

EVALUATION I: METRICS AND BENCHMARK DESIGN

INF385T: Post-Training of LLMs

CHAN YOUNG PARK

Assistant Professor, iSchool, School of Computing, The University of Texas at Austin

Recap: what did Week 2 establish?

- **Prompting matters** (how you design your prompt can really change the performance) + **which base model you use has a significant impact** on the actual final performance
- A strong model + prompt baseline should come before changing weights
- **No one metric decides the final model decision.** You should check diverse metrics to evaluate diverse aspects (accuracy, valid JSON, failures, and cost described different behaviors)
- **Question: what claim did a 95% accuracy actually support?**

Evaluation I: Metrics and Benchmark Design

- From claim to measurement
- Benchmark == sampling design
- Metrics == imperfect tools
- Lab 3: Preparing evaluation for your group project
- Core questions
 - How do we turn a desired behavior into decision-relevant evidence?
 - What can a benchmark score (not) tell us?

Which model is better?

Task: Summarize the announcement in one sentence.

The group project proposal is due on September 23 and should be approximately 1–2 pages. It should explain the task and motivation, why post-training may be needed, the baseline and evaluation plan, the training data and initial method, and major feasibility or safety risks. It should also state each team member's initial responsibility.

A: The proposal is due **September 23** and should briefly explain the project goal, why adaptation may be necessary, how the team will establish and evaluate a baseline, what data and training approach it plans to use, what major risks it **anticipates**. It should also include what each member will handle.

B: The **1–2 page** project proposal, due **September 16**, should motivate the task and need for post-training, describe the baseline, evaluation, data, and initial method, identify major risks, and assign initial team responsibilities.

Different metrics different winners

Evaluation metric	A	B	Likely winner
Exact match to reference	X	X	Tie
Factual consistency	Correct	Wrong date	A
Format compliance	Wrong	Correct	B
Compression rate	Low	High	B

Metrics disagree because constructs differ

Evaluation as measurement

- A score is not model *quality* itself
- A score is *evidence* for a claim
- Every metric makes assumptions
- Evaluation requires a measurement argument

An evaluation score is evidence for a claim, not the claim itself.

Three distinctions to remember

- Task performance $\neq$ construct
- Benchmark $\neq$ dataset
- Benchmark score $\neq$ product quality

From claim to decision

what will this
evaluation
be used for?

what do we want
to conclude?

what capability /
behavior do we mean?

what counts as
one observation?

what population are
we sampling from?

USE

CLAIM

CONSTRUCT

UNIT

CASES

Evaluation design

what counts as
good behavior?

how does behavior
become a score?

how do observations
become evidence?

where does the
conclusion hold / fail?

what changes
next?

**REFERENCE
+ RUBRIC**

**METRIC/
SCORER**

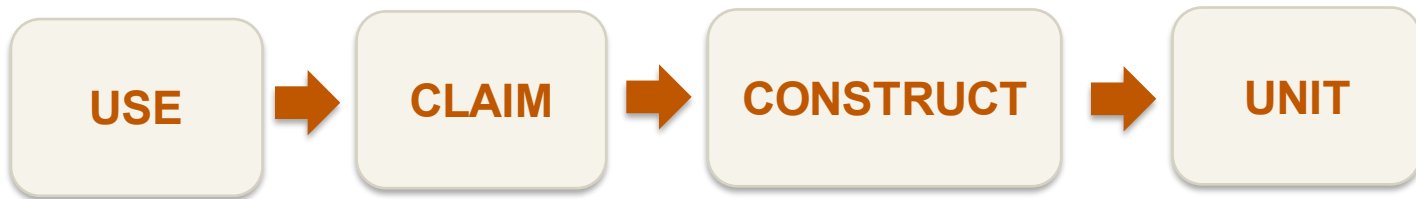
**AGGREGATION
+ UNCERTAINTY**

SLICES

DECISION

Benchmark design

Diagnosis & Action



What are we evaluating and why?

EVALUATION DESIGN

1. Start with the decision

- What will this evaluation change?
 - Choose a checkpoint?
 - Compare baseline vs fine-tuned model?
 - Decide whether a system is deployable?
 - Identify what to train next?

2. behavioral claim you want to make

- “Our model is better” → too vague
- “Our fine-tuned model answers campus policy questions more accurately than the baseline under student-like queries.” → testable

A useful claim names a desired behavior, setting, condition, and comparison target

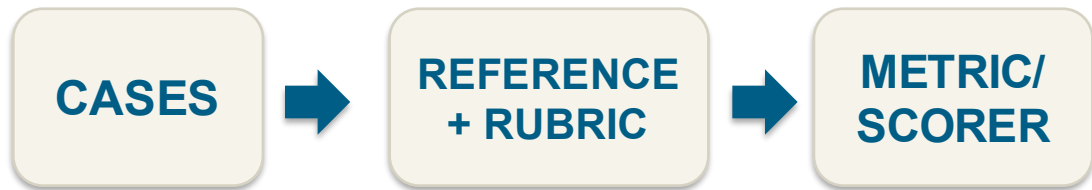
3. Construct

- Construct: the underlying capability/quality we intend to measure
- Claim: “more reliable answers”
- Possible constructs:
 - Factual correctness
 - Completeness
 - Grounding in source
 - Appropriate refusal / uncertainty
 - Clarity
- “helpful”, “reliable”, “safe” → break down into constructs
- Operationalization: turning a construct into observable measures

4. Unit of analysis

- What is one observation?
 - One token/span
 - One prompt-response pair
 - One multi-turn conversation
 - One document-summary pair
 - One tool-use trajectory (sequence of actions)
 - One user session

One evaluation row should correspond to the claim's unit of behavior



How will we measure it?

BENCHMARK DESIGN

Benchmark design = sampling design

- What population do these cases represent?
- Where do cases come from?
- What gets underrepresented?
- What gets overrepresented?

Case sources

Evaluation cases can be

- Naturally occurring
- Expert-written
- Crowd-written
- Template-generated
- Synthetic LLM-generated
- Adversarial / stress-test cases

Benchmark composition is a value judgment

Example case mix

- 60% common cases
- 20% difficult cases
- 10% ambiguous cases
- 10% high-risk cases

Changing the mix changes the meaning of the score

Task type should shape the metric

OUTPUT / OUTCOME	COMMON MEASURES
Class or action	Accuracy, precision/recall, F1
Structured output	Parse, schema, field accuracy
Ranked results	Recall@k, MRR, nDCG
Open response	Criteria, human/LLM judgment
Tool/agent outcome	Success, steps, errors, cost

Classification

- Accuracy: How often is the label correct?
- Precision: When predicted, how often is it right?
- Recall: How many target cases were found?
- Macro F1: How well across labels, equally weighted?

Always inspect which classes are confused, not only how many
Each class might carry different weight (risk)

Structured output

Layers of correctness

- Parse: Can we read the output?
- Schema: Are fields and values allowed?
- Field: Is each required value correct?

Different layers often require different fixes

Retrieval and Ranking

Measure the pipeline in parts

Retrieve → Rank → Generate

- Retrieve: recall@k; was relevant evidence found?
- Rank: MRR/nDCG; did useful evidence appear early
- Generate: grounded task success; did the answer use the retrieved items correctly?

Primary outcome + component diagnostics

Open-ended generation: decompose “quality”

CORRECT

GROUNDED

COMPLETE

CLEAR

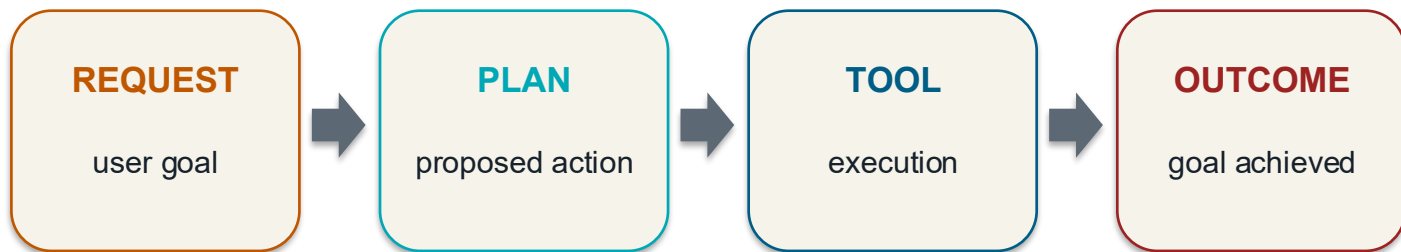
SAFE

CONCISE

- A single similarity score rarely represents every dimension.
- Similar to reference $\neq$ good answer
- Model-based metrics: Classifier-based evaluation, Learned reward models, LLM-based scoring
- Risk: Evaluator bias, Prompt sensitivity, Hidden failure modes

Tool/agent outcome

- Sometimes the response is not the outcome
- Budget/environment



Give each metric a role

- Primary: Did the intervention improve the target behavior?
- Diagnostic: Where and why does it succeed or fail?
- Guardrail: What must not regress or become unsafe?

A map of well-known LLM benchmarks

TARGET	EXAMPLES	TYPICAL SCORER
Reasoning	MMLU / GPQA / GSM8K / MATH / BBH	Exact / multiple choice
Instruction following	IFEval	Programmatic checks
Code	HumanEval / MBPP / SWE-bench	Tests / resolved task
Truthfulness / bias	TruthfulQA / BBQ	Labels / targeted judgments
General assistants	MT-Bench / AlpacaEval / Chatbot Arena	Human or LLM preference
Multimodal / holistic	MMMU / HELM / LiveBench	Mixed / multi-metric

Knowledge benchmarks

- MMLU / GPQA / GSM8K
- MMLU: Broad academic and professional knowledge, 57 subjects, multiple choice
- GPQA: Expert-level science questions, Depth and difficult oversight, diamond subset: 200 Q
- GSM8K / MATH: Worked mathematical problems, Answer checking is relatively objective, saturated

Verifiers when success is verifiable

- IFEval: Did the response satisfy each checkable instruction? Python verifier
- HumanEval: Does generated code pass unit tests?
- SWE-bench: Does a patch resolve a real repository issue?

When outcome is objective, try to use verifiers before LLM judge

Targeted benchmarks are designed to elicit certain failures

- Truthful QA
 - Questions built around misconceptions
 - questions selected to tempt models into imitating popular falsehoods
 - Design lesson: sample the failure mechanism
- BBQ (bias benchmark for QA)
 - Ambiguous ↔ disambiguated contexts
 - uses paired contexts to distinguish stereotype reliance under ambiguity
 - Design lesson: construct contrast pairs and slices

Common evaluation failure modes

- Vague claim
- Convenience test set
- Metric-construct mismatch
- Overinterpreting small differences
- Reporting only aggregate score
- Designing evaluation after seeing results

LAB 3 + PROPOSAL WORKSHOP

Timeline

- 1:40-2pm: Project idea discussion
- 2-2:30pm: Lab 3
- 2:30-2:45: Feedback

Project guidelines

The proposal should be concise (1–2 pages) and address:

1. Task and motivation: What behavior will you study and who or what setting motivates it?
2. Why post-training: Why might prompting or retrieval be insufficient? What evidence could change your mind?
3. Baseline: What model and prompting or retrieval baseline will you establish?
4. Evaluation: What is the evaluation unit, primary measure, held-out design, and validation plan?
5. Data: What training data may be used and what are its source, scale, quality?
6. Initial method: What intervention are you considering and why is it appropriate?
7. Feasibility and risks: What are the main compute, data, evaluation, or safety risks?
8. Team responsibilities: What will each team member initially own?

Lab 3: Evaluation Contract v0

Today you will specify

- Project title
- Claim → Construct → Unit
of analysis → Target
population → Case
buckets → Candidate
metrics → Two examples
- Complete by 2:30pm