CS336: Language Modeling from Scratch

Lecture 12: Evaluation

Stanford University - Spring 2025

Abstract

Abstract: This lecture provides a comprehensive examination of language model evaluation, addressing the fundamental question: "How good is a model?" We explore the evaluation crisis in modern AI, analyze popular benchmarks (MMLU, GPQA, HLE), examine instruction-following evaluation methods, and discuss agent benchmarks and safety evaluation. The lecture emphasizes that evaluation is not merely mechanical but profoundly shapes model development, requiring careful consideration of inputs, prompting strategies, output assessment, and result interpretation.

Contents

1 The Evaluation Crisis

1.1 Current State of Evaluation

The Problem: Despite numerous benchmarks and leaderboards, the field faces an "evaluation crisis" where traditional metrics may be saturated, gamed, or misaligned with real-world utility.

Symptoms of the Crisis:

- MMLU saturation (models achieving 90
- Benchmark gaming and optimization targeting
- Train-test contamination concerns
- Disconnect between benchmark performance and practical utility
- Proliferation of new benchmarks with diminishing returns

1.2 Fundamental Questions

Purpose-Driven Evaluation: There is no "one true evaluation" - the assessment must align with the underlying question:

- **Purchase Decision:** Which model best serves a specific use case?
- **Scientific Progress:** What are the raw capabilities and limitations?
- **Policy Assessment:** What are the benefits and risks at the current time?
- **Development Feedback:** How can we improve the model?

2 Evaluation Framework

2.1 Input Design

Critical Questions:

- **Source and Coverage:** Which use cases and domains are represented?
- **Difficulty Distribution:** Are there challenging edge cases or only easy examples?
- **Tail Representation:** Does the evaluation capture rare but important scenarios?
- **Adaptation:** Should inputs be tailored to specific models?

Adaptation Trade-offs:

- **Necessity:** Multi-turn conversations require model-adaptive inputs
- **Efficiency:** Red teaming benefits from targeted, model-specific prompts
- **Comparability:** Adaptation complicates cross-model comparison

2.2 Prompting Strategies

Strategy Options:

- **Zero-shot:** Direct task instruction without examples
- **Few-shot:** Including demonstration examples
- **Chain-of-thought:** Encouraging step-by-step reasoning
- **Tool use:** Allowing calculator, web search, or code execution

Impact on Results: Prompting strategy significantly affects performance, with order, format, and example selection all introducing variance.

2.3 Output Assessment

Assessment Challenges:

- **Reference Quality:** Are ground truth answers error-free and comprehensive?
- **Metric Selection:** Pass@1 vs. Pass@10, exact match vs. semantic similarity
- **Cost Integration:** How to factor computational expense into evaluation
- **Error Weighting:** Not all mistakes have equal real-world impact
- **Open-ended Generation:** No clear "correct" answer for creative tasks

2.4 Result Interpretation

Interpretation Questions:

- **Absolute vs. Relative:** Is 91
- **Generalization:** Does performance indicate true capability acquisition?
- **Contamination:** Is performance inflated by train-test overlap?
- **Target Assessment:** Are we evaluating the model, system, or method?

3 Perplexity and Language Modeling

3.1 Perplexity as Foundation

Definition: Perplexity measures how well a language model assigns probability to a held-out dataset, serving as the fundamental metric for language modeling capability.

Historical Context:

- **2010s Standard:** Penn Treebank, WikiText, One Billion Word Benchmark
- **Traditional Approach:** Train on designated split, evaluate on test split
- **Research Focus:** Architecture improvements for perplexity reduction

3.2 GPT Era Transformation

GPT-2 Paradigm Shift:

- Training on 40GB of diverse web text
- Zero-shot evaluation on traditional benchmarks
- Out-of-distribution performance through broad training
- Transfer learning superiority on small datasets

Modern Transition: Post-GPT focus shifted from perplexity to downstream task performance, though perplexity remains valuable.

3.3 Perplexity Advantages

Benchmark Analysis

Why Perplexity Remains Valuable:

- **Smoothness:** Fine-grained token probabilities vs. binary correct/incorrect
- **Scaling Laws:** Enables clean curve fitting for predictive modeling
- **Universality:** Evaluates every token rather than curated subsets
- **Unbiased:** Harder to game than task-specific metrics
- **Downstream Application:** Can be computed for specific tasks

3.4 Perplexity Limitations

Trust Requirements: Requires access to model probabilities and verification of valid distributions.

Implementation Pitfalls: Easy to introduce bugs in probability computation and normalization.

3.5 The Perplexity Maximalist View

Theoretical Argument: If true distribution is t and model is p, optimal perplexity achieved when $p = t$, potentially solving all tasks.

Counter-argument: May be inefficient to push down on irrelevant parts of the distribution rather than focusing on curated capabilities.

4 Knowledge and Reasoning Benchmarks

4.1 MMLU (Massive Multitask Language Understanding)

Overview:

- **Origin:** 2020, designed for GPT-3 era base models
- **Structure:** 57 subjects, multiple-choice questions from web sources

- **Current State:** Approaching saturation with 90

Historical Performance:

- GPT-3 (2020): 45
- Modern models: 90

Evaluation Insights:

- Better suited for evaluating base models than instruction-tuned models
- Performance reflects knowledge acquisition, not language understanding
- Subject to train-test contamination concerns

4.2 MMLU-Pro

Improvements over MMLU:

- Removed noisy and trivial questions
- Increased from 4 to 10 answer choices
- Encouraged chain-of-thought reasoning
- Restored differentiation between frontier models

4.3 GPQA (Graduate-Level Google-Proof Q&A)

Design Philosophy: PhD-level questions that experts can solve but non-experts cannot, even with Google access.

Creation Process:

1. PhD-level question writers in specialized domains
2. Expert validation and feedback cycles
3. Non-expert testing with 30-minute Google access
4. Multi-stage quality control

Performance Benchmarks:

- Domain experts: 65
- Non-experts with Google: 30
- GPT-4 (original): 39
- O3: 75

4.4 Humanity's Last Exam (HLE)

Extreme Difficulty Approach:

- Multimodal questions requiring expert knowledge
- Prize pool incentives for question creation
- Frontier model filtering to reject "easy" questions
- Co-authorship offers to question creators

Current Performance: O3 achieving 20

Critical Challenge

Bias in Question Creation: Open calls for questions attract LLM-aware contributors, potentially creating artificially specific rather than representative difficulty.

5 Instruction Following Evaluation

5.1 Chatbot Arena

Methodology:

- Users submit prompts to anonymous model pairs
- Pairwise preference judgments
- ELO rating system for model ranking
- Dynamic, continuously updated evaluation

Advantages:

- Fresh, non-static evaluation data
- Real user preferences and use cases
- Accommodates new model releases
- Reflects practical utility

Recent Controversies:

- "Leaderboard Illusion" - privileged access and gaming
- Protocol issues with evaluation methodology
- Questions about user representativeness

5.2 IFEval (Instruction Following Evaluation)

Narrow Focus: Tests constraint-following ability with automatically verifiable requirements.

Constraint Types:

- Word/sentence count requirements
- Specific word inclusion/exclusion
- Formatting requirements
- Structural constraints

Limitations: Evaluates constraint adherence, not content quality or semantic correctness.

5.3 AlpacaEval

LLM-as-Judge Approach: Uses GPT-4 to compute win rates against reference model outputs.

Known Issues:

- Length bias - longer responses initially scored higher
- Model bias - GPT-4 potentially favoring similar responses
- Corrected with length-normalized variants

Validation: Correlation with Chatbot Arena provides confidence in automated approach.

5.4 WildBench

Real Conversation Data: Evaluation based on actual human-bot conversation logs.

Structured Assessment: LLM judges with checklists ensuring comprehensive response evaluation.

6 Agent Benchmarks

6.1 SWEBench (Software Engineering)

Task Structure:

- Real GitHub issues and codebases
- Goal: Generate PR that passes unit tests
- Requires code understanding, modification, and testing

Evaluation Approach: Success measured by test suite pass rate after patch application.

6.2 CyBench (Cybersecurity)

Capture-the-Flag Format:

- Agent must hack into servers to retrieve secret keys
- Requires iterative command execution and reasoning
- Success binary: key retrieved or not

Human Baseline Context: Some challenges took human teams up to 24 hours to solve.
Current Performance: 20

6.3 MLEBench (Machine Learning)

Kaggle Competition Format:

- 75 real Kaggle competitions
- Agent must understand data, train models, debug, and optimize
- Full ML pipeline evaluation

Success Metric: Achieving medal-level performance (competitive threshold).

6.4 ARC AGI (Abstract Reasoning)

Pure Reasoning Focus: Pattern recognition tasks without language or domain knowledge requirements.

Design Philosophy: Factor out memorization to focus on reasoning and creativity.
Performance Evolution:

- Traditional models: 0
- O3 with high compute: Significant improvement (but expensive per task)

7 Safety Evaluation

7.1 HarmBench

Harmful Behavior Assessment: 510 identified harmful behaviors tested through direct prompting.

Example Evaluation: Request for dangerous chemical synthesis instructions.
Response Classification: Compliance vs. appropriate refusal measurement.

7.2 AIR-Bench

Regulatory Grounding: Safety evaluation anchored in actual legal frameworks and company policies.

Taxonomy Development: Systematic categorization of safety concerns based on real-world regulations.

Practical Relevance: Connects abstract safety concepts to concrete compliance requirements.

7.3 Jailbreaking

Adversarial Safety Testing: Automated prompt optimization to bypass safety measures.

Methodology: Optimize prompt suffixes to induce harmful completions from safety-trained models.

Transferability: Attacks developed on open models often transfer to closed models.

Critical Challenge

Safety-Capability Tension: High refusal rates can appear safe but may indicate reduced helpfulness. Safety evaluation must balance harm prevention with utility preservation.

7.4 Pre-deployment Testing

Institutional Framework: Safety Institutes from US, UK, and other countries conduct voluntary pre-release evaluations.

Process: Early model access, comprehensive safety testing, feedback to developers before public release.

Limitations: Voluntary participation, no binding enforcement, undefined safety standards.

8 Critical Evaluation Challenges

8.1 Train-Test Contamination

The Problem: Models may have seen test data during training, inflating performance estimates.

Detection Methods:

- N-gram overlap detection
- Document-level deduplication
- Paraphrase and translation detection

Persistent Issues:

- Near-duplicates escape detection
- Translated versions of test content
- Quoted test materials in training data

8.2 Benchmark Gaming

Goodhart's Law: "When a measure becomes a target, it ceases to be a good measure."

Gaming Strategies:

- Direct optimization for benchmark performance
- Training data curation to include benchmark-like content
- Prompt engineering specific to evaluation format
- Length manipulation for automated judges

8.3 Evaluation Bias

Selection Bias: Question creators often have specific expertise and exposure to existing models.

Cultural Bias: Benchmarks may reflect particular cultural, linguistic, or educational perspectives.

Use Case Bias: Focus on expert-level tasks may not reflect general population needs.

8.4 Context Dependency

Safety Contextualization: Harmful content depends on context, user, and application.

Cultural Variation: Safety norms vary across legal systems and social contexts.

Dynamic Standards: Safety and capability standards evolve over time.

9 Evaluation Best Practices

9.1 Multi-faceted Assessment

Comprehensive Evaluation: No single metric captures all aspects of model quality.

Benchmark Portfolio: Combine knowledge, reasoning, instruction-following, and safety evaluations.

Cost-Performance Trade-offs: Include computational cost in evaluation frameworks.

9.2 Methodological Rigor

Clear Objectives: Define evaluation purpose before selecting metrics.

Baseline Establishment: Include human performance benchmarks where possible.

Error Analysis: Examine failure modes, not just aggregate performance.

Reproducibility: Document prompting strategies, evaluation procedures, and model configurations.

9.3 Dynamic Evaluation

Fresh Data: Regularly introduce new evaluation content to prevent gaming.

Adaptive Testing: Adjust evaluation difficulty based on model capabilities.

Real-world Validation: Correlate benchmark performance with practical deployment outcomes.

10 Future Directions

10.1 Evaluation Innovation

Beyond Static Benchmarks: Development of dynamic, interactive evaluation environments.

Holistic Assessment: Integration of capability, safety, and efficiency measurements.

Meta-evaluation: Better methods for evaluating evaluation quality itself.

10.2 Emerging Challenges

Multimodal Evaluation: Extending evaluation frameworks to vision-language models.

Agent Assessment: Long-horizon, multi-step evaluation scenarios.

Personalization: Evaluation methods that account for individual user preferences and needs.

11 Evaluation Mindmap and Overview

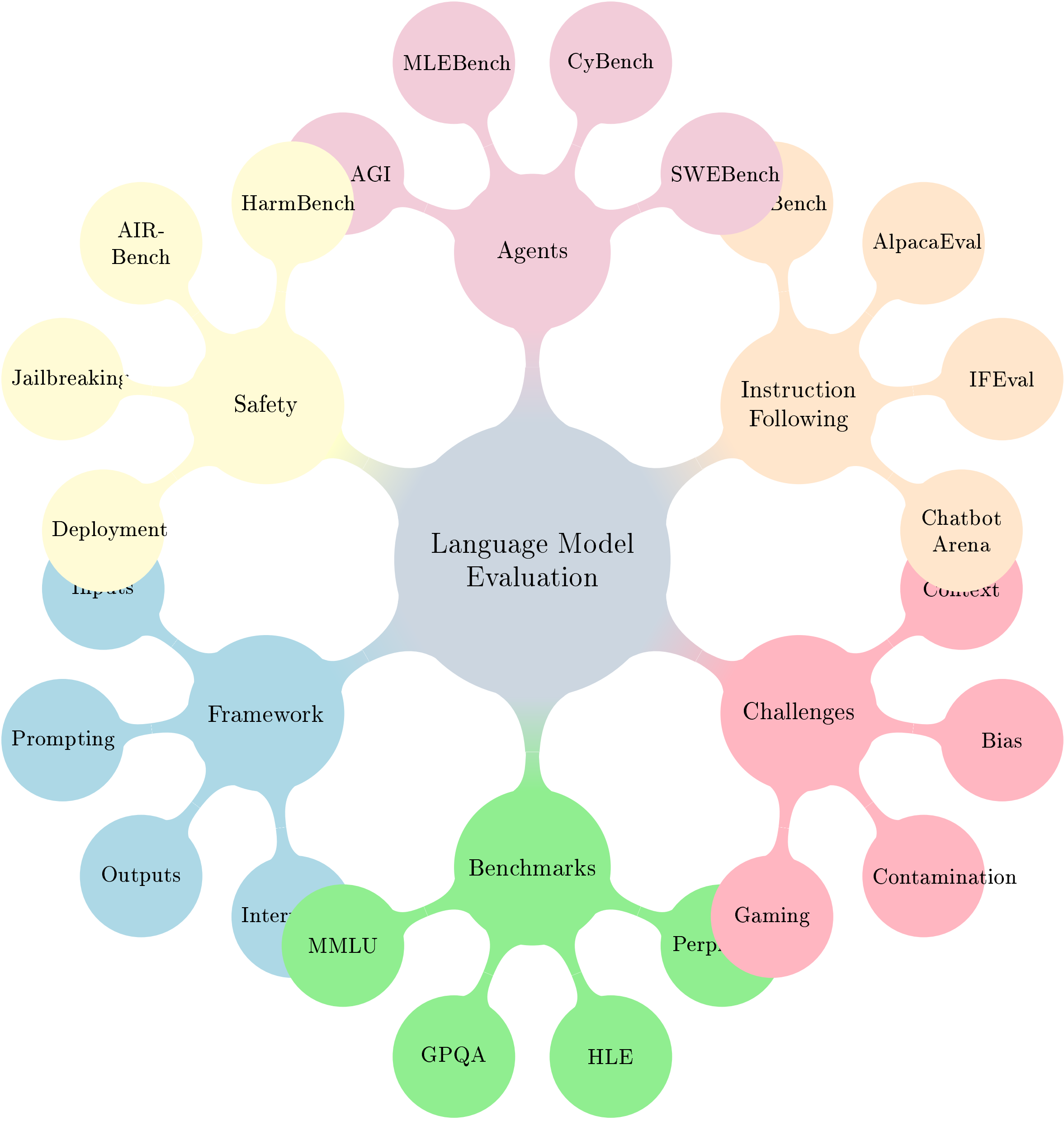

Figure 1: Comprehensive mindmap of language model evaluation covering framework, benchmarks, challenges, instruction following, agents, and safety considerations.

11.1 Mindmap Description

The evaluation mindmap illustrates the complex, interconnected nature of language model assessment:

Central Hub - Language Model Evaluation: The core challenge of determining model quality, which appears simple but reveals profound complexity upon deeper examination.

Framework Branch (Blue): The systematic approach to evaluation encompassing input selection, prompting strategies, output assessment, and result interpretation - each requiring careful consideration to avoid misleading conclusions.

Benchmarks Branch (Green): Traditional evaluation methods including knowledge-based tests (MMLU), expert-level assessments (GPQA), extreme difficulty challenges (HLE), and foundational perplexity measures that form the backbone of model comparison.

Challenges Branch (Red): Critical issues threatening evaluation validity including benchmark gaming, train-test contamination, selection bias, and contextual dependencies that can invalidate evaluation results.

Instruction Following Branch (Orange): Modern evaluation approaches for interactive models including human preference systems (Chatbot Arena), constraint satisfaction tests (IFEval), automated judging (AlpacaEval), and real-world conversation analysis (WildBench).

Agents Branch (Purple): Complex multi-step evaluation scenarios including software engineering (SWEBench), cybersecurity (CyBench), machine learning competitions (MLEBench), and pure reasoning challenges (ARC AGI) that test beyond single-turn capabilities.

Safety Branch (Yellow): Critical assessment of harmful behavior potential including direct harm evaluation (HarmBench), regulatory compliance (AIR-Bench), adversarial robustness (Jailbreaking), and pre-deployment testing protocols.

Evaluation Insight

The mindmap reveals that evaluation is not a single metric but an ecosystem of interconnected assessment methods, each with distinct purposes, limitations, and validity conditions. The complexity reflects the fundamental challenge of measuring intelligence and capability in artificial systems.