A Systematic Framework for LLM Evaluation

Abi Aryan

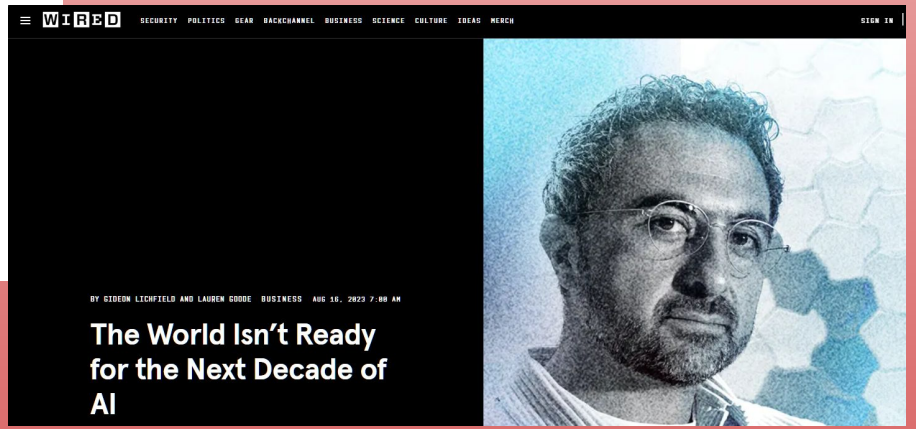
Embrace:AI Meetup, 20th March 2025

One of the hardest problems for LLMs-

Evaluation

Evaluating LLMs is a minefield

Arvind Narayanan & Sayash Kapoor
Princeton University



Evaluation for Large Language Models (LLMs) is the process of assessing their performance and capabilities.

It involves a combination of methods to determine how well an LLM achieves its intended purpose and adheres to ethical guidelines.

- **Goals**: Identify strengths and weaknesses of the LLM
- **Methods**: g human experts, pre-defined benchmarks with specific tasks and metrics, or even other LLMs specifically designed for evaluation.
- **Challenges**: The scale of LLM capabilities and the difficulty of defining "good" performance make evaluation complex.
- **Outcomes**: Evaluation results

Today, we will talk about:

#1. Why Evaluation is Hard for LLMs?

#2. A Systematic Framework for Evaluating LLMs

#3. Reasoning LLMs

#4. Open Challenges

#1.
Why Evaluation is hard for LLMs?

Evaluating LLMs is hard

1. **Scale of the problem**: LLMs are trained on massive amounts of data, and the number of possible inputs they can receive is practically infinite.

   This makes it impossible to exhaustively test them on every scenario, unlike simpler programs. Even evaluating a tiny fraction of possibilities is a monumental task.

Some of the possibilities would be:-

- ❏ **Informativeness & Factuality** - *Assessing how well the outputs are informative and correspond to factual information.*
- ❏ **Fluency & Coherence** - *Measuring how well the outputs are grammatically correct, readable, and follow a logical flow.*
- ❏ **Engagement & Style** - *Evaluating how engaging and interesting the LLM's outputs are, along with stylistic aspects.*
- ❏ **Safety & Bias** - *Potential biases or harmful content generated by the LLM.*
- ❏ **Grounding** - *Assessing how well the LLM's response is grounded in real-world information and avoids hallucinations.*
- ❏ **Efficiency** - *Measuring the computational resources required by the LLM to generate outputs.*

Evaluating LLMs is hard

2. Defining "good": Unlike some models where there's a clear success metric (think image recognition accuracy), what constitutes a "good" response from an LLM can be subjective. Is it providing relevant information? Is it creative? Is it factually accurate? These goals can conflict, making it hard to design a single metric that captures everything.

“Good performance” can mean several things-

- **Task-Specific Success:** *An LLM's "goodness" is highly dependent on the task it's designed for.*
- **Accuracy and Factuality:** *For tasks like question answering or summarizing topics, an LLM should be demonstrably accurate and avoid generating false or misleading information.*
- **Fluency and Coherence:** *The language should be appropriate for the context and audience.*
- **Relevance and Informativeness:** *The LLM's response should be relevant to the prompt or query and provide useful information. It shouldn't go off on tangents or introduce irrelevant details.*
- **Engagement and Creativity:** *Depending on the context, "good" might mean generating outputs that are interesting, engaging, or even surprising.*
- **Safety and Fairness:** *An LLM shouldn't generate harmful content, promote biases, or perpetuate stereotypes. It should be fair and inclusive in its responses.*
- **Efficiency:** *Ideally, an LLM should be able to generate good outputs while using computational resources efficiently. This becomes important for real-world applications where processing power is limited.*

LLM Evaluation Criteria

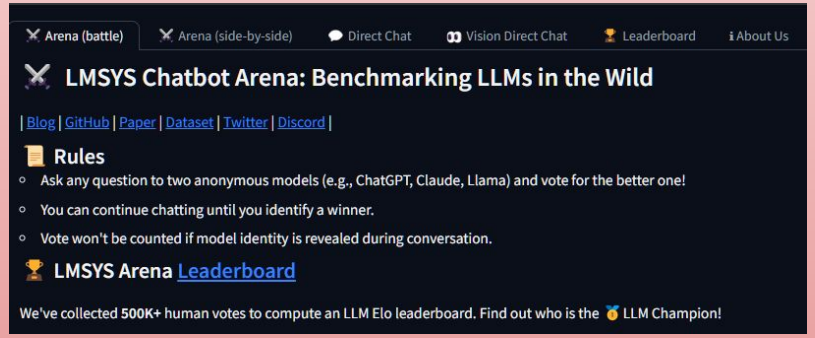
Evaluation Criteria	**Broad goals for LLM performance**	**Task Understanding, Reasoning Capabilities, Appropriateness, Safety**
Language Features	Focus on core language skills	Fluency, Coherence, Factuality
Task Independence	Metrics applied across various tasks	Toxicity, Fairness, Bias
Task Dependence	Metrics specific to a particular task	Relevance (Question Answering), Appropriateness (Machine Translation)
Metrics	Quantitative measures of performance	BLEU (Machine Translation), ROUGE (Summarization), Accuracy (Question Answering)
LLM Specific Metrics	Pre-defined datasets and metrics for specific tasks	GLUE, SuperGLUE, HellaSwag
Custom Metrics	User-defined metrics based on specific needs	Guideline Adherence, Presence of Certain Words
Frameworks	Tools and platforms for conducting evaluations	Ragas, Promptfoo, RAG Triad
Human Evaluation	Judgments by human experts (considered Gold Standard)	LMSys Arena

Some Limitations

- **Human Evaluation Limitations**: While Human Evaluation is considered the gold standard, it can be expensive, time-consuming, and prone to subjectivity.

- **Benchmark Limitations**: Pre-defined benchmarks can be susceptible to reverse-engineering, where the model learns to perform well on the benchmark without necessarily generalizing to real-world tasks.

- **LLM Evaluator Limitations:** Powerful but potentially biased, opaque, and resource-intensive.



BLEU (Machine Translation), ROUGE (Summarization), Accuracy (Question Answering)

GLUE, SuperGLUE, HellaSwag

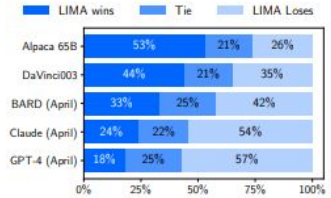


Figure 1: Human preference evaluation, comparing LIMA to 5 different baselines across 300 test prompts.

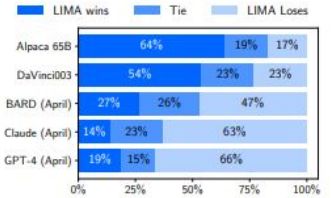
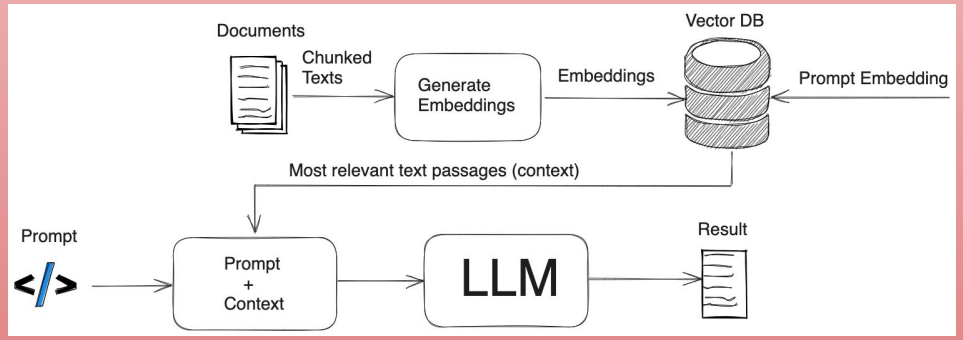


Figure 2: Preference evaluation using GPT-4 as the annotator, given the same instructions provided to humans.

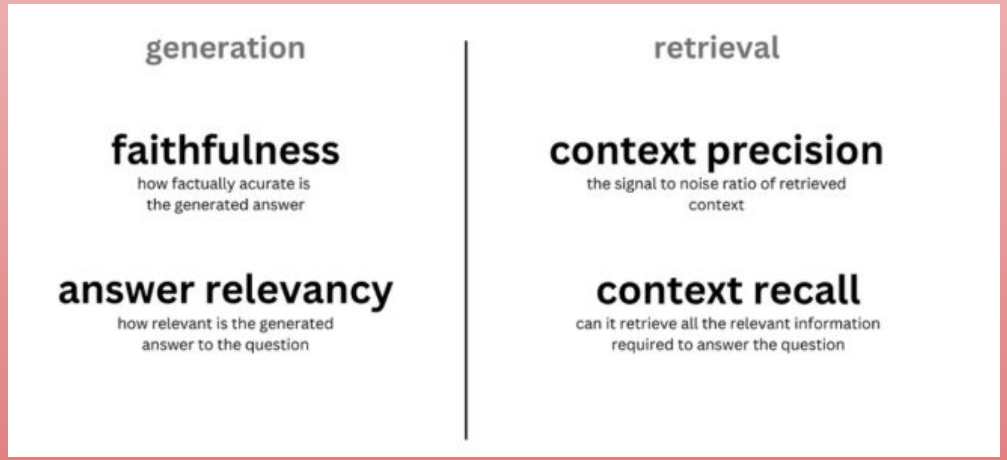
#2.
A Systematic Framework for LLM Evaluation

A Simple RAG Application

1. **User Input:** The user submits a question or prompt to the RAG application.
2. **Retrieval:** The application utilizes a retrieval system to search through a database of relevant documents or text data. This database could contain articles, manuals, code snippets, or any other information relevant to the LLM's domain.
3. **Matching:** The retrieval system identifies the most relevant portions of the data based on the user's query using techniques like vector similarity search.
4. **Prompt Augmentation:** This might involve concatenating the user's original query with snippets from the retrieved data.
5. **LLM Generation:** The augmented prompt is then sent to the LLM which uses the additional context provided by the retrieved information to generate a response.
6. **Output:** Finally, the LLM's response is presented to the user.



Two main categories of metrics-



1. Retrieval
2. Generation

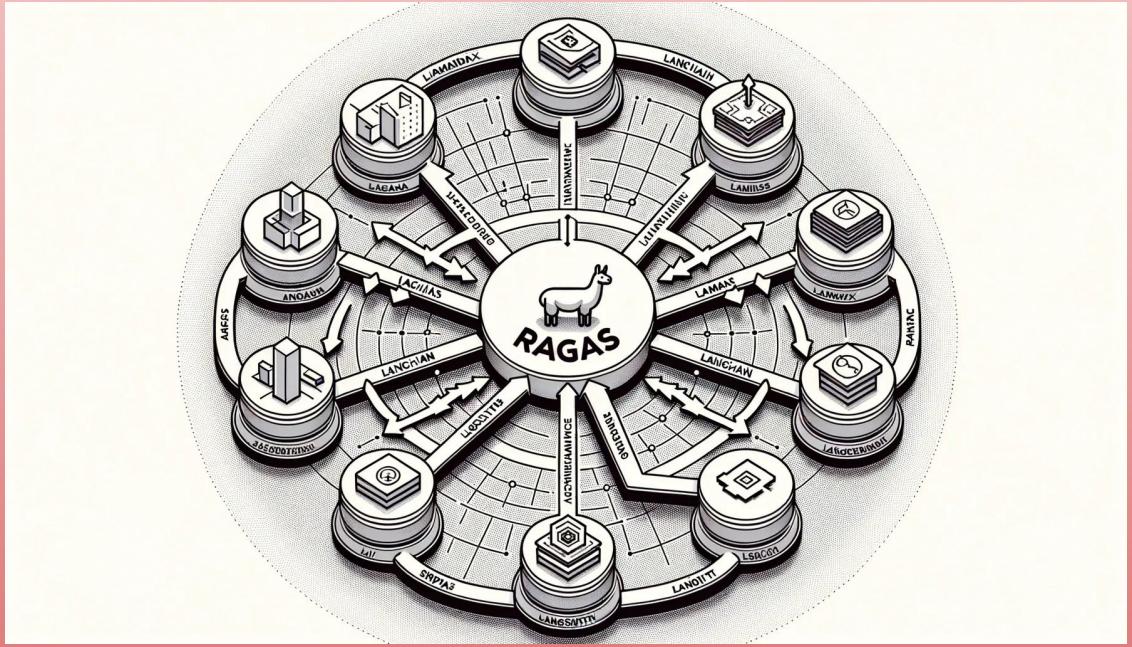
Retrieval Metrics

These metrics assess the effectiveness of the retrieval component in finding relevant information for the LLM.

- **Recall**: This measures the proportion of relevant documents retrieved from the database compared to all the truly relevant documents available. A high recall indicates the retrieval system captures most of the valuable information.
- **Mean Reciprocal Rank (MRR)**: This metric considers the rank of the first relevant document retrieved for each query. A higher MRR signifies the relevant documents are typically found early in the search results.
- **Mean Average Precision (MAP)**: This metric takes the average of the precision (proportion of relevant documents among retrieved documents) at each position in the ranked list. A higher MAP indicates the retrieved documents are consistently relevant throughout the list.
- **Context Recall**: This metric specifically focuses on whether the retrieved information directly addresses the user's query and provides context relevant to the LLM's task.
- **Context Precision**: Similar to recall, it measures the proportion of retrieved information that is truly relevant and useful for the LLM's generation process.
- **Context Relevance**: This broader metric combines aspects of recall and precision, evaluating how well the retrieved information aligns with the specific needs of the LLM for generating an accurate and focused response.

Retrieval Evaluators

1. **Recall:** Catching most relevant info
2. **MRR:** Finding key info early
3. **MAP:** Consistent relevance throughout retrieved info
4. **Context Recall:** Retrieving info specific to user query and LLM task
5. **Context Precision:** Retrieving info truly useful for LLM
6. **Context Relevance:** Aligning retrieved info with LLM's needs



Using Frameworks like RAGAS

1. Pre-built Functionality:

RAGAS provides built-in functions for calculating various Retrieval Metrics, including Recall, MRR, MAP, Context Recall, and Context Precision. This eliminates the need to write custom code for each metric.

2. Streamlined Workflow:

RAGAS offers a structured approach for feeding your retrieval predictions and gold standard data (relevant documents for each query). It then automatically calculates the desired metrics.

3. Ease of Use:

RAGAS is designed to be user-friendly, with clear documentation and examples. This makes it easier for researchers and developers, even those without extensive coding experience, to evaluate their RAG systems.

Measuring “good”

Metric	General Consideration	General Threshold Range
Recall	Higher Recall is desirable, but might be unrealistic depending on data quality and task difficulty.	0.7 - 0.8
MRR (Mean Reciprocal Rank)	A high MRR indicates the top retrieved documents are highly relevant to the query.	0.5 - 0.7
MAP (Mean Average Precision)	A high MAP suggests consistently good relevance across retrieved documents, not just the top ones.	0.4 - 0.6
Context Recall	Higher Context Recall is desirable, especially for tasks requiring very specific information.	0.6 - 0.8
Context Precision	A high Context Precision indicates retrieved information directly aids the LLM.	0.7 - 0.8

Choosing the right tool

LLAMA Index: Useful for managing private data used in a RAG system's retrieval process, but you'll still need separate tools for retrieval and evaluation.

Promptfoo: An indirect tool for exploration and refinement during RAG system development, not a dedicated evaluation tool.

RAGAS: A good choice when you need a simple and efficient way to evaluate the retrieval component of an existing RAG system.

RAGTriad: Expands on RAGAS by offering human evaluation and visualization tools for a more comprehensive assessment.

LangChain Evaluator: More suitable if you're building a custom RAG system from scratch within a LangChain framework and want to integrate retrieval evaluation within your pipeline.

Generation Metrics

These metrics assess the quality of the outputs generated by the LLM after being augmented with retrieved information.

- **Accuracy**: This measures how well the LLM's response aligns with the factual truth, especially important for tasks like question answering.
- **Fluency and Coherence**: These metrics assess the readability and logical flow of the generated text. The LLM's response should be grammatically correct and easy to understand.
- **Relevance**: This measures how well the LLM's response addresses the user's query and stays on topic.
- **Informativeness**: This metric assesses how much useful information the LLM's response conveys to the user.
- **Engagement**: Depending on the context, the response might be evaluated on its ability to be interesting, creative, or capture the user's attention.
- **Safety and Fairness**: These metrics assess if the generated text is free from harmful biases or offensive content.

Generation Evaluators

1. N-Gram Based Metrics:

These metrics focus on how well the generated text matches existing text data based on the overlap of n-grams (sequences of n words).

- **BLEU**: Compares the generated text to reference sentences, considering n-gram precision.
- **ROUGE**: Similar to BLEU, it focuses on n-gram recall and considers different types of n-gram matches.
- **METEOR**: This metric combines features from BLEU and ROUGE with additional factors like synonym matching, making it potentially more robust.

2. Similarity-Based Metrics:

These metrics leverage similarity measures to assess the quality and coherence of the generated text.

- **BERTScore**: Compares the generated text to a reference using pre-trained BERT models, considering both similarity and fluency.
- **SemScore (Semantic Similarity Score):** Measures semantic similarity between generated text and a reference using pre-trained language models
- **MoverScore (Mover's Distance Score):** Measures how much the generated text "moves" semantically from the reference.
- **Word Perplexity:** Measures how well a language model predicts the next word in a sequence.
- **Perplexity Reduction:** Measures the decrease in perplexity of the LLM's outputs when conditioned on retrieved information.

Generation Evaluators

3. LLM-Based Metrics:

These metrics use other LLMs to evaluate the generation quality and identify potential hallucinations.

- **G-eval**: Scores the generated text based on its coherence, fluency, and factual consistency as judged by another LLM.
- **UniEval**: This metric considers multiple factors like fluency, grammaticality, and factual coherence through an ensemble of LLM evaluators.
- **GPTScore**: Designed specifically for GPT-like models, it evaluates aspects like coherence, safety, and factual consistency using an LLM.
- **TRUE**: This metric uses other LLMs to assess factual correctness and identify potential factual hallucinations.
- **SelfCheckGPT**: Designed for GPT models, it focuses on identifying logical inconsistencies and factual errors in the generated text.
- **ChatProtect**: This metric aims to identify harmful or unsafe content generated by the LLM through interaction with another LLM.
- **Chainpoll**: Evaluates factual correctness by comparing the generated text to multiple retrieved documents and assessing consistency.

Measuring “good”

N-gram Based Metrics	**BLEU (Bi-Lingual Evaluation Understudy)**	**0.8+** (Generally considered good for high-quality generation)
	ROUGE (Recall-Oriented Understudy for Gisting Evaluation)	**0.6+** (Generally considered good for high-quality generation)
	METEOR (Metric for Evaluation of Translation with Ordering)	Typically **0.2+** (Higher is better)
Model-Based Metrics	**BERTScore (BERT-based Evaluation Score)**	**0.8+** (Generally considered good for high-quality generation)
	SemScore, MoverScore	Threshold depends on specific task and desired quality level
	Word Perplexity	Lower is better (Suggests the model can predict words accurately)
	Perplexity Reduction	Higher reduction indicates retrieved information improves LLM's prediction accuracy.
LLM-Based Metrics	**G-eval, UniEval, TRUE (Text REtrieval for Unbiased Evaluation), GPTScore, SelfCheckGPT, ChatProtect, Chainpoll**	Threshold depends on specific task and desired quality level.

#2.
Reasoning Models

Some of the top reasoning models

OpenAI's o1 Model: In September 2024, OpenAI introduced the o1 model, designed to tackle complex problems by simulating human-like reasoning

DeepSeek's R1: In January 2025, DeepSeek released the R1 model family under an open MIT license, with the largest version containing 671 billion parameters. T

Alibaba's Qwen Series: Alibaba's AI models, known as Qwen, have been developed to compete with leading AI models globally. In January 2025, Alibaba launched Qwen 2.5-Max, which reportedly outperforms other foundational models, including GPT-4o and DeepSeek-V3, in key benchmarks.

Baidu's Ernie X1: Baidu introduced Ernie X1, claiming it offers capabilities akin to DeepSeek's R1 but at half the cost. Ernie X1 can handle tasks such as AI image generation, code interpretation, web page reading, and complex calculations.

The Key Differences

Feature	Reasoning Models (e.g., OpenAI o1, DeepSeek R1)	RAG-Based LLMs (e.g., GPT-4 RAG, Claude + RAG)
Core Mechanism	Internal reasoning, **multi-step problem-solving**, self-reflection	External retrieval, augmenting LLM responses with external sources
Data Source	Uses **parametric memory** (knowledge encoded within the model during training)	Uses **non-parametric memory** (retrieves fresh documents from a database or search)
Strengths	Good at complex reasoning (math, logic, code generation, theorem proving)	Good at f**actual accuracy,** real-time updates, knowledge-intensive queries
Weaknesses	Can **hallucinate** facts since it relies only on its trained knowledge	Struggles with deep logical chains, limited by retrieval quality

Reasoning models are better for autonomous, deep problem-solving e.g., AI agents whereas **RAG-based LLMs** are better for handling **dynamic, real-world factual information**.

So naturally, their evals are different too

RAG-Based LLM Models

Knowledge Recall & Fact-Checking (TruthfulQA, FEVER) → Measures how accurately retrieved data is incorporated.

Retrieval Accuracy (MRR, Top-k Precision) → Assesses whether the model finds relevant documents.

Response Coherence with External Context → Ensures that the model correctly interprets retrieved information.

Hallucination Rate → Tests whether the model invents information beyond retrieved sources.

Reasoning-Based LLM Models

Coding Benchmarks (Codeforces, HumanEval) → Measures step-by-step logical execution.

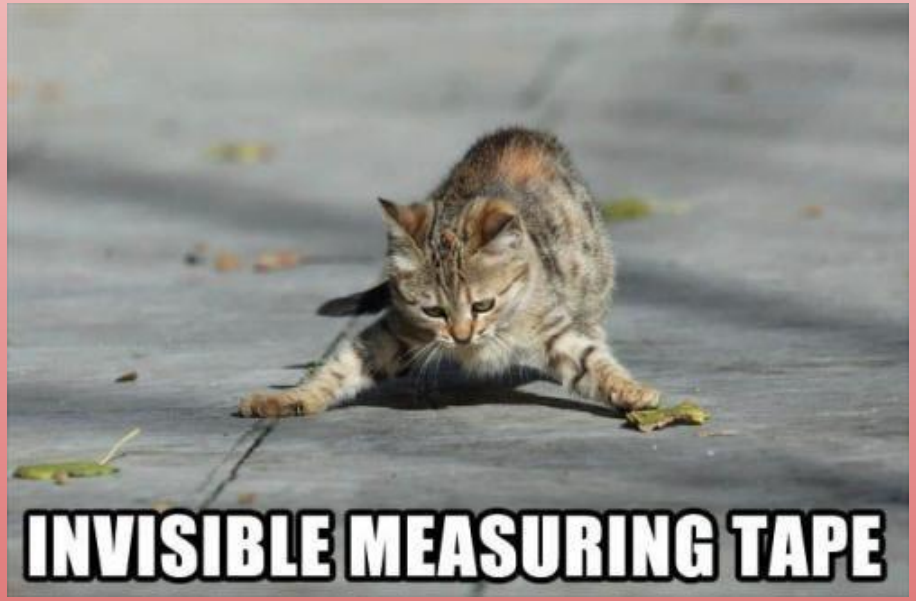
Mathematical Reasoning (MATH, GSM8K, IMO Qualifier) → Tests multi-step deduction.

Scientific Problem-Solving (ARC, AI2 Reasoning) → Assesses logical consistency.

Multi-Step Chain of Thought (CoT) → Checks if the model can self-correct mistakes.

But it presents its own set of problems (currently unsolved)

1. Benchmark leakage
2. Most benchmarks test performance on static datasets, meaning models can perform well on known problems but fail in novel scenarios. So, we don’t have a established methodology to evaluate **out-of-distribution generalization for reasoning**.
3. Lack of Multi-Step, Interactive Evaluations
4. Failure to Measure Causal vs. Correlational Reasoning

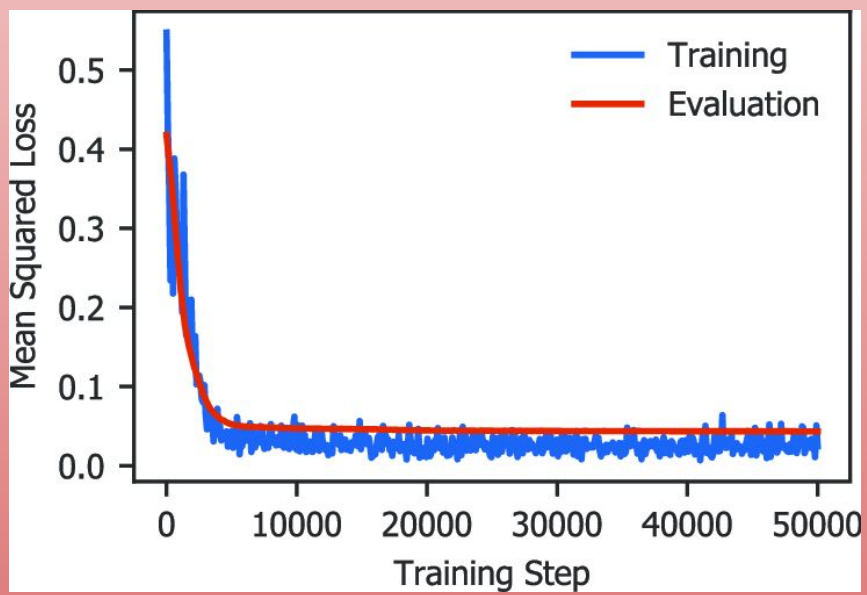


#4.
Open Challenges

More LLM Evaluation Criteria..

- **Training Loss**: While not directly an evaluation metric, training loss (eval/loss) indicates how well the model is learning during training.

1. **Knowledge and Capacity Evaluation**
2. **Domain Specialization Evaluation**
3. Alignment Evaluation
4. Safety Evaluation



Metrics – Challenges of Static Benchmarks for LLM Evaluation

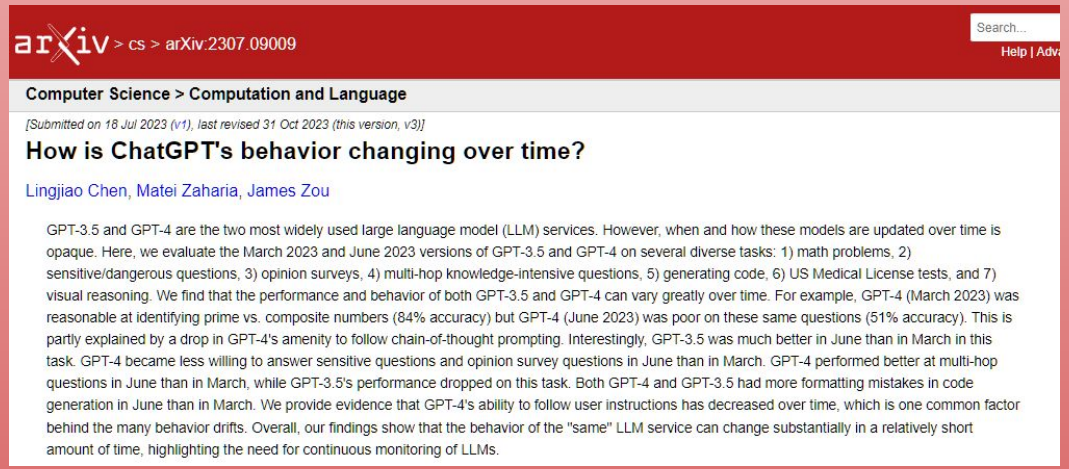
Benchmark	Description	Focus
GLUE (General Language Understanding Evaluation)	Suite of tasks assessing core NLP abilities	Natural Language Understanding (NLU)
SuperGLUE	Successor to GLUE, featuring more challenging tasks	Natural Language Understanding (NLU)
HellaSwag	Focuses on reasoning and commonsense understanding	Natural Language Inference (NLI)
TruthfulQA	Evaluates factual correctness and avoids factual hallucinations	Question Answering (QA)
MMLU (Massive Multitask Language Understanding)	Large-scale benchmark with diverse tasks	Multi-task Learning

- **Data Leakage:** Static datasets can be memorized by LLMs, inflating their performance. Dynamic evaluation with frequently updated data can prevent this.
- **Limited Task Scope:** Static benchmarks often focus on multiple-choice questions, neglecting open-ended tasks. Dynamic evaluation could consider debates between LLMs for open-ended tasks.
- **Outdated Knowledge:** Static benchmarks test on static knowledge, while real-world information changes. Dynamic evaluation should consider evolving factual data.
- **Limited Difficulty:** As LLMs improve, static benchmarks become outdated. Dynamic benchmarks with increasing difficulty are needed.

Open Challenges

1. Prompt Sensitivity

LLMs are highly sensitive to the prompts used to guide their generation. A seemingly minor update in model can lead to drastically different outputs. This makes it difficult to design prompts that consistently elicit the desired response and assess the LLM's true capabilities.



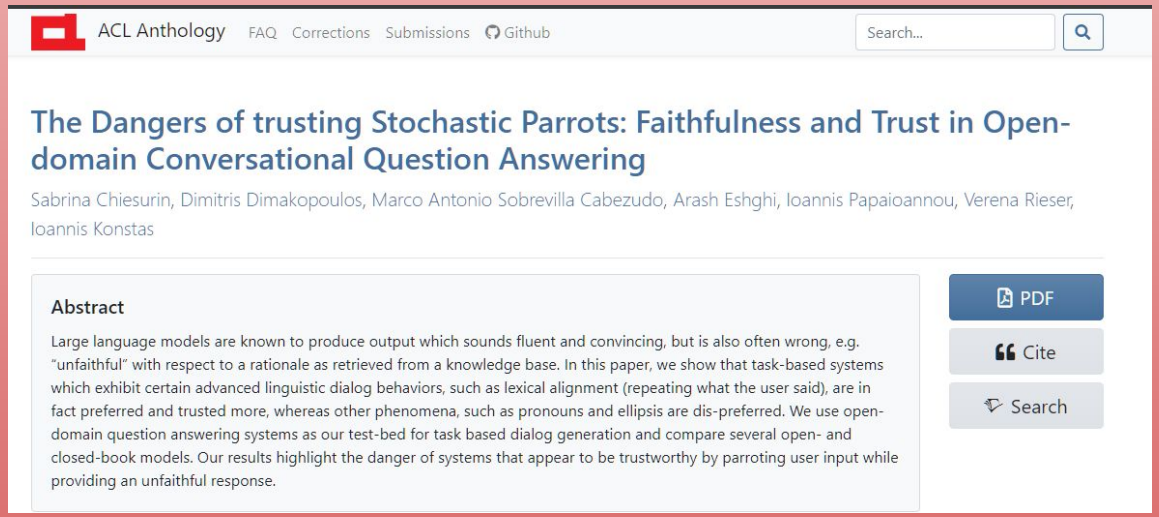
Observations:

- a decrease in verbosity between March and June 2023
- a decrease in GPT-4's willingness to answer sensitive questions
- dramatic decrease in accuracy for GPT-4 between March and June 2023, highlighting the model's behavioral shift.

Open Challenges

2. Construct Validity

Are we measuring the qualities or capabilities of LLMs that we truly care about?



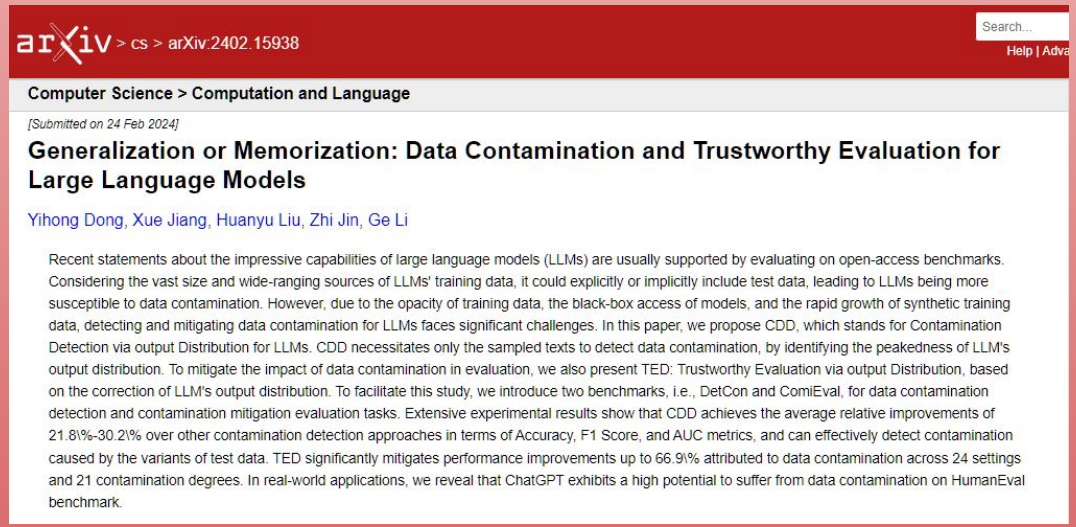
Observations:

- LLMs, despite their fluency, lack genuine semantic understanding
- LLMs can produce fluent and grammatically correct text that is factually incorrect or nonsensical.
- Evaluation metrics that go beyond simple benchmark scores, and take into account factors like bias, fairness, and robustness are needed.

Open Challenges

3. Contamination

LLMs are trained on massive amounts of data, which can harbor biases and factual inaccuracies. These biases can be reflected in the LLM's outputs.



arXiv > cs > arXiv:2402.15938

Search...

Help | Adva

Computer Science > Computation and Language

[Submitted on 24 Feb 2024]

Generalization or Memorization: Data Contamination and Trustworthy Evaluation for Large Language Models

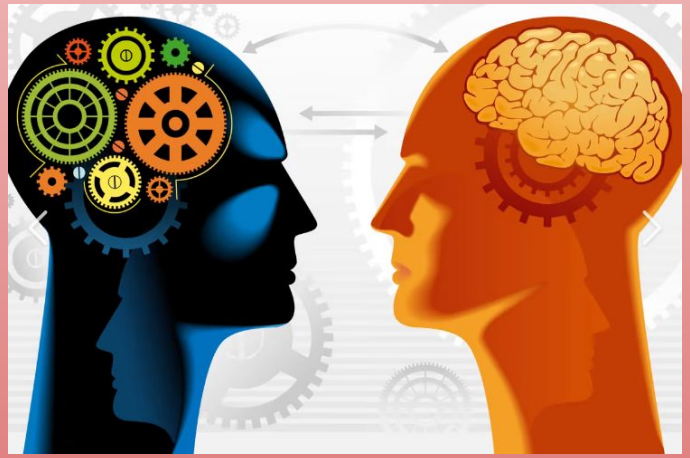
Yihong Dong, Xue Jiang, Huanyu Liu, Zhi Jin, Ge Li

Recent statements about the impressive capabilities of large language models (LLMs) are usually supported by evaluating on open-access benchmarks. Considering the vast size and wide-ranging sources of LLMs' training data, it could explicitly or implicitly include test data, leading to LLMs being more susceptible to data contamination. However, due to the opacity of training data, the black-box access of models, and the rapid growth of synthetic training data, detecting and mitigating data contamination for LLMs faces significant challenges. In this paper, we propose CDD, which stands for Contamination Detection via output Distribution for LLMs. CDD necessitates only the sampled texts to detect data contamination, by identifying the peakedness of LLM's output distribution. To mitigate the impact of data contamination in evaluation, we also present TED: Trustworthy Evaluation via output Distribution, based on the correction of LLM's output distribution. To facilitate this study, we introduce two benchmarks, i.e., DetCon and ComiEval, for data contamination detection and contamination mitigation evaluation tasks. Extensive experimental results show that CDD achieves the average relative improvements of 21.8\%-30.2\% over other contamination detection approaches in terms of Accuracy, F1 Score, and AUC metrics, and can effectively detect contamination caused by the variants of test data. TED significantly mitigates performance improvements up to 66.9\% attributed to data contamination across 24 settings and 21 contamination degrees. In real-world applications, we reveal that ChatGPT exhibits a high potential to suffer from data contamination on HumanEval benchmark.

Observations:

- LLMs can appear smart by mimicking language patterns (stochastic parrots) but may not truly grasp the meaning.
- **N-gram metrics miss the point:** Focusing on matching words (n-grams) overlooks the actual content and factual accuracy of the generated text.
- **Reasoning evaluation is hard:** Current methods might not effectively assess how well LLMs reason through complex problems.

Final Recommendations!



1. **Instructional Scaffolding:** Break complex prompts into smaller, more manageable steps via prompt chaining. This helps the LLM focus on specific aspects of the task and reduces the opportunity for drift as well as help reduce sensitivity to specific wordings in the prompt.

2. **Semantic Similarity Metrics:** Explore metrics that capture the semantic meaning of generated text, like MoverScore or Sentence Transformers. These metrics can assess the LLM's ability to understand and convey the core concepts of the task.

3. **Data Cleaning and Filtering:** Before training, implement data cleaning techniques to reduce biases and factual errors within the training data. This helps to minimize the risk of contamination influencing the LLM's outputs.

Thank you! *Time for Q & A?*

For collaboration, you can reach out to me at

abi@abiaryan.com

Socials: @goabiaryan

(LinkedIn, Twitter, Threads, Twitter)