

# COSI-230B: Natural Language Annotation for Machine Learning

## Lecture 15: How Annotation Shapes & Breaks Models

Jin Zhao

Brandeis University

Spring 2026

# Today's Agenda

## Part I: How Annotation Shapes Models

- ① Annotation as inductive bias
- ② Error ceilings from annotation quality
- ③ Annotation artifacts and evaluation leakage

## Part II: How Annotation Breaks Models

- ④ Over-specification: too many distinctions
- ⑤ Noise amplification from labels
- ⑥ Annotation shift and distribution mismatch
- ⑦ Label collapse: losing distinctions that matter

### Core claim

Annotation decisions define what a model can learn — and what it gets wrong.

# Annotation as Inductive Bias

## What is inductive bias?

Assumptions a learner makes to generalize beyond training data.

### Standard examples:

- Architecture: CNNs assume spatial locality
- Regularization: L2 prefers smaller weights
- Feature engineering: Which inputs the model sees

### Annotation as inductive bias:

- Label set defines the hypothesis space
- Annotation guidelines encode linguistic theory
- Boundary decisions encode what distinctions matter

### Key Insight

The schema IS a theory of the task. The model will learn that theory, whether or not it is the right one.

# How Schemas Constrain Learning



**The model can never recover information the schema discards.**

## Example: Sentiment schemes

- Binary: collapses nuance
- 3-class: what counts as “neutral”?
- 5-point: more expressive, lower agreement

## Example: NER schemes

- CoNLL (4 types): MISC is a catch-all
- OntoNotes (18 types): finer distinctions
- F1 scores are *incomparable* across schemas

# Schema-Induced Error Ceilings

**Definition:** The maximum achievable performance given annotation ambiguities inherent in the schema.

## Sources of error ceilings:

- ① **Genuine ambiguity:** The text is truly ambiguous
- ② **Schema mismatch:** Categories don't carve reality at its joints
- ③ **Underspecified guidelines:** Edge cases not covered
- ④ **Annotator disagreement:** Reflects schema problems, not annotator failure

## Relationship to IAA:

Human agreement  $\kappa = 0.80 \Rightarrow$  Model ceiling  $\approx 0.80$

Human agreement  $\kappa = 0.95 \Rightarrow$  Model ceiling  $\approx 0.95$

A model that exceeds human agreement is likely **overfitting** to annotator biases.

# Measuring Error Ceilings

## How to detect schema-induced ceilings:

### ① Compute human upper bound:

- Multi-annotator agreement as performance ceiling
- Compare model to individual annotator performance

### ② Error analysis by category:

- Are certain labels systematically confused?
- Do errors cluster on schema boundaries?

### ③ Disagreement analysis:

- Items where annotators disagree = items where the model will struggle
- Confusion matrix of annotator pairs reveals schema weaknesses

### ④ Re-annotation experiment:

- Have annotators re-label their own data after a delay
- Intra-annotator disagreement reveals category instability

# Annotation Artifacts: Spurious Patterns Models Exploit

**Gururangan et al. (2018): Hypothesis-only baseline for NLI**

**Finding:** A model given *only the hypothesis* achieves 67% on SNLI (chance = 33%).

**Why?** Annotators introduced systematic patterns:

Label	Hypothesis Pattern
Entailment	Generic descriptions (“outdoors”, “animal”)
Contradiction	Negation words (“nobody”, “never”, “nothing”)
Neutral	Hedging language (“might”, “some”, “probably”)

**Cross-dataset evidence:** Performance drops 10–25 points when switching annotation schemes (e.g., CoNLL → OntoNotes: 93 → 78 F1).

Models learn the annotation scheme, not just the linguistic phenomenon.

# Detecting Annotation Artifacts

## Diagnostic tests:

### ① Partial-input baselines:

- Can the model succeed with incomplete input?
- If yes, artifacts exist in the labels

### ② Cross-dataset evaluation:

- Train on Dataset A, test on Dataset B (same task, different annotation)
- Large performance drops reveal schema dependence

### ③ Adversarial evaluation:

- Create challenge sets targeting known annotation artifacts
- Checklist-style probing (Ribeiro et al., 2020)

### ④ Annotator split:

- Ensure different annotators for train vs. test
- Measure performance gap

## Critical Problem

If your evaluation set shares annotation biases with your training set, your metrics are **inflated**.

# Discussion: Identifying Artifacts in Practice

Think about a dataset or task you have worked with:

- ① What annotation patterns might a model exploit as **shortcuts**?
  - Length cues, lexical cues, positional cues?
- ② If you gave the model only **partial input**, could it still predict labels?
  - Hypothesis-only in NLI, question-only in QA...
- ③ How would you **test** whether your model learned the task or the annotation scheme?
- ④ What would **change** if a completely different team re-annotated the same data?

## Key takeaway

The annotation scheme is a policy decision, not just a technical one.

# Over-specification: Too Many Distinctions

**Definition:** A schema draws distinctions finer than annotators can reliably apply or models can reliably learn.

Schema	Labels	Human IAA ( $\kappa$ )	Model F1
Binary (pos/neg)	2	0.85	0.91
Ternary (+/0/-)	3	0.73	0.82
5-point scale	5	0.54	0.58
7-point scale	7	0.38	0.41

**The paradox:** You designed more labels to capture more nuance, but the model ends up learning *less* because the signal is drowned in noise.

**Key question:** Can your annotators and your model reliably distinguish each category?

# Over-specification: Consequences and Examples

## Fine-grained NER (100+ types):

- “the Bay Area” — REGION? CITY?
- “Washington” — CITY? STATE? PERSON?
- Rare types have <10 examples
- Model confuses siblings constantly

**Solution:** Start coarse, refine only where it helps

## The over-specification cascade:

- ① Experts design fine distinctions
- ② Crowd annotators *cannot* reliably apply them
- ③ Models learn the noise from inconsistent labels
- ④ Simpler schemas often *outperform* complex ones
- ⑤ Community converges on coarser labels

### Pattern

Schemas designed by experts who can make fine distinctions often fail when applied at scale by non-experts.

# Noise Amplification: From Disagreement to Bad Models

**When annotators disagree, what does the model learn?**

**Scenario:**

- Sentence: “This movie was fine.”
- Annotator 1: Neutral
- Annotator 2: Positive
- Annotator 3: Negative
- Gold label (majority): Neutral

**But in training:**

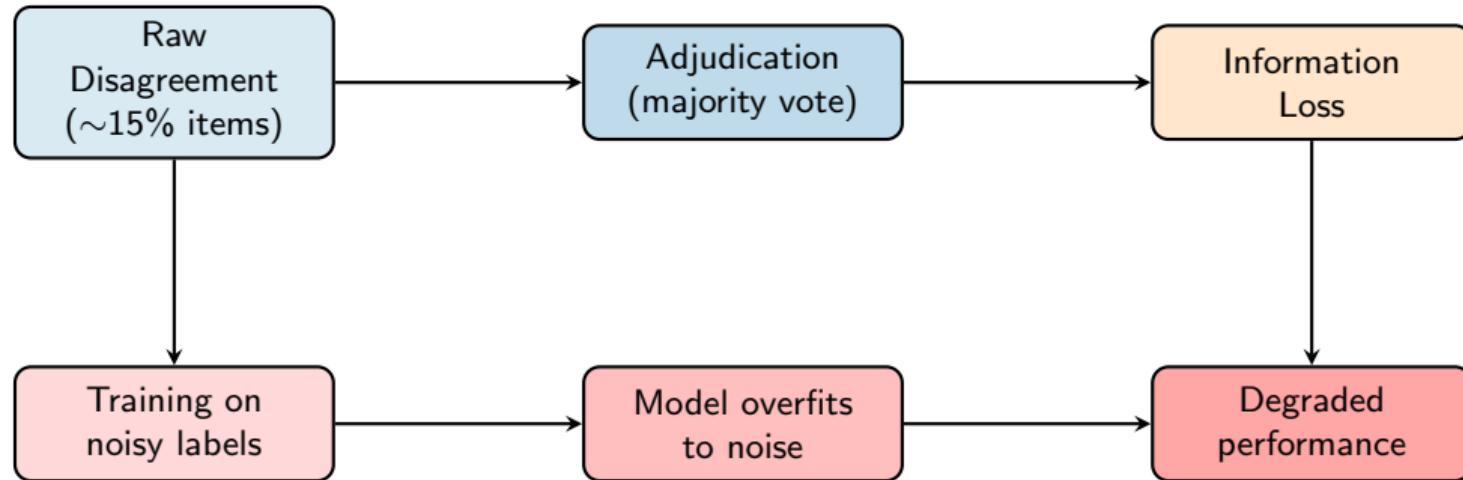
- Similar sentences get different labels
- Model receives contradictory gradients
- Learns to be uncertain everywhere

**The way to break the cycle:** Fix the schema, not add more data.

## Noise amplification cycle

- ① Ambiguous schema → disagreement
- ② Disagreement → noisy labels
- ③ Noisy labels → confused model
- ④ Confused model → poor predictions
- ⑤ Poor predictions → “need more data”
- ⑥ More data → more noise

# How Noise Gets Amplified



**Two paths, same outcome:**

- Majority vote *discards* valid minority perspectives
- Keeping all labels *sends contradictory signal* to the model

# Annotation Shift: Training $\neq$ Deployment

**Definition:** The distribution of labels in training data diverges from what the model encounters at deployment.

## Sources of annotation shift:

- ① **Temporal shift:** Language and norms evolve
  - "That's so lame": Not toxic (2019) → Ableist (2025)
- ② **Annotator population shift:** Different people, different judgments
  - In-house experts vs. crowdworkers vs. end users
- ③ **Domain shift:** Schema designed for one context, applied to another
  - Product reviews → social media
- ④ **Schema drift:** Guidelines change mid-project
  - New categories added, old ones redefined

## Unlike standard domain adaptation

This mismatch is in the **label space**, not just the input space. Perfectly representative inputs can have systematically wrong labels.

# Annotation Shift: Toxicity Over Time

What counts as “toxic” changes:

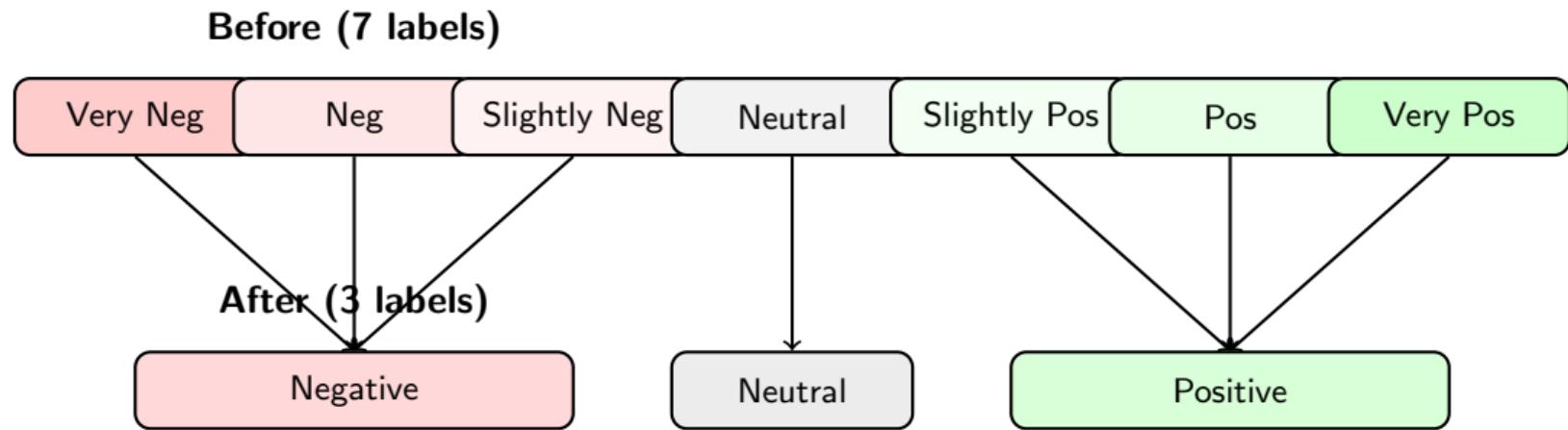
Example text	2019 label	2025 label
“That’s so lame”	Not toxic	Ableist
“Illegal aliens crossing. . . ”	Not toxic	Dehumanizing
“She’s pretty for an engineer”	Mildly toxic	Toxic
“I hate Mondays”	Not toxic	Not toxic

Impact:

- Models trained on 2019 labels under-flag content by 2025 standards
- Adding 2025 annotations to 2019 data creates *internal inconsistency*
- The “right” answer depends on *when* you ask

# Label Collapse: Losing Distinctions

**Definition:** Deliberately reducing the label set by merging categories that models cannot distinguish.



**Result:** IAA  $\kappa$ : 0.38 → 0.73   Model F1: 0.41 → 0.82

**When to collapse:** Low pairwise IAA ( $\kappa < 0.4$ ), persistent confusion matrix clusters, low support (<50 examples), downstream irrelevance

# Discussion: When Has Annotation Hurt Your Models?

**Think about your own experience:**

- ① Have you ever seen performance **improve** when you *removed* label categories?
- ② Have you encountered a dataset where annotator disagreement was a **feature**, not a bug?
  - What information was hiding in the disagreement?
- ③ Can you think of a task where fine-grained labels genuinely **helped**?
  - What made the fine distinctions learnable in that case?
- ④ How do you decide the right level of granularity **before** annotation?
  - Is there a principled method, or is it trial and error?

# Mitigation Strategies: Prevention

## Before annotation:

- ① Acknowledge your schema is a theory — make it explicit
- ② Pilot with 3+ annotators on 100+ examples
- ③ Compute pairwise IAA for *every label pair*
- ④ Merge any pair with  $\kappa < 0.4$
- ⑤ Use LLM diagnostics to test category distinguishability

## During annotation:

- ① Monitor per-label confusion rates continuously
- ② Flag items where majority vote margin  $< 60\%$
- ③ Consider *removing* hopelessly ambiguous items rather than adjudicating
- ④ Ensure different annotators for train vs. test splits

## Principle

A dollar spent on better guidelines is worth ten dollars spent on more labels with bad guidelines.

# Mitigation Strategies: Detection and Repair

## After annotation:

- ① Train with full and collapsed schemas — compare on downstream task
- ② Run partial-input baselines to detect artifacts
- ③ Evaluate on deployment-like data, not just held-out test
- ④ Report human upper bounds alongside model scores

## Task reframing options:

- Classification → Generation  
Explain *why* content is toxic
- Labels → Comparisons  
“Which is more positive, A or B?”
- Span labels → QA format  
“Who did what to whom?”
- Multi-class → Binary cascade  
Sequence of yes/no questions

**The same information can often be captured through a different annotation interface that produces cleaner data.**

# Key Takeaways

- ① **Annotation is inductive bias:** Labels define the hypothesis space a model can explore
- ② **Schemas encode theory:** Every label set embeds assumptions about language
- ③ **Error ceilings are real:** Annotator agreement bounds model performance — exceeding it is a red flag
- ④ **Artifacts contaminate evaluation:** Models exploit annotation patterns, not linguistic knowledge
- ⑤ **Over-specification hurts:** More labels  $\neq$  better models; noise drowns signal
- ⑥ **Noise amplification** is a vicious cycle — fix the schema, not the data volume
- ⑦ **Annotation shift** means old labels can actively mislead new models
- ⑧ **Label collapse and task reframing** are powerful, underused remedies
- ⑨ **Annotation scheme design** is a first-class modeling decision

# Questions?

## Questions?

Office Hours: Wednesdays 1–3pm, Volen 109

 [jinzhaob@brandeis.edu](mailto:jinzhaob@brandeis.edu)