Roadmap for Benchmark Construction and Evaluation Framework

For Incoming PhD Students

Overview

This document provides a six-month, step-by-step guideline and timeline for constructing and evaluating a benchmark (LLM benchmark) in operations research and related fields. It is designed for first-year PhD students and includes literature review, data collection, framework development, and pilot evaluation.

Overall Timeline (6 Months)

Phase	**Month**	**Main Objective**
Phase 1: Preparation & Literature Review	1	Clarify goals; survey foundational textbooks and existing benchmarks.
Phase 2: Question Collection & Draft Bank	2	Gather 300–500 candidate problems; set up version control/database.
Phase 3: Categorization & Annotation Design	3	Group problems by topic; write reference solutions and common-error annotations.
Phase 4: Dialogue Formatting & Data Construction	4	Convert Q&A into multi-turn dialogue format; validate data schema.
Phase 5: Evaluation Framework Development	5	Define metrics; implement automated scoring scripts and unit tests.
Phase 6: Pilot Run & Refinement	6	Run pilot on 30–50 items; analyze results; refine data, scripts, and documentation.

1 Step-by-Step Guide

Phase 1: Preparation & Literature Review (Month 1)

1. **Define Scope & Objectives**
   - Review the project brief: what constitutes a "benchmark" for LLM performance?
   - Set specific goals (e.g., coverage of estimation, hypothesis testing, optimization, pricing).
2. **Literature Survey**
   - Read foundational textbooks:
     - Casella & Berger (2002), *Statistical Inference*
     - Shreve (2004), *Stochastic Calculus for Finance*
     - Björk (2009), *Arbitrage Theory in Continuous Time*
   - Study existing LLM benchmarks (e.g., MMLU, HumanEval): construction methods, strengths, weaknesses.
3. **Source Identification**
   - List potential problem sources: qualifying exams, course assignments, published problem sets.
   - Obtain access (with advisor) to these repositories.

Phase 2: Question Collection & Draft Bank (Month 2)

1. **Multi-Channel Collection**
   - Extract problems from past qualifying exams, homework, and textbook exercises.
   - Aim for 300–500 varied questions.
2. **Version Control & Metadata**
   - Use Git (GitHub/GitLab) to manage the question bank.
   - Maintain a spreadsheet or database with fields: ID, source, topic, difficulty, keywords.
3. **Initial Filtering**
   - Remove duplicates, trivial or overly obscure questions.
   - Ensure broad coverage: estimation, testing, optimization, pricing, etc.

Phase 3: Categorization & Annotation Design (Month 3)

1. **Topic Classification**
   - Define domain tags: Estimation, Hypothesis Testing, Portfolio Optimization, Options Pricing, etc.
   - Assign each problem one or more tags.
2. **Reference Solutions**
   - Write a clear, concise standard solution for each problem.
   - Identify and document 1–3 common errors or misconceptions per problem.

Phase 4: Dialogue Formatting & Data Construction (Month 4)

1. **Multi-Turn Dialogue Design**
   - Structure each problem as a conversation:
     (a) Student asks the question.
     (b) Tutor (LLM) gives an initial answer.
     (c) Student requests clarification or points out errors.
     (d) Tutor provides a full, detailed derivation.
2. **Data Schema & Validation**
   - Example JSON record:

```
{
  "id": "...",
  "topic": "Estimation",
  "dialogue": [
    {"role": "student", "text": "..."},
    {"role": "tutor",   "text": "..."}
  ],
  "solution": "...",
  "common_errors": ["..."]
}
```

   - Write a small validation script to enforce required fields and types.

Phase 5: Evaluation Framework Development (Month 5)

1. **Define Evaluation Metrics**
   - *Accuracy*: correctness of numerical answers or conclusions.
   - *Completeness*: presence and logical order of key solution steps.
   - *Error Detection*: ability of the model to identify documented common errors.
2. **Automation Scripts**
   - Implement Python scripts using SymPy or NumPy for symbolic/numeric checks.
   - Write unit tests to verify scoring logic across different question types.

Phase 6: Pilot Run & Refinement (Month 6)

1. **Pilot Execution**
    - Randomly select 30–50 problems.
    - Generate responses with your target LLM (e.g., GPT-4, LLaMA).
    - Run evaluation scripts and collect metrics.
2. **Analysis & Iteration**
    - Review pilot results to identify weaknesses in problem set, dialogue design, or scoring.
    - Update tags, solutions, common errors, and scripts accordingly.
3. **Documentation**
    - Prepare a README, data dictionary, and user guide.
    - Schedule a review meeting with advisor/lab for feedback.