COSI-230B: Natural Language Annotation for Machine Learning

Lecture 2: Annotation Fundamentals

Jin Zhao

Brandeis University
Computational Linguistics Program

Spring 2026

Today's Agenda

1. What is Annotation?
2. Types of Annotation Tasks
3. The MATTER Cycle
4. Quality vs. Quantity
5. Key Concepts
6. Looking Ahead

Defining Annotation

Definition

Annotation is the process of adding structured labels or metadata to raw data to make it usable for machine learning.

In NLP, annotation involves:

- Identifying linguistic phenomena in text
- Assigning labels according to a predefined schema
- Creating training signal for ML models

Key insight:
Annotation is *teaching machines through examples*.

Annotation = Explicit Knowledge

Before annotation:

- Raw text
- Implicit patterns
- Human intuition

"I love this movie!"

After annotation:

- Labeled data
- Explicit categories
- Machine-readable

"I love this movie!" $\rightarrow$ **POSITIVE**

Annotation transforms **implicit human knowledge** into **explicit training signal**.

Why Can't Machines "Just Learn"?

A common question:

"Why do we need annotation? Can't models just learn from data?"

How machines learn (core idea):

- Learning = optimizing an objective
- The objective defines what counts as a "good" output
- Different paradigms define this objective in different ways

Let's look at the three main learning paradigms...

Learning Paradigm 1: Supervised Learning

Objective: Predict human-provided labels

Example — Sentiment Classification:

- **Text:** "This movie was great"
- **Label:** positive

The model learns to map: inputs $\rightarrow$ labels

$\Rightarrow$ **Requires annotated data**

Learning Paradigm 2: Self-Supervised / Unsupervised

Objective: Predict or reconstruct parts of the data itself

Examples:

- **Language modeling:** Predict the next word
  "The cat sat on the ___" $\rightarrow$ *"mat"*
- **Word embeddings:** Predict surrounding words
- **Autoencoders:** Reconstruct the input

$\Rightarrow$ **No external labels during training**
The "label" is derived from the data itself

Learning Paradigm 3: Reinforcement Learning

Objective: Maximize a reward signal

Examples:

- **Game playing:** reward = winning the game
- **Robotics:** reward = staying upright, reaching target
- **RLHF (for LLMs):**
  - Humans rank outputs (which response is better?)
  - Rankings are used to learn a reward model

$\Rightarrow$ **The reward often comes from humans**

Why Annotation Still Matters

Even if training does not use labels directly...

Evaluation:

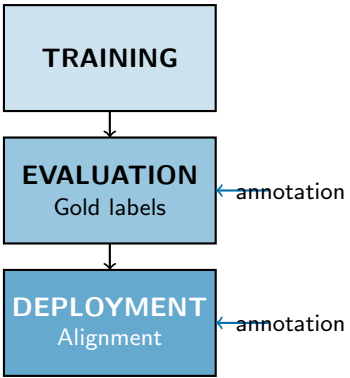
- We need gold annotations to measure accuracy, F1, etc.

Task-specific behavior:

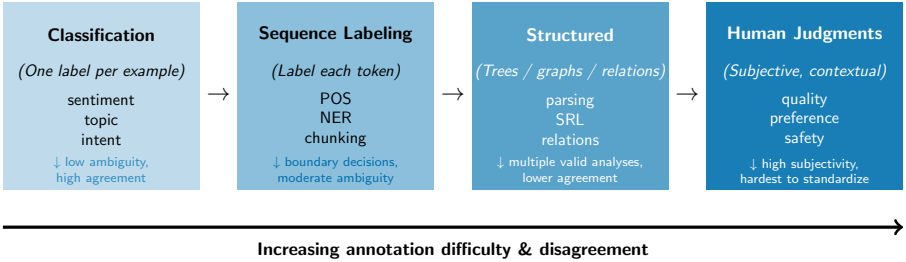
- Tasks like NER, parsing, event extraction require labeled data

Alignment:

- Human preferences define what "good" behavior means



The Landscape of NLP Annotation



As we move right, annotations shift from simple labels to structured representations to human judgments, which increases ambiguity, cost, and disagreement.

Classification Tasks

Goal: Assign one (or more) labels to a text unit.

Examples:

- Sentiment analysis
- Topic classification
- Intent detection
- Spam detection
- Language identification

Annotation format:

- Single label (multi-class)
- Multiple labels (multi-label)
- Ordinal scales (1-5 stars)

Example:

‘‘`The service was terrible but the food was amazing.`’’

→ Sentiment: **MIXED** — Aspect: **service:NEG, food:POS**

Sequence Labeling Tasks

Goal: Assign a label to each token (or span) in a sequence.

Common tasks:

- **Named Entity Recognition (NER):** Identify people, places, organizations
- **Part-of-Speech (POS) tagging:** Noun, verb, adjective, etc.
- **Chunking:** Noun phrases, verb phrases

Example (NER with BIO scheme):

Apple	announced	a	new	iPhone	in	Cupertino
B-ORG	O	O	O	B-PROD	O	B-LOC

Key challenge: Span boundaries—where does an entity start and end?

Structured Annotation Tasks

Goal: Capture relationships and hierarchical structure.

Common tasks:

- **Syntactic parsing:** Sentence structure (trees)
- **Semantic Role Labeling (SRL):** Who did what to whom
- **Relation extraction:** Links between entities
- **Coreference resolution:** What refers to what

Example (Relation Extraction):
"Tim Cook is the CEO of Apple."
→ `(Tim Cook, CEO_OF, Apple)`

Key challenge: Complex annotation schemas, longer training time.

Generation Evaluation Tasks

Goal: Evaluate quality of generated text.

Modern annotation tasks (especially for LLMs):

- **Quality ratings:** Is this response helpful? Accurate?
- **Preference judgments:** Which response is better? (A vs B)
- **Safety annotation:** Is this harmful? Biased?
- **Factuality:** Is this claim supported by evidence?

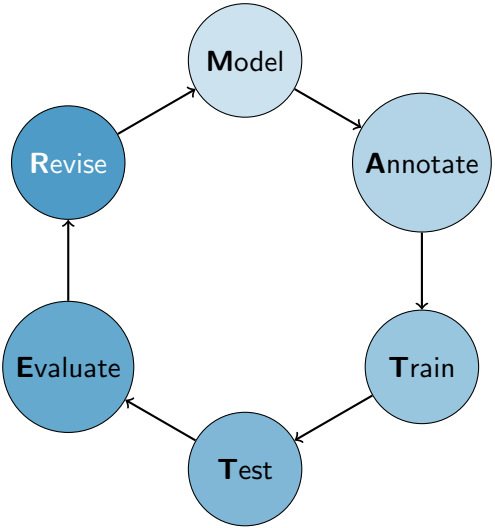
Example (Preference for RLHF): *Prompt: "Explain quantum computing"*

Response A:
Quantum computing uses quantum bits, or qubits, which can exist in multiple states at once through superposition. This allows quantum computers to process certain types of problems more efficiently than classical computers. However, quantum computers are still experimental and difficult to scale.

Response B:
Quantum computing is a very fast type of computing that uses special physics to solve problems much better than normal computers. It works by doing many calculations at the same time and will soon replace classical computers.

$\rightarrow$ **A ¿ B**

The MATTER Cycle



From Pustejovsky & Stubbs (2012), *Natural Language Annotation for Machine Learning*

MATTER: Step by Step

1. **Model:** Define what you want to annotate
   - What categories? What distinctions?
   - Create annotation schema and guidelines
2. **Annotate:** Apply labels to data
   - Multiple annotators for reliability
   - Measure inter-annotator agreement
3. **Train:** Build ML model on annotated data
4. **Test:** Apply model to held-out data
5. **Evaluate:** Measure performance (precision, recall, F1)
6. **Revise:** Improve based on errors
   - Refine guidelines? Add more data? Change schema?

Why "Cycle"?

Annotation is iterative, not one-shot.

First pass:

- Draft guidelines
- Pilot annotation
- Low agreement $\rightarrow$ guidelines unclear

Revision:

- Clarify edge cases
- Add examples
- Re-annotate problematic items

After training:

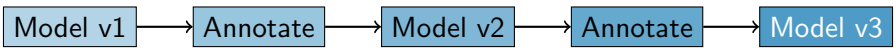
- Model errors reveal annotation gaps
- Error analysis $\rightarrow$ guideline refinement
- More annotation $\rightarrow$ better model

MAMA: A Lighter Alternative

MAMA: Model → Annotate → Model → Annotate...

When to use MAMA:

- Exploratory annotation
- Developing guidelines iteratively
- Active learning scenarios



Key idea: Each annotation round informs the next modeling decision.

The Fundamental Trade-off

Quality vs. **Quantity**

High Quality:

- Expert annotators
- Detailed guidelines
- Multiple passes
- Adjudication
- Expensive, slow

High Quantity:

- Crowdsourcing
- Simple tasks
- Single annotation
- No adjudication
- Noisy, inconsistent

The Question

Is it better to have 1,000 perfect labels or 10,000 noisy ones?

It Depends On...

Task complexity:

- Simple tasks (sentiment) $\rightarrow$ quantity often wins
- Complex tasks (SRL, coreference) $\rightarrow$ quality essential

Model architecture:

- Deep learning is somewhat noise-tolerant
- But systematic errors propagate

Use case:

- Research benchmark $\rightarrow$ need high quality
- Production prototype $\rightarrow$ may accept noise

Modern approach:

- LLM pre-annotation + human correction
- Best of both worlds?

The "Garbage In, Garbage Out" Principle

"Your model is only as good as your data."

Consequences of bad annotation:

- Model learns wrong patterns
- Evaluation metrics are misleading
- Errors compound in downstream tasks
- Research conclusions may be invalid

Key Insight

Time spent on annotation quality is **never wasted**.
Time spent training on bad data **often is**.

Key Terms to Know

Annotation schema: The set of labels/categories used

Annotation guidelines: Instructions for how to apply labels

Gold standard: The "correct" annotations (usually adjudicated)

Inter-annotator agreement (IAA): How much annotators agree

Adjudication: Resolving disagreements between annotators

Pilot annotation: Small-scale test before full annotation

Span: A contiguous sequence of tokens (for sequence labeling)

What Makes Annotation Hard?

Ambiguity:

- Language is inherently ambiguous
- Same text, different valid interpretations

Subjectivity:

- Opinions differ (sentiment, quality)
- Cultural and individual variation

Context dependence:

- Meaning depends on surrounding text
- World knowledge required

Edge cases:

- Guidelines can't cover everything
- Real data is messier than examples

Next Class: When to Annotate

Lecture 3 topics:

- When do you need annotation?
- Rule-based approaches vs. ML
- When can LLMs replace human annotation?
- Decision framework for annotation projects
- Overview of annotation tools

Reading:

- Pustejovsky & Stubbs, Chapters 1-2
- (Optional) Gilardi et al., "ChatGPT Outperforms Crowd-Workers"

Key Takeaways

1. **Annotation** transforms implicit human knowledge into explicit training signal
2. **Task types** range from simple classification to complex structured annotation
3. **MATTER cycle** emphasizes iteration: Model $\rightarrow$ Annotate $\rightarrow$ Train $\rightarrow$ Test $\rightarrow$ Evaluate $\rightarrow$ Revise
4. **Quality vs. quantity** is a fundamental trade-off—context determines the right balance
5. **Annotation is hard** because language is ambiguous, subjective, and context-dependent

Questions?

jinzhao@brandeis.edu

Office Hours: Wed 1–3pm (Volen 109)

MOODLE for announcements