COSI-230B: Natural Language Annotation for Machine Learning

Lectures 5 & 6: What Models Learn from Annotation

Jin Zhao

Brandeis University
Computational Linguistics Program

February 2 & 4, 2026

Today's Agenda

1. **Opening Question**
   - What does a supervised NLP model actually learn?
2. **Core Claim: Models Learn Annotator Behavior**
   - From labels to decision boundaries
3. **Mini-Demo: Same Text, Different Labels**
   - Labeling regimes and their consequences
4. **Annotation Choices as Inductive Bias**
   - Connecting annotation to ML theory
5. **Failure Modes Caused by Annotation**
6. **In-Class Activity: Diagnose the Failure**
7. **Reframing "Noise"**
8. **From Supervised Models to Foundation Models**
   - Same lesson, bigger surface area
9. **Closing Synthesis**

What Does a Supervised NLP Model Actually Learn?

What does a supervised NLP model actually learn?

Take a moment. What's your answer?

Common Answers

What people usually say:

- Meaning
- Semantics
- Patterns in language
- Linguistic structure

The Problem

These answers are not wrong, but they are incomplete in important ways.

The Actual Answer

Models learn decision boundaries induced by labels.

More precisely: **regularities in annotation decisions.**

Meaning, truth, and language only enter the model through labels.

This idea is the backbone of today's lecture.

What a Supervised Model Actually Learns

A supervised model learns:

- What annotators **consistently agree** on
- What annotators **consistently ignore**
- What annotators **systematically confuse**

Not:

- Speaker intent
- Social meaning
- World truth
- Moral correctness

Key Principle

If annotators can't see it, the model can't learn it.

Concrete Example: Sentiment Classification

What annotators do:

- Rely on **lexical cues** (“great,” “terrible,” “love”)
- Label sarcasm **inconsistently**
- Interpret cultural references **unevenly**

What the model learns:

- Strong lexical heuristics
- Weak pragmatic understanding
- Overconfidence on surface patterns

Reframing

This is less a model failure than **annotation fidelity**.

The model learned the patterns in the labels it was given.

One Text, Many Labels

"That's one way to do it."

What is this?

- Positive? (acknowledging creativity)
- Negative? (passive-aggressive dismissal)
- Sarcastic? (implying it's the wrong way)
- Neutral? (just an observation)

You will disagree. That's the point.

Three Datasets, Three Models

Thought experiment:

Dataset A: Majority vote = **neutral**

- Model learns: hedged observations → neutral

Dataset B: Sarcasm labeled **explicitly** as its own category

- Model learns: understatement ≠ neutral

Dataset C: Annotators split **across all four labels**

- Model learns: uncertain prediction near decision boundary

Which model "understands" this sentence?

Answer

None of them understand it.
They learn different **labeling regimes**.

Annotation = Inductive Bias Injection

Annotation Choice	What the Model Learns
Binary labels	Sharp decision boundaries
Forced choice	Tendency toward overconfidence
Majority vote	Most common perspective
No "uncertain" option	False certainty on edge cases
Single annotator	Individual perspective
Cleaned data	Reduced robustness

Key Insight

Annotation introduces inductive bias.

Annotation design decisions shape what the model can and cannot learn. This reframes annotation as a **first-class modeling decision**.

Why This Matters

Implications:

- Annotation design is **not** a preprocessing step
- It is **part of** the modeling pipeline
- Changing the labels changes the model—even with the same data and architecture

"Better labels" is underspecified without a modeling goal.

Better *for what*?

- Better for majority accuracy?
- Better for capturing ambiguity?
- Better for fairness across groups?
- Better for downstream robustness?

Each of these requires a **different** annotation scheme.

Failure Mode 1: False Confidence

Symptom:

- Model predicts with high confidence
- Humans disagree on the correct label

Cause:

- Forced-choice labels eliminate uncertainty
- Model never sees "I'm not sure" as an option
- Learns to always commit, even when the data is ambiguous

Result: Confident predictions on inherently uncertain inputs.

The model is doing exactly what we trained it to do.

Failure Mode 2: Systematic Blind Spots

Symptom:

- Model consistently misses certain phenomena
- Appears “biased” in deployment

Cause:

- Rare phenomena were never annotated
- Annotation guidelines didn’t cover edge cases
- Model never had the chance to learn them

Result: What was invisible in annotation is invisible to the model.

You can’t learn what you’ve never been shown.

Failure Mode 3: Distributional Collapse

Symptom:

- Model performs well on benchmarks
- Real-world performance is significantly worse

Cause:

- Annotation guidelines simplify reality
- Model overfits the simplification
- Real-world distribution is more complex than the labels suggest

Result: The model learned the annotation scheme, not the phenomenon.

Key Line

When models fail, they often fail **honestly**.
They are telling you what was in the labels.

Activity: Diagnose the Failure

For each scenario, we'll work through four questions together:

1. What **annotation assumption** likely caused this?
2. What information was **systematically erased**?
3. What **alternative annotation** could help?
4. What would it **cost**?

Keep in Mind

Consider trade-offs. Every fix has costs.
There are no free lunches in annotation design.

Scenario Examples

Scenario A:
A toxicity classifier flags African American Vernacular English (AAVE) as toxic at disproportionately high rates. It also flags reclaimed slurs used within communities.

Scenario B:
An event detection system misses implicit events ("The company went silent after the scandal") while catching explicit ones ("The company announced layoffs").

Scenario C:
A clinical NER system trained on discharge summaries fails on nursing notes, missing abbreviations, informal shorthand, and context-dependent references.

What annotation choices led to each failure?

Takeaways

Key takeaways:

- Every fix introduces **new problems**
  - More categories $\rightarrow$ more annotator disagreement
  - More nuanced labels $\rightarrow$ higher cost, lower throughput
- Annotation is about choosing **which errors you prefer**
  - False positives vs. false negatives
  - Precision vs. recall
  - Majority accuracy vs. minority coverage
- There is no annotation scheme that is **simultaneously**:
  - Cheap, fast, nuanced, fair, and robust

Disagreement $\neq$ Noise

Disagreement is not noise
unless you **define** it as noise.

Random noise $\neq$ systematic disagreement

Disagreement often encodes:

- **Ambiguity** — the text genuinely supports multiple readings
- **Multiple valid interpretations** — different but defensible analyses
- **Social variation** — different lived experiences, different judgments

The Cost of Throwing Away Disagreement

Common default in practice:

- Collect multiple annotations
- Take majority vote
- Discard disagreement

What you lose:

- Signal about ambiguous cases
- Information about difficulty
- Minority perspectives
- Calibration data for uncertainty

Key Sentence

Discarding disagreement often means discarding useful information.

This leads directly into future lectures on disagreement-aware modeling.

Transition: What About LLMs?

Everything we've said so far is true
for classifiers and taggers.

Now let's see what happens when the model
is generative and general-purpose.

Extension Thesis

LLMs don't change the annotation story.
They **expose** it.

What Has Not Changed

Same facts still hold:

1. Models learn from annotated signals
2. Signals encode human decisions
3. Decisions reflect constraints and norms
4. Models reproduce systematic regularities

Key Point

There is no magical break between "classic NLP" and LLMs.
The pipeline got longer, not different.

What Changed: Annotation Is Now Behavioral Policy

Old regime (classic NLP):

- Annotation defines task output
- Errors are localized
- Model behavior is narrow

LLM regime:

- Annotation defines **acceptable behavior**
- Outputs are open-ended
- Errors generalize across tasks

Key Line

When outputs are unconstrained, annotation becomes **policy**.

The Shift in Consequence

Classic NLP	**LLMs**
Label = class	Label = preference
Error = misclassification	Error = behavioral failure
Bias = skewed predictions	Bias = normative stance

This is a **scaling law**, not a category shift.

The same dynamics we identified earlier—just with wider reach and higher stakes.

Preference Signals as Labels-in-Disguise

In LLMs, we didn't remove labels. We made them implicit.

Examples of preference signals:

- Accept / reject
- Ranking (response A $>$ response B)
- Edits to model output
- "Better response" judgments

All of these are:

- **Compressed** judgments
- **One-dimensional** (better/worse)
- **Context-stripped** (why is lost)

Key Insight

Preference data collapses *why* into *which*.

Disagreement Revisited: From Noise to Value Pluralism

Earlier we said:

- Disagreement is signal
- Not all disagreement is error

Now extend:

Classic NLP disagreement:

- Linguistic ambiguity
- Task underspecification

LLM disagreement:

- Value conflict
- Normative trade-offs
- Competing notions of "helpful" or "safe"

Key Line

Disagreement didn't increase with LLMs.
The **cost of ignoring it** did.

Consequences of Suppressed Disagreement

What happens when we force consensus on value-laden judgments:

- Majority vote $\rightarrow$ **dominant norms** encoded as default
- Forced consensus $\rightarrow$ **brittle alignment**
- Suppressed disagreement $\rightarrow$ **strange refusal behavior**

These can be understood as **predictable consequences** of the same annotation dynamics we identified earlier—now applied to open-ended generation.

Thought Experiment: What Gets Rewarded?

Prompt:

"Give a safe but useful answer to this question."

Response A: Provides a direct, informative answer with a brief caveat.

Response B: Declines to answer and suggests consulting a professional.

Questions:

- Which would an annotator prefer?
- *Why?*
- What behavior would that preference train?

Takeaway

The model learns what gets **rewarded**—which may or may not align with what is **right**.
No new theory—just consequence at a larger scale.

Four Statements

1. Models learn what annotation makes legible.

2. Annotation determines what errors are possible.

3. No labeling scheme is fully neutral.

4. At scale, annotation decisions become social decisions.

The Practical Implication

If you don't like your model's behavior,
don't start with the optimizer.

Start with the annotation.

LLMs didn't so much introduce new problems
as make existing ones **harder to ignore**.

Questions?

jinzhao@brandeis.edu

Office Hours: Wed 1–3pm (Volen 109)

MOODLE for announcements