COSI-230B: Natural Language Annotation for Machine Learning

Lecture 3: When to Annotate — Tools & Formats

Jin Zhao

Brandeis University
Computational Linguistics Program

Spring 2025

Today's Agenda

1. **Review & When to Annotate**
   - When rules suffice vs. when ML is necessary
   - Human vs. LLM annotation
   - Decision framework
2. **Annotation Tools**
   - Traditional: brat, MAE, WebAnno
   - Modern: Label Studio, Argilla, Prodigy
3. **Data Formats**
   - Standoff, inline, BIO, JSON
4. **Finding Data**
   - Sources, licensing, contamination concerns

Quick Review

From Last Lecture:

- Annotation transforms implicit knowledge into explicit training signal
- Task types: Classification $\rightarrow$ Sequence $\rightarrow$ Structured $\rightarrow$ Human Judgments
- The MATTER cycle: Model, Annotate, Train, Test, Evaluate, Revise
- Quality vs. quantity trade-offs

Today's Question:

When do we actually need annotation?

Do We Need to Annotate?

Consider these tasks:

Probably NO:

- Finding email addresses
- Removing HTML tags
- Sentence segmentation (English)
- Finding words with suffix "-ish"
- Tokenization (English)

Probably YES:

- Sentiment analysis
- Named entity recognition
- Event extraction
- Coreference resolution
- Quality evaluation

Key Question

Can the task be solved with **patterns/rules**, or does it require **learning from examples**?

When Rules Suffice

Rule-based approaches work when:

- Patterns are **explicit and consistent**
- Limited variation in how the phenomenon appears
- High precision is more important than recall
- Domain is narrow and well-defined

Examples:

- **Email extraction:** `[^@]+@[^@]+\.[^@]+`
- **Date formats:** `MM/DD/YYYY`, `YYYY-MM-DD`
- **HTML stripping:** Remove <tag>...</tag>
- **URL detection:** `https?://...`

If you can write a regex or a few rules that cover 95%+ of cases, you may not need ML.

When ML (and Annotation) is Necessary

Machine learning is needed when:

- Patterns are **implicit or complex**
- High variation in expression
- Context matters for interpretation
- Rules would be too numerous or brittle

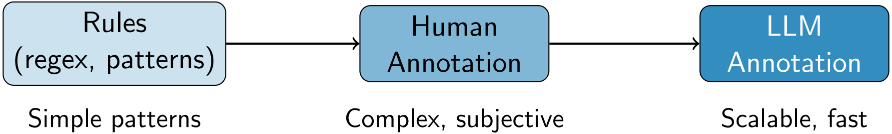
Examples:

- **Sentiment:** "Not bad" = positive? "Could be better" = negative?
- **NER:** "Apple" = company or fruit? Depends on context
- **Intent:** "Can you help me?" vs "Can you swim?"
- **Sarcasm:** "Oh great, another meeting!"

If meaning depends on **context**, **world knowledge**, or **subtle cues**, you need ML.

The New Question: Human vs. LLM Annotation

In 2025, we have a third option:



New questions:

- When can LLMs replace human annotators?
- When is human annotation still essential?
- When should we use hybrid approaches?

When LLMs Can Replace Human Annotation

LLMs work well for:

- Tasks with **clear, objective criteria**
- Tasks where LLMs were **trained on similar data**
- **High-resource languages** (English, Chinese, Spanish)
- Tasks where **speed and scale** matter more than perfect accuracy
- **Preliminary annotation** to bootstrap human review

Evidence:

- Gilardi et al. (2023): GPT-4 outperforms crowd-workers on some classification tasks
- Cost: \$0.001–0.01 per annotation vs. \$0.10–1.00 for humans
- Speed: Thousands of annotations per minute

When Human Annotation Remains Essential

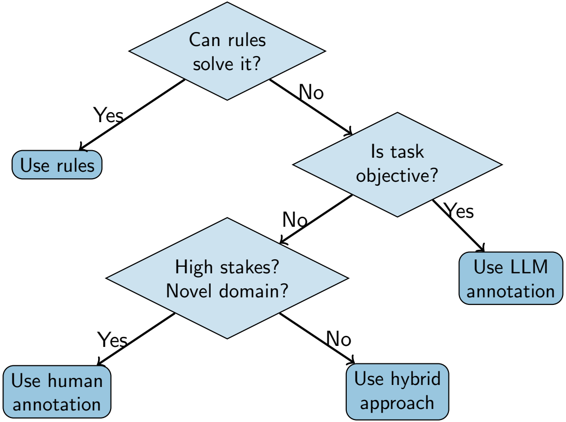
Humans are still needed for:

- **Subjective tasks** requiring cultural context or lived experience
- **Safety-critical applications** (medical, legal, high-stakes)
- **Novel domains** not in LLM training data
- **Low-resource languages** where LLMs perform poorly
- **Creating evaluation benchmarks** (can't evaluate LLMs with LLM labels!)
- **Edge cases** that require expert judgment

Warning

Using LLM annotations to evaluate LLM performance creates **circular evaluation**!

Decision Framework



The Annotation Tool Landscape

Traditional Tools:

- **brat** — Web-based, standoff format
- **MAE** — Multi-document, relations
- **WebAnno** — Collaborative, multi-layer
- **GATE** — Pipeline integration

Focus: Linguistic annotation, research

Modern Tools (with LLM support):

- **Label Studio** — Flexible, ML backends
- **Argilla** — RLHF, feedback loops
- **Prodigy** — Active learning, spaCy
- **Doccano** — Simple, open-source

Focus: ML pipelines, production

Choosing a Tool

Consider: task type, team size, LLM integration needs, export formats, cost

brat: The Classic

brat (brat rapid annotation tool)

- Web-based annotation interface
- Excellent for **sequence labeling** and **relation annotation**
- Uses **standoff format** (annotations separate from text)
- Good for linguistic research
- Free and open-source

Best for:

- NER, event extraction, relation extraction
- Small to medium annotation projects
- When you need fine-grained control over annotation schema

Website: `https://brat.nlplab.org/`

Label Studio: The Modern Choice

Label Studio

- Highly flexible — supports text, image, audio, video
- **ML backends** for pre-annotation and active learning
- **LLM integration** for auto-labeling
- Team collaboration features
- Multiple export formats (JSON, CSV, COCO, etc.)
- Free open-source version + enterprise option

Best for:

- Production ML pipelines
- Multi-modal annotation
- Teams needing LLM-assisted workflows

Website: `https://labelstud.io/`

Argilla: For RLHF and Feedback

Argilla

- Designed for **LLM feedback collection**
- Native support for **preference annotation** (RLHF)
- Integration with Hugging Face ecosystem
- Built-in **weak supervision** and **active learning**
- Collaborative workflows

Best for:

- RLHF data collection
- LLM evaluation and red-teaming
- Human-AI collaborative annotation

Website: `https://argilla.io/`

Tool Comparison

Feature	brat	Label Studio	Argilla	Prodigy
Cost	Free	Free/Paid	Free/Paid	Paid
LLM Integration	No	Yes	Yes	Yes
RLHF Support	No	Limited	Yes	Limited
Multi-modal	No	Yes	Limited	Limited
Active Learning	No	Yes	Yes	Yes
Self-hosted	Yes	Yes	Yes	Yes
Learning Curve	Medium	Low	Medium	Low

Why Data Formats Matter

Annotation data must be:

- **Machine-readable** — for training models
- **Human-readable** — for debugging and review
- **Interoperable** — works with different tools
- **Preserving** — maintains original text integrity

Two main approaches:

1. **Standoff annotation** — annotations stored separately from text
2. **Inline annotation** — annotations embedded in text

Standoff Annotation

Annotations stored separately, referenced by offsets

Raw text:

```
The Massachusetts State House is located in Boston.
```

Annotation file:

```
T1    Location 4 30    Massachusetts State House
T2    Location 45 51    Boston
R1    Located_in Arg1:T1 Arg2:T2
```

Advantages:

- Original text unchanged
- Easy to add/remove annotation layers
- Supports overlapping annotations

Tools: brat, MAE, WebAnno

Inline Annotation

Annotations embedded directly in the text

XML format:

```
The <location>Massachusetts State House</location>
is located in <location>Boston</location>.
```

Advantages:

- Easy to read and understand
- Self-contained (one file)

Disadvantages:

- Modifies original text
- Difficult with overlapping annotations
- Can break text processing pipelines

BIO/IOB Tagging Scheme

For sequence labeling tasks (NER, chunking)

- **B** = Beginning of entity
- **I** = Inside entity (continuation)
- **O** = Outside any entity

Example:

The	Massachusetts	State	House	is	located	in	Boston	.
O	B-LOC	I-LOC	I-LOC	O	O	O	B-LOC	O

Variants:

- **IOB1:** B only used when two entities are adjacent
- **IOB2 (BIO):** B always starts an entity
- **BIOES:** Adds E (End) and S (Single token entity)

JSON Format

Modern, flexible, widely supported

```
{
  "text": "Apple announced a new iPhone.",
  "entities": [
    {"start": 0, "end": 5, "label": "ORG"},
    {"start": 22, "end": 28, "label": "PRODUCT"}
  ],
  "sentiment": "neutral"
}
```

Advantages:

- Human-readable and machine-readable
- Flexible schema
- Native to most programming languages
- Used by Label Studio, Hugging Face, most APIs

Format Comparison

	Standoff	**XML**	**BIO**	**JSON**
Preserves text	Yes	No	Yes	Yes
Overlapping spans	Yes	Difficult	No	Yes
Human readable	Medium	High	Medium	High
ML-ready	Medium	Low	High	High
Relations	Yes	Yes	Limited	Yes
Common use	Research	Legacy	Seq. labeling	Production

Recommendation:

- Use **JSON** for most modern workflows
- Use **BIO** for sequence labeling models
- Understand **standoff** for research tools (brat)

Where to Find Data

Existing datasets:

- **Hugging Face Datasets:** `https://huggingface.co/datasets`
- **Kaggle:** `https://www.kaggle.com/datasets`
- **LDC:** `https://www.ldc.upenn.edu/`
- **ACL Anthology:** `https://aclanthology.org/`
- **Papers With Code:** `https://paperswithcode.com/datasets`

Raw data sources:

- **Common Crawl:** Web scrape archive
- **Wikipedia / Wikimedia**
- **Project Gutenberg:** Public domain books
- **Social media APIs:** Reddit, etc.

Create your own:

- Wizard of Oz data collection
- **Synthetic data generation with LLMs**

Copyright and Licensing

Why it matters:

- Annotation is a lot of work — you want to share it
- Legal requirements for redistribution
- Reproducibility of research

Common licenses:

CC-BY	Attribution required, commercial use OK
CC-BY-SA	Attribution + ShareAlike (derivatives same license)
CC-BY-NC	Attribution + Non-commercial only
MIT/Apache	Permissive software licenses

Workaround for restricted data:

- Release **annotations as offsets** only
- Provide script to download and reconstruct
- Problem: What if original data gets deleted?

Data Contamination (New Concern)

The problem:

- LLMs are trained on massive web data
- Your "test set" might be in the LLM's training data
- Leads to **inflated performance** and **invalid evaluation**

Decontamination strategies:

- Use **recent data** (after LLM training cutoff)
- Create **novel synthetic examples**
- Check for **n-gram overlap** with known training data
- Use **held-out test sets** that were never public

Important for Projects

When evaluating LLMs, ensure your test data wasn't in their training!

Key Takeaways

1. **Not everything needs annotation** — rules work for simple, explicit patterns
2. **ML needs annotation** when patterns are implicit, context-dependent, or subjective
3. **LLMs can help** but aren't always the answer — consider task type, stakes, and evaluation needs
4. **Tool choice matters** — brat for research, Label Studio for production, Argilla for RLHF
5. **Data formats** — know standoff, BIO, and JSON for different use cases

Questions?

jinzhao@brandeis.edu

Office Hours: Wed 1–3pm (Volen 109)

MOODLE for announcements