COSI-230B: Natural Language Annotation for Machine Learning

Lecture 19: RLHF & Preference Annotation

Jin Zhao

Brandeis University

Spring 2026

Today's Agenda

1. From traditional annotation to LLM alignment
2. The RLHF pipeline
3. Preference annotation task design
4. Reward modeling and annotation data
5. Constitutional AI: principle-based feedback
6. Direct Preference Optimization (DPO)
7. Practical considerations and quality challenges
8. Discussion and open problems

From Traditional Annotation to LLM Alignment

Traditional annotation:

- Train models to perform specific tasks (classification, NER, parsing)
- Ground truth labels for supervised learning
- Question: "Is this correct?"

LLM-era annotation:

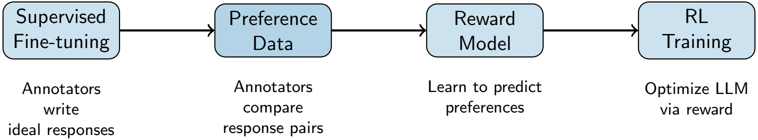
- Align models with human values and preferences
- Evaluate open-ended generation quality
- Ensure safety, helpfulness, and honesty
- Human feedback as training signal

Key shift: From "is this correct?" to "is this better?"

What is RLHF?

Reinforcement Learning from Human Feedback

Goal: Align LLM behavior with human preferences



How ChatGPT, Claude, etc. are trained — annotation is the foundation

Why Preferences Over Demonstrations?

Comparing is easier than generating

Demonstration annotation:

- "Write the ideal response"
- Requires expertise
- Time-consuming
- Single point of reference

Preference annotation:

- "Which response is better?"
- Easier cognitive task
- Faster annotation
- Captures relative quality

Key insight: The InstructGPT paper used ∼13K demonstrations but ∼33K preference comparisons — preferences scale better

Annotation is essential for both SFT (step 1) and reward modeling (step 2)

Preference Annotation Task Design

Core task: Given a prompt and two responses, which is better?

Example

Prompt: "Explain quantum computing to a 10-year-old."
Response A: "Quantum computing uses quantum bits that can be 0, 1, or both at the same time..."
Response B: "Imagine you have a magical coin that can be heads and tails at the same time..."

Annotation: $A > B$ or $B > A$ or $A \approx B$

Interface design: Show prompt at top, responses side by side, randomize A/B ordering

Preference Data Formats

Common comparison formats:

1. **Binary comparison:**
   - A wins, B wins, or tie
   - Simple, clear — most common in RLHF
2. **Graded comparison:**
   - A much better, A slightly better, tie, B slightly better, B much better
   - More granular signal
3. **Rating scale:**
   - Rate each response 1–5 independently
   - More information but harder to calibrate across annotators
4. **Best-of-N ranking:**
   - Order multiple responses from best to worst
   - Rich signal, more complex annotation task

Quality Challenges in Preference Annotation

Subjectivity issues:

- Different annotators have different preferences
- Cultural and individual variation
- Task interpretation differences

Cognitive biases:

- **Position bias:** Prefer first or second response
- **Length bias:** Prefer longer (or shorter) responses
- **Verbosity bias:** More words = better?
- **Fatigue:** Annotation quality degrades over time
- **Inconsistency:** Same annotator gives different judgments on similar pairs

Hard cases: Both responses good (small differences), both responses bad (which is less bad?), conflicting criteria

Reward Model Training

Converting human preferences into a reward function

Process:

1. Collect preference pairs: (x, y_w, y_l) — prompt, winner, loser
2. Train reward model: $r(x, y_w) > r(x, y_l)$
3. Use Bradley-Terry model for pairwise comparisons

Loss function:

$$L = -\log \sigma\big(r(x, y_w) - r(x, y_l)\big)$$

Annotation requirements:

- High-quality, consistent preference labels
- Diverse prompt distribution
- Sufficient scale: 50K–100K+ pairs typical

Annotation Guidelines for Reward Modeling

Common evaluation criteria (InstructGPT):

Primary criteria:

1. **Helpful:** Provides useful, relevant information
2. **Harmless:** Avoids dangerous or unethical content
3. **Honest:** Accurate, avoids hallucination

Secondary criteria:

- Follows instructions
- Appropriate length
- Clear and well-organized
- Appropriate tone
- Addresses all parts of prompt

Critical challenge: Criteria can conflict!

A response can be helpful but contain inaccuracies, or safe but unhelpful.

Guidelines must specify how to weigh trade-offs.

Constitutional AI: Principle-Based Feedback

Anthropic's alternative to pure human preference annotation

Process:

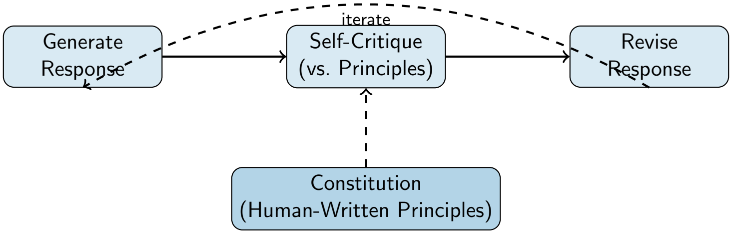
1. Define a constitution — principles like "be helpful," "be harmless," "be honest"
2. AI critiques its own outputs against these principles
3. AI revises its responses based on self-critique
4. Human feedback validates *principles*, not individual instances

Annotation role:

- Validate that principles are correct and complete
- Check that self-critique is reasonable
- Identify edge cases where principles conflict

Benefit: More scalable — annotate principles once, apply to many instances

Constitutional AI: Self-Improvement Loop



Comparison with RLHF:

- RLHF: Human annotates *each response pair* — expensive, doesn't scale
- CAI: Human defines *principles* — AI applies them to all instances
- Trade-off: Less fine-grained feedback, but much more scalable

Direct Preference Optimization (DPO)

Skip the reward model entirely

Traditional RLHF:
Preferences → Reward Model → RL Training (PPO)

DPO:
Preferences → Direct LLM Training

Benefits:

- Simpler pipeline — no separate reward model
- More stable training — no RL instabilities
- Mathematically equivalent objective under certain assumptions
- **Same preference data requirements**

Key insight: The preference data is the bottleneck, not the training algorithm

DPO Data Requirements vs. RLHF

What DPO needs:

- Prompts (from target distribution)
- Chosen responses (preferred by annotators)
- Rejected responses (dis-preferred by annotators)

Data format (Hugging Face standard):

- `prompt`: The input query
- `chosen`: Preferred response
- `rejected`: Dis-preferred response

Scale:

- 10K–100K pairs for good results
- Quality matters more than quantity
- Diverse prompts are critical

Same annotation process as RLHF — the difference is only in training

Practical Considerations: Scale and Cost

Real-world numbers:

- InstructGPT: $\sim$33K preference comparisons from 40 annotators
- Llama 2: $\sim$1M+ preference annotations
- Commercial annotation: $1–5 per comparison (depending on complexity)

Cost drivers:

- Response length (longer responses take more time to evaluate)
- Domain expertise required (medical, legal, coding)
- Number of annotators per pair (typically 3+)
- Quality control overhead

Platforms:

- **Argilla:** Open-source, built for RLHF, Hugging Face integration
- **Label Studio:** Customizable comparison templates
- **Scale AI / Surge AI:** Commercial, managed workforce

Annotator Training and Quality Control

Best practices for preference annotation:

1. **Clear criteria:** Define what "better" means with examples
2. **Calibration sessions:** Regular alignment on edge cases
3. **Randomize order:** Counteract position bias
4. **Multiple annotators:** 3+ per pair recommended
5. **Quality monitoring:** Track agreement over time
6. **Fair compensation:** Preference annotation is cognitively demanding

Measuring agreement:

- Clear quality differences: $\kappa > 0.6$
- Similar quality responses: $\kappa \approx 0.3$–0.5
- Highly subjective: $\kappa < 0.3$

Note: Low agreement may reflect genuine subjectivity — not always a problem

Discussion: Designing Preference Annotation for Your Task

Consider a domain you care about (e.g., medical QA, code generation, tutoring)

Design questions:

1. What criteria would you use to define "better"?
2. How would you weigh conflicting criteria (helpful vs. safe)?
3. What format: binary comparison, graded, or ranking?
4. Who are your annotators? Domain experts or general crowd?
5. How would you handle cases where both responses are good (or both bad)?

Activity: In pairs, sketch a 1-page annotation guideline for preference annotation in your chosen domain.

5 minutes — then share with the class

Current Challenges in Preference Annotation

Scalability:

- Human annotation is slow and expensive
- LLM-as-judge: Can AI replace human annotators?
- Debate: Is synthetic preference data sufficient?

Representation:

- Whose preferences are captured?
- Annotator demographics shape model behavior
- WEIRD (Western, Educated, Industrialized, Rich, Democratic) bias

Annotator wellbeing:

- Safety annotation exposes workers to harmful content
- Psychological impact and burnout
- Need for support, rotation, and fair pay

Open Problems

Research frontiers:

1. **Pluralistic alignment:** How to represent diverse preferences, not just majority?
2. **Reward hacking:** Models learn to exploit reward model weaknesses
3. **Distributional shift:** Preferences collected on Model v1 may not transfer to Model v2
4. **Multi-dimensional preferences:** Moving beyond single "better/worse" to structured feedback
5. **Automated evaluation:** Can we reduce human annotation needs without sacrificing quality?

The annotation challenge persists: Better training algorithms don't solve the data quality problem

Key Takeaways

1. **RLHF** uses human preferences to align LLMs — annotation is the foundation
2. **Preferences are easier** to annotate than demonstrations, and they scale better
3. **Reward models** convert preference annotations into a training signal
4. **Constitutional AI** replaces instance-level annotation with principle-based feedback
5. **DPO** simplifies the pipeline but needs the same preference data
6. **Quality of preference data** is the bottleneck — not the algorithm
7. **Annotation design** (criteria, formats, bias mitigation) directly shapes model behavior

Questions?

Questions?

Office Hours: Wednesdays 1–3pm, Volen 109

jinzhao@brandeis.edu