

This is not a minor nuisance — it is a **methodological crisis** if your annotations depend on arbitrary prompt choices.

Prompt Sensitivity: Mitigation Strategies

Strategy 1: Prompt Ensembling

- Run k diverse prompts per instance
- Aggregate via majority vote or averaging
- Report variance across prompts

Strategy 2: Calibration

- Estimate label prior from a held-out set
- Apply contextual calibration (Zhao et al., 2021)
- Normalize output probabilities

Strategy 3: Few-Shot Anchoring

- Provide exemplars that span label space
- Reduces sensitivity to instruction wording
- But introduces exemplar selection bias

Strategy 4: Sensitivity Auditing

- Systematically vary prompts before deployment
- Flag instances with high cross-prompt disagreement
- Route unstable instances to humans

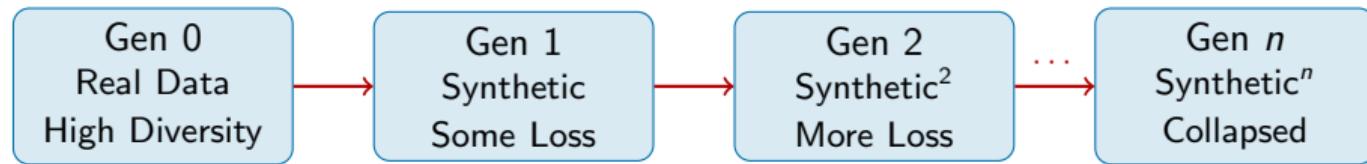
Best Practice

Always report which prompt(s) you used, and ideally show results are robust across paraphrases.

Model Collapse: What Is It?

Definition

Model collapse: When a model is trained on data generated by a previous version of itself (or a similar model), successive generations lose diversity and amplify biases, eventually “collapsing” to a narrow distribution.



Key finding (Shumailov et al., 2023): After 5–10 generations of self-training, models lose tail distribution coverage entirely. Rare phenomena disappear from the data.

Model Collapse: Implications for NLP

What Collapses First?

- Rare syntactic constructions
- Low-frequency named entities
- Dialectal and minority language features
- Nuanced or ambiguous labels
- Edge cases in reasoning

Why It Matters for Annotation:

- If you train on LLM labels, then use that model to label more data, you are on this path
- Each iteration *amplifies* the LLM's original biases

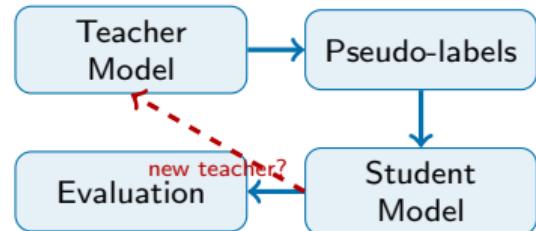
Mitigation:

- ➊ **Always mix real human data** into each training generation
- ➋ **Track distributional statistics** across generations (vocabulary, label distribution, entity frequency)
- ➌ **Preserve original data** — never discard real data in favor of synthetic
- ➍ **Use diverse models** — don't recycle a single model family

Rule of Thumb

Synthetic data should *supplement*, never *supplant*, real data.

Feedback Loops in Self-Training



The Danger:

- Errors become **training signal**
- Confident-but-wrong predictions are **self-reinforcing**
- Calibration degrades: overconfidence in wrong answers

Breaking the Cycle:

- ➊ **Human checkpoints:** inject human data regularly
- ➋ **Confidence filtering:** only use pseudo-labels above threshold
- ➌ **Negative sampling:** include examples the model gets wrong
- ➍ **Diverse teacher ensembles:** multiple independent models
- ➎ **Held-out monitoring:** track accuracy on fixed human dev set

Practical Rule

If dev accuracy **plateaus or drops** while training loss falls, you are in a feedback loop. Stop, audit, inject fresh human data.

Systematic Differences: Human vs. LLM Annotations

Dimension	Human Annotators	LLM Annotators
Label distribution	Reflects genuine disagreement	Often biased toward majority class
Ambiguity handling	Embrace uncertainty, split votes	Force a single answer
Cultural sensitivity	Can apply lived experience	Relies on training corpus
Consistency	Drift over time (fatigue)	High consistency, but consistently biased
Cost	Expensive, slow	Cheap, fast
Explainability	Can articulate reasoning	Generates plausible-sounding rationales

Key Insight

LLM annotations are **systematically different** from human annotations, not just noisier. This means LLM errors are **correlated**, which is worse than random noise for downstream models.

Discussion: Human vs. LLM Annotation

Key Questions

- ① Are LLM annotation errors **random** or **systematic**? What evidence supports your view?
- ② Which *types* of text or tasks are hardest for LLMs to annotate reliably?
- ③ Would you trust LLM labels in a published benchmark paper? Under what conditions?
- ④ How would you design a hybrid pipeline for a task you care about?

Common Findings from Empirical Comparisons:

- LLMs struggle with **implicit sentiment** ("The service was... interesting.")
- LLMs over-predict toxicity for texts mentioning **identity groups**
- LLMs are **more consistent** than humans but less **valid** on edge cases
- Human annotators bring **contextual knowledge** that prompts cannot fully encode

Ethical Considerations

Transparency

- Disclose when LLMs are used in annotation
- Report which model, prompt, and parameters
- Distinguish LLM labels from human labels in released datasets

Labor Implications

- LLM annotation displaces crowd workers
- Often the most vulnerable workers
- Ethical responsibility to consider impact

Bias Amplification

- LLMs encode societal biases from training data
- Using LLM labels propagates these biases into new datasets
- Downstream models inherit amplified biases

Accountability

- Who is responsible when LLM-labeled data leads to harm?
- No clear audit trail for automated labels
- Need institutional policies for LLM-in-the-loop research

Best Practices for Using LLMs in Annotation

- ① **Start with a pilot study:** Compare LLM labels to expert annotations on a sample *before* scaling
- ② **Use multiple prompts:** Report sensitivity analysis
- ③ **Maintain human-labeled test sets:** Never evaluate on LLM-generated gold data
- ④ **Document everything:** Model version, prompt text, temperature, decoding strategy
- ⑤ **Monitor for collapse:** Track label distributions across iterations of self-training
- ⑥ **Mix human and LLM data:** Never rely solely on synthetic labels for training
- ⑦ **Be transparent:** Disclose LLM involvement in publications and data releases
- ⑧ **Consider the task:** Objective tasks → LLM-friendly; subjective tasks → human-essential

Key Takeaways

- ① LLMs are powerful annotation tools **for the right tasks** — well-defined, objective, large-scale
- ② LLM annotation is **dangerous for subjective, expert, or culturally sensitive tasks**
- ③ **Hybrid pipelines** that combine LLM pre-labeling with human review are the current best practice
- ④ **Prompt sensitivity** means your results may depend on arbitrary wording choices — always audit
- ⑤ **Model collapse** is a real threat when training on LLM-generated data iteratively
- ⑥ **Feedback loops** in self-training amplify errors and biases — inject human data at every cycle
- ⑦ **Transparency** in reporting LLM use is an ethical imperative

Remember

Use LLMs. Don't worship them.

Questions?

Questions?

Office Hours: Wednesdays 1–3pm, Volen 109

 jinzhaob@brandeis.edu