Annotation-Informed Modeling II

Evaluation and Error Analysis

Jin Zhao

Brandeis University

April 15, 2025

Today's Agenda

1. Evaluation metrics review
2. Metrics for different task types
3. Error analysis using annotations
4. LLM-as-judge evaluation
5. Building evaluation benchmarks
6. Avoiding common pitfalls

Evaluation Metrics: Basics

For classification:

Precision:

$$P = \frac{TP}{TP + FP}$$

How many predictions are correct?

Recall:

$$R = \frac{TP}{TP + FN}$$

How many actual positives found?

F1 Score:

$$F_1 = \frac{2 \cdot P \cdot R}{P + R}$$

Harmonic mean of precision and recall

Multi-Class Metrics

How to aggregate across classes:

Macro-averaged:

- Calculate metric for each class
- Average across classes (equal weight)
- Good for imbalanced data

Micro-averaged:

- Pool all predictions
- Calculate single metric
- Dominated by frequent classes

Weighted:

- Weight by class frequency
- Balance between macro and micro

Metrics for Sequence Labeling

Two levels of evaluation:

Token-level:

- Each token is a prediction
- Standard P/R/F1 on BIO labels
- Easier to achieve high scores

Entity-level:

- Each entity span is a prediction
- Match requires correct type AND boundaries
- More meaningful for task performance

Recommendation: Report both, emphasize entity-level

Entity-Level Evaluation

Exact match:

- Entity must match exactly (start, end, type)
- Strict but clear

Partial match options:

- **Type match:** Correct type, overlapping span
- **Partial:** Some overlap between predicted and gold
- **Boundary:** Allow 1-token boundary variation

Standard practice:

- Report exact match (for comparison)
- Can also report partial (for analysis)

Error Analysis

Beyond aggregate metrics:

Confusion matrix:

- Which classes are confused?
- Systematic patterns in errors?

Error categorization:

- Boundary errors (span too long/short)
- Type errors (wrong category)
- Missing errors (entity not found)
- Spurious errors (non-entity labeled)

Connect to annotations:

- Do errors correlate with low IAA?
- Which guidelines need improvement?

LLM-as-Judge

Using LLMs to evaluate model outputs

Setup:

1. Model generates output
2. LLM evaluates quality (1-5 scale, pass/fail, etc.)
3. Aggregate LLM scores

Advantages:

- Scalable to large test sets
- Consistent (unlike human judges)
- Can provide explanations

Disadvantages:

- Bias toward LLM preferences
- May not match human judgment
- Circular if evaluating LLMs

LLM-as-Judge Cautions

When NOT to use:

- Evaluating the same LLM that's judging
- Tasks where LLMs are known to fail
- High-stakes decisions
- Creating research benchmarks

Best practices:

- Validate against human judgments
- Report correlation with humans
- Use as supplement, not replacement
- Be transparent about methodology

Building Evaluation Benchmarks

Creating test sets for fair comparison

Requirements:

1. High-quality human annotations
2. Representative of real task
3. Held out from all training
4. Well-documented

Decontamination:

- Ensure test data not in LLM training
- Use recent data after training cutoff
- Check for n-gram overlap
- Consider paraphrase contamination

Benchmark Design Principles

Good benchmarks:

1. **Clear task definition:** Unambiguous what success means
2. **Representative data:** Covers realistic scenarios
3. **Sufficient size:** Enough for statistical significance
4. **High quality labels:** Human-annotated, adjudicated
5. **Versioned and documented:** Track changes over time
6. **Accessible:** Available for research use

Anti-patterns:

- LLM-generated test labels
- Too small sample size
- Undocumented preprocessing

Reporting Results

What to include:

1. **Metrics:** P, R, F1 (macro and micro)
2. **Confidence intervals:** Bootstrap or cross-validation
3. **Baselines:** For comparison
4. **Per-class breakdown:** Identify weak points
5. **Error analysis:** Common failure modes
6. **Statistical significance:** If comparing systems

Example: "Our model achieves 78.3 F1 ($\pm$1.2) on entity-level evaluation, compared to 72.1 F1 for the baseline. Performance on PERSON (85.2) exceeds ORG (71.4)."

Common Evaluation Pitfalls

Avoid these mistakes:

1. **Test set contamination:** Training on test data
2. **Overfitting to dev:** Too much tuning on dev set
3. **Cherry-picking metrics:** Only reporting best metric
4. **Missing baselines:** No comparison point
5. **Ignoring variance:** Single run without confidence
6. **Unfair comparisons:** Different preprocessing/data

Next Class: Preference Data & RLHF

Lecture 23 (Apr 20): Preference Data & RLHF Annotation

Topics:

- Introduction to RLHF
- Preference annotation design
- Reward modeling data collection
- Constitutional AI and principle-based feedback
- DPO data requirements

Assignment: HW 4 due next week

Key Takeaways

1. **Precision/Recall/F1** are standard metrics – know when to use each
2. **Entity-level** evaluation is more meaningful for NER
3. **Error analysis** reveals systematic model weaknesses
4. **LLM-as-judge** is useful but has limitations
5. **Good benchmarks** require human annotation and decontamination
6. **Report results** with confidence intervals and baselines

Questions?

Questions?

Office Hours: Wednesdays 1-3pm, Volen 109

jinzhao@brandeis.edu