Data Engineering and MLOps in Business

MLOps: LLM Benchmarking

Eskil Olav Andersen

AAUBS

March 31, 2025

eoa@business.aau.dk

Outline

Benchmarking LLMs by Major AI Firms

- **Purpose of Benchmarking:**
  - Evaluate and compare the performance of LLMs across various tasks.
  - Identify strengths and areas for improvement in model capabilities.
- **Common Evaluation Metrics:**
  - **Accuracy:** Measures the correctness of model outputs.
  - **Latency:** Assesses the response time of the model.
  - **Throughput:** Evaluates the number of tasks processed in a given time frame.
  - **Cost-efficiency:** Analyzes the computational resources required relative to performance.

Prominent LLM Benchmarks

- **MMLU (Massive Multitask Language Understanding):**
  - Comprises approximately 16,000 multiple-choice questions across 57 subjects, including mathematics, philosophy, law, and medicine.
  - Widely used to assess the breadth and depth of LLM knowledge.
- **HumanEval:**
  - Contains 164 programming problems designed to evaluate code generation capabilities of LLMs.
  - Focuses on functional correctness using the pass@k metric.
- **Open LLM Leaderboard:**
  - Provides a platform to compare LLMs based on metrics like accuracy, speed, and versatility.
  - Assists developers in understanding model strengths and guiding selection for specific applications.

Limitations of General LLM Benchmarks

- **Broad Scope:** General benchmarks assess a wide range of tasks, which may not align with the specific requirements of individual projects.
- **Lack of Domain Specificity:** These benchmarks often fail to capture nuances and complexities inherent in specialized fields, leading to an incomplete evaluation of model performance in those areas.
- **Potential for Misleading Results:** Relying solely on general benchmarks can result in overestimating a model's effectiveness for a particular application, as high scores on broad metrics do not guarantee suitability for specialized tasks.

Necessity for Custom Evaluation Strategies

- **Tailored Assessments:** Designing custom benchmarks allows for evaluation criteria that directly reflect the goals and challenges of the specific project, ensuring more relevant performance insights.
- **Enhanced Relevance:** Custom evaluations can incorporate real-world scenarios and data relevant to the application, providing a more accurate measure of model effectiveness in the intended context.
- **Continuous Improvement:** Implementing project-specific benchmarks facilitates ongoing monitoring and iterative refinement of the model, leading to sustained performance aligned with evolving project needs.

Approaches to Evaluating LLMs

- **Human Evaluation:**
  - Involves domain experts or crowdworkers assessing model outputs based on predefined criteria.
  - Provides nuanced insights into aspects like coherence, relevance, and ethical considerations.
  - Challenges include scalability, potential biases, and resource intensiveness.
- **Automated Evaluation:**
  - Utilizes computational methods and metrics to assess model performance.
  - Offers scalability and consistency in evaluations.
  - May lack the depth of understanding that human evaluation provides.

Tools for LLM Evaluation

- **Human Evaluation Tools:**
  - **Toloka:** A crowdsourcing platform facilitating data labeling and human evaluation tasks, supporting AI development from training to evaluation. :contentReference[oaicite:0]index=0
  - **LLM Comparator:** An interactive tool for side-by-side human assessments of LLM responses, providing both quantitative and qualitative insights.
- **Automated Evaluation Tools:**
  - **OpenAI Evals:** An open-source framework enabling developers to design and execute custom tests for LLMs, fostering a community-driven approach to evaluation.
  - **DeepEval:** An open-source framework that automates LLM evaluations using various metrics, including answer relevancy and hallucination detection.

Evaluating Different LLMs Using Agentic Frameworks

- **Objective:** Compare how various LLMs respond to identical prompts to assess their performance and suitability for specific tasks.
- **Methodology:**
  - Utilize an agentic framework where each LLM acts as an agent processing the same set of prompts.
  - Collect and analyze outputs based on predefined evaluation metrics such as accuracy, coherence, and relevance.
- **Example:**
  - **Prompt:** "Summarize the key findings of the latest climate change report."
  - **LLMs Evaluated:** Model A, Model B, Model C.
  - **Evaluation:** Compare summaries generated by each model for factual accuracy, conciseness, and readability.

Evaluating Prompt Engineering Using Agentic Frameworks

- **Objective:** Assess how different prompt formulations affect the output of a single LLM to optimize prompt design.
- **Methodology:**
  - Implement an agentic framework where the LLM processes various rephrasings of a prompt.
  - Evaluate outputs based on consistency, informativeness, and alignment with desired responses.
- **Example:**
  - **Original Prompt:** "Explain the theory of relativity."
  - **Revised Prompts:**
    - "Provide a brief overview of Einstein's theory of relativity suitable for a high school student."
    - "Describe the key principles of the theory of relativity in simple terms."
  - **Evaluation:** Analyze how each prompt variation influences the clarity and depth of the LLM's response.

Agentic Framework for LLM Output Evaluation - Example

- **Critical Reviewer Agent:**
  - Evaluates the depth of analysis and logical coherence in the LLM's response.
  - Identifies areas lacking critical insight or depth.
- **Style Analyst Agent:**
  - Assesses the writing style for clarity, tone, and adherence to specified guidelines.
  - Ensures consistency and appropriateness for the target audience.
- **Accuracy Checker Agent:**
  - Verifies the factual correctness of information presented in the LLM's output.
  - Cross-references claims with reliable sources to detect inaccuracies.
- **Summarization Agent:**
  - Compiles the evaluations of the other agents into a cohesive

Performance Monitoring

- **Tracking Performance Metrics:**
  - Latency, throughput, accuracy
  - Resource consumption (memory, GPU, etc.)
- **Detecting Drift and Concept Shift:**
  - Monitor distribution of input/output data over time
  - Update model if data shifts
- **Logging and Debugging:**
  - Collect logs for analysis
  - Automate alerts for performance drops

Benchmarking LLMs

- **Common Benchmarks:**
  - MMLU (Massive Multitask Language Understanding)
  - HellaSwag, TruthfulQA, etc.
- **Custom Benchmarking:**
  - Create test cases based on real-world data
  - Include diverse edge cases and rare scenarios
- **Evaluation Metrics:**
  - BLEU, ROUGE, Accuracy, F1 Score
  - Latency and cost per request

Test 1: Lower Model Size

- **Goal:** Reduce model size to improve efficiency (costs)
- **Example:** Move from LLaMA 13B to LLaMA 7B
- **Trade-offs:**
  - Lower memory and cost requirements
  - Possible loss of performance in complex tasks
- **Evaluation:**
  - Track loss in accuracy vs. latency improvements
  - Monitor GPU utilization and response time

Test 2: Improve the Prompt

- **Goal:** Keep the same model but improve prompt quality
- **Example:**
  - Original: "Write a story about a cat."
  - Improved: "Write a creative and humorous short story about a mischievous cat who gets into trouble with a dog."
- **Techniques:**
  - Add more context and examples
  - Use chain-of-thought prompting
  - Few-shot vs. zero-shot prompting
- **Evaluation:**
  - Compare output quality using BLEU and human evaluation
  - Monitor latency changes

Best Practices

- A/B testing for model and prompt changes
- Monitor for ethical bias and fairness issues
- Monitor cost-performance trade-offs
- Optimize resource utilization and scaling

Questions?

Thank you for your attention!

Questions?