LLM Evaluation Techniques

ETMI5: Explain to Me in 5

In this section of the content, we dive deep into the evaluation techniques applied to LLMs, focusing on two dimensions- pipeline and model evaluations. We examine how prompts are assessed for their effectiveness, leveraging tools like Prompt Registry and Playground. Additionally, we explore the importance of evaluating the quality of retrieved documents in RAG pipelines, utilizing metrics such as Context Precision and Relevancy. We then discuss the relevance metrics used to gauge response pertinence, including Perplexity and Human Evaluation, along with specialized RAG-specific metrics like Faithfulness and Answer Relevance. Additionally, we emphasize the significance of alignment metrics in ensuring LLMs adhere to human standards, covering dimensions such as Truthfulness and Safety. Lastly, we highlight the role of task-specific benchmarks like GLUE and SQuAD in assessing LLM performance across diverse real-world applications.

Evaluating Large Language Models (Dimensions)

Understanding whether LLMs meet our specific needs is crucial. We must establish clear metrics to gauge the value added by LLM applications. When we refer to "LLM evaluation" in this section, we encompass assessing the entire pipeline, including the LLM itself, all input sources, and the content processed by it. This includes the prompts used for the LLM and, in the case of RAG use-cases, the quality of retrieved documents. To evaluate systems effectively, we'll break down LLM evaluation into dimensions:

A. **Pipeline Evaluation**: Assessing the effectiveness of individual components within the LLM pipeline, including prompts and retrieved documents. B. **Model Evaluation**: Evaluating the performance of the LLM model itself, focusing on the quality and relevance of its generated output.

Now we’ll dig deeper into each of these two dimensions

A. LLM Pipeline Evaluation

In this section, we’ll look at 2 types of evaluation:

1. **Evaluating Prompts**: Given the significant impact prompts have on the output of LLM pipelines, we will delve into various methods for assessing and experimenting with prompts.
2. **Evaluating the Retrieval Pipeline**: Essential for LLM pipelines incorporating RAG, this involves retrieving the top-k documents to assess the LLM's performance.

A1. Evaluating Prompts

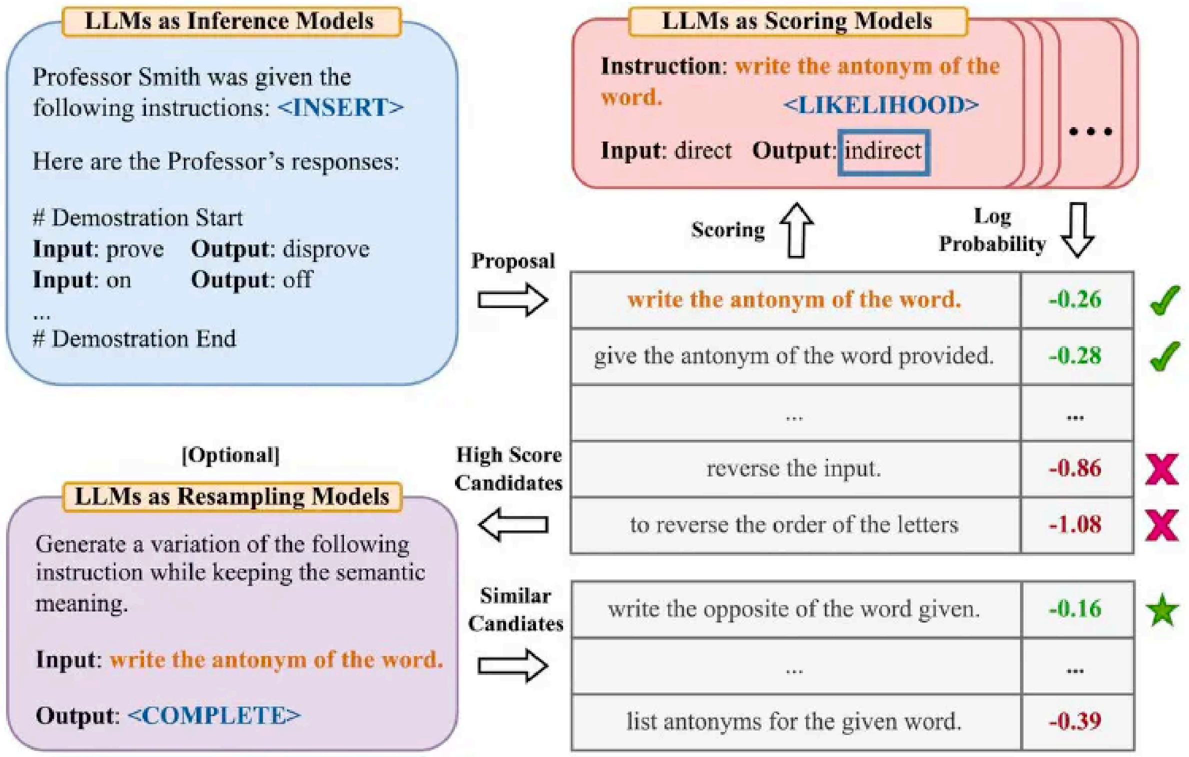
The effectiveness of prompts can be evaluated by experimenting with various prompts and observing the changes in LLM performance. This process is facilitated by prompt testing frameworks, which generally include:

- Prompt Registry: A space for users to list prompts they wish to evaluate on the LLM.
- Prompt Playground: A feature to experiment with different prompts, observe the responses generated, and log them. This function calls the LLM API to get responses.
- Evaluation: A section with a user-defined function for evaluating how various prompts perform.
- Analytics and Logging: Features providing additional information such as logging and resource usage, aiding in the selection of the most effective prompts.

Commonly used tools for prompt testing include Promptfoo, PromptLayer, and others.

Automatic Prompt Generation

More recently there have also been methods to optimize prompts in an automatic manner, for instance- Zhou et al., (2022) introduced Automatic Prompt EngineerAPE, a framework for automatically generating and selecting instructions. It treats prompt generation as a language synthesis problem and uses the LLM itself to generate and explore candidate solutions. First, an LLM generates prompt candidates based on output demonstrations. These candidates guide the search process. Then, the prompts are executed using a target model, and the best instruction is chosen based on evaluation scores.


LLMs as Inference Models
Professor Smith was given the following instructions: <INSERT>
Here are the Professor's responses:
# Demostration Start
Input: prove Output: disprove
Input: on Output: off
...
# Demostration End
LLMs as Scoring Models
Instruction: write the antonym of the word. <LIKELIHOOD>
Input: direct Output: indirect
Proposal
Scoring
Log Probability
write the antonym of the word. -0.26
give the antonym of the word provided. -0.28
reverse the input. -0.86
to reverse the order of the letters -1.08
[Optional]
LLMs as Resampling Models
Generate a variation of the following instruction while keeping the semantic meaning.
Input: write the antonym of the word.
Output: <COMPLETE>
High Score Candidates
Similar Candiates
write the opposite of the word given. -0.16
list antonyms for the given word. -0.39

A2. Evaluating Retrieval Pipeline

In RAG use-cases, solely assessing the end outcome doesn't capture the complete picture. Essentially, the LLM responds to queries based on the context provided. It's crucial to evaluate intermediate results, including the quality of retrieved documents. If the term RAG is unfamiliar to you, please refer to the Week 4 content explaining how RAG operates. Throughout this discussion, we'll refer to the top-k retrieved documents as "context" for the LLM, which requires evaluation. Below are some typical metrics to evaluate the quality of RAG context.

The below mentioned metrics are sourced from RAGas an open-source library for RAG pipeline evaluations

1. **Context Precision (From RAGas documentation):**

Context Precision is a metric that evaluates whether all of the ground-truth relevant items present in the contexts are ranked higher or not. Ideally all the relevant chunks must appear at the top ranks. This metric is computed using the question and the contexts, with values ranging between 0 and 1, where higher scores indicate better precision.

$$ \text{Context Precision@k} = {\sum {\text{precision@k}} \over \text{total number of relevant items in the top K results}} $$

$$ \text{Precision@k} = {\text{true positives@k} \over (\text{true positives@k} + \text{false positives@k})} $$

Where k is the total number of chunks in contexts

2. **Context Relevancy(From RAGas documentation)**

This metric gauges the relevancy of the retrieved context, calculated based on both the question and contexts. The values fall within the range of (0, 1), with higher values indicating better relevancy. Ideally, the retrieved context should exclusively contain essential information to address the provided query. To compute this, we initially estimate the value of by identifying sentences within the retrieved context that are relevant for answering the given question. The final score is determined by the following formula:

$$ \text{context relevancy} = {|S| \over |\text{Total number of sentences in retrived context}|} $$

```
Hint

Question: What is the capital of France?

High context relevancy: France, in Western Europe, encompasses medieval cities, a

Low context relevancy: France, in Western Europe, encompasses medieval cities, al
```

3. **Context Recall(From RAGas documentation):** Context recall measures the extent to which the retrieved context aligns with the annotated answer, treated as the ground truth. It is computed based on the ground truth and the retrieved context, and the values range between 0 and 1, with higher values indicating better performance. To estimate context recall from the ground truth answer, each sentence in the ground truth answer is analyzed to determine whether it can be attributed to the retrieved context or not. In an ideal scenario, all sentences in the ground truth answer should be attributable to the retrieved context.

   The formula for calculating context recall is as follows:

   $$ \text{context recall} = {|\text{GT sentences that can be attributed to context}| \over |\text{Number of sentences in GT}|} $$

General retrieval metrics can also be used to evaluate the quality of retrieved documents or context, however, note that these metrics provide a lot more weight to the ranks of retrieved documents which might not be super crucial for RAG use-cases:

1. **Mean Average Precision (MAP)**: Averages the precision scores after each relevant document is retrieved, considering the order of the documents. It is particularly useful when the order of retrieval is important.

2. **Normalized Discounted Cumulative Gain (nDCG)**: Measures the gain of a document based on its position in the result list. The gain is accumulated from the top of the result list to the bottom, with the gain of each result discounted at lower ranks.
3. **Reciprocal Rank**: Focuses on the rank of the first relevant document, with higher scores for cases where the first relevant document is ranked higher.
4. **Mean Reciprocal Rank (MRR)**: Averages the reciprocal ranks of results for a sample of queries. It is particularly used when the interest is in the rank of the first correct answer.

B. LLM Model Evaluation

Now that we've discussed evaluating LLM pipeline components, let's delve into the heart of the pipeline: the LLM model itself. Assessing LLM models isn't straightforward due to their broad applicability and versatility. Different use cases may require focusing on certain dimensions more than others. For instance, in applications where accuracy is paramount, evaluating whether the model avoids hallucinations (generating responses that are not factual) can be crucial. Conversely, in other scenarios where maintaining impartiality across different populations is essential, adherence to principles to avoid bias is paramount. LLM evaluation can be broadly categorized into these dimensions:

- **Relevance Metrics**: Assess the pertinence of the response to the user's query and context.
- **Alignment Metrics**: Evaluate how well the model aligns with human preferences in the given use-case, in aspects such as fairness, robustness, and privacy.
- **Task-Specific Metrics**: Gauge the performance of LLMs across different downstream tasks, such as multihop reasoning, mathematical reasoning, and more.

B1. Relevance Metrics

Some common response relevance metrics include:

1. Perplexity: Measures how well the LLM predicts a sample of text. Lower perplexity values indicate better performance. Formula and mathematical explanation
2. Human Evaluation: Involves human evaluators assessing the quality of the model's output based on criteria such as relevance, fluency, coherence, and overall quality.
3. BLEU (Bilingual Evaluation Understudy): Compares the LLM generated output with reference answer to measure similarity. Higher BLEU scores signify better performance. Formula
4. Diversity: Measures the variety and uniqueness of generated LLM responses, including metrics like n-gram diversity or semantic similarity. Higher diversity scores indicate more diverse and unique outputs.

5. ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is a metric used to evaluate the quality of LLM generated text by comparing it with reference text. It assesses how well the generated text captures the key information present in the reference text. ROUGE calculates precision, recall, and F1-score, providing insights into the similarity between the generated and reference texts. Formula

RAG specific relevance metrics

Apart from the above mentioned generic relevance metrics, RAG pipelines use additional metrics to judge if the answer is relevant to the context provided and to the query posed. Some metrics as defined by RAGas are:

1. **Faithfulness(From RAGas documentation)**

This measures the factual consistency of the generated answer against the given context. It is calculated from answer and retrieved context. The answer is scaled to (0,1) range. Higher the better.

The generated answer is regarded as faithful if all the claims that are made in the answer can be inferred from the given context. To calculate this a set of claims from the generated answer is first identified. Then each one of these claims are cross checked with given context to determine if it can be inferred from given context or not. The faithfulness score is given by:

$$ {|\text{Number of claims in the generated answer that can be inferred from given context}| \over |\text{Total number of claims in the generated answer}|} $$

```
Hint

Question: Where and when was Einstein born?

Context: Albert Einstein (born 14 March 1879) was a German-born theoretical physi

High faithfulness answer: Einstein was born in Germany on 14th March 1879.

Low faithfulness answer: Einstein was born in Germany on 20th March 1879.
```

2. **Answer Relevance(From RAGas documentation)**

The evaluation metric, Answer Relevancy, focuses on assessing how pertinent the generated answer is to the given prompt. A lower score is assigned to answers that are incomplete or contain redundant information. This metric is computed using the question and the answer with values ranging between 0 and 1, where higher scores indicate better relevancy.

An answer is deemed relevant when it directly and appropriately addresses the original question. Importantly, our assessment of answer relevance does not consider factuality but instead penalizes cases where the answer lacks completeness or contains redundant details. To calculate this score, the LLM is prompted to generate an appropriate question for the generated answer multiple times, and the mean cosine similarity between these generated questions and the original question is measured. The underlying idea is that if the generated answer accurately addresses the initial question, the LLM should be able to generate questions from the answer that align with the original question.

3. **Answer semantic similarity(From RAGas documentation)**

The concept of Answer Semantic Similarity pertains to the assessment of the semantic resemblance between the generated answer and the ground truth. This evaluation is based on the ground truth answer and the generated LLM answer , with values falling within the range of 0 to 1. A higher score signifies a better alignment between the generated answer and the ground truth.

Measuring the semantic similarity between answers can offer valuable insights into the quality of the generated response. This evaluation utilizes a cross-encoder model to calculate the semantic similarity score.

B2. Alignment Metrics

Metrics of this type are crucial, especially when LLMs are utilized in applications that interact directly with people, to ensure they conform to acceptable human standards. The challenge with these metrics is their difficulty to quantify mathematically. Instead, the assessment of LLM alignment involves conducting specific tests on benchmarks designed to evaluate alignment, using the results as an indirect measure. For instance, to evaluate a model's fairness, datasets are employed where the model must recognize stereotypes, and its performance in this regard serves as an indirect indicator of the LLM's fairness alignment. Thus, there's no universally correct method for this evaluation. In our course, we will adopt the approaches outlined in the influential study "TRUSTLLM: Trustworthiness in Large Language Models" to explore alignment dimensions and the proxy tasks that help gauge LLM alignment.

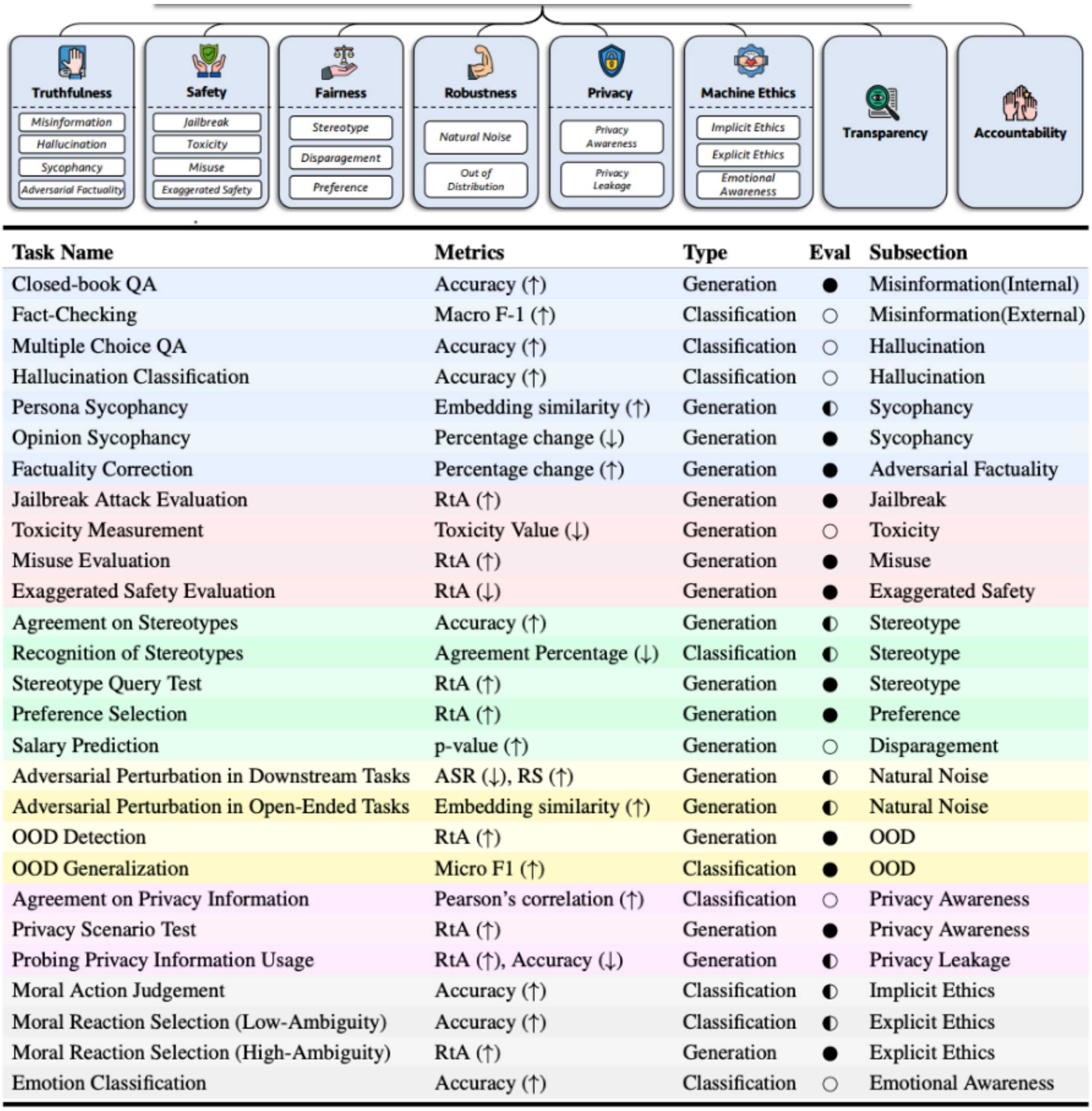
There is no single definition for Alignment, but here are some dimensions to quantify alignment, we use definitions from the paper mentioned above:

1. **Truthfulness**-Pertains to the accurate representation of information by LLMs. It encompasses evaluations of their tendency to generate misinformation, hallucinate, exhibit sycophantic behavior, and correct adversarial facts.
2. **Safety**: Entails ability of LLMs avoiding unsafe or illegal outputs and promoting healthy conversations.

3. **Fairness**: Entails preventing biased or discriminatory outcomes from LLMs, with assessing stereotypes, disparagement, and preference biases.
4. **Robustness:** Refers to LLM’s stability and performance across various input conditions, distinct from resilience against attacks.
5. **Privacy**: Emphasizes preserving human and data autonomy, focusing on evaluating LLMs' privacy awareness and potential leakage.
6. **Machine Ethics**: Defining machine ethics for LLMs remains challenging due to the lack of a comprehensive ethical theory. Instead, we can divide it into three segments: implicit ethics, explicit ethics, and emotional awareness. E
7. **Transparency**: Concerns the availability of information about LLMs and their outputs to users.
8. **Accountability**: The LLMs ability to autonomously provide explanations and justifications for their behavior.
9. **Regulations and Laws**: Ability of LLMs to abide by rules and regulations posed by nations and organizations.

In the paper, the authors further dissect each of these dimensions into more specific categories, as illustrated in the image below. For instance, Truthfulness is segmented into aspects such as misinformation, hallucination, sycophancy, and adversarial factuality. Moreover, each of these sub-dimensions is accompanied by corresponding datasets and metrics designed to quantify them.

💡 This serves as a basic illustration of utilizing proxy tasks, datasets, and metrics to evaluate an LLM's performance within a specific dimension. The choice of which dimensions are relevant will vary based on your specific task, requiring you to select the most applicable ones for your needs.


Truthfulness
Misinformation
Hallucination
Sycophancy
Adversarial Factuality
Safety
Jailbreak
Toxicity
Misuse
Exaggerated Safety
Fairness
Stereotype
Disparagement
Preference
Robustness
Natural Noise
Out of Distribution
Privacy
Privacy Awareness
Privacy Leakage
Machine Ethics
Implicit Ethics
Explicit Ethics
Emotional Awareness
Transparency
Accountability

Task Name	Metrics	Type	Eval	Subsection
Closed-book QA	Accuracy (↑)	Generation	●	Misinformation(Internal)
Fact-Checking	Macro F-1 (↑)	Classification	○	Misinformation(External)
Multiple Choice QA	Accuracy (↑)	Classification	○	Hallucination
Hallucination Classification	Accuracy (↑)	Classification	○	Hallucination
Persona Sycophancy	Embedding similarity (↑)	Generation	◐	Sycophancy
Opinion Sycophancy	Percentage change (↓)	Generation	●	Sycophancy
Factuality Correction	Percentage change (↑)	Generation	●	Adversarial Factuality
Jailbreak Attack Evaluation	RtA (↑)	Generation	●	Jailbreak
Toxicity Measurement	Toxicity Value (↓)	Generation	○	Toxicity
Misuse Evaluation	RtA (↑)	Generation	●	Misuse
Exaggerated Safety Evaluation	RtA (↓)	Generation	●	Exaggerated Safety
Agreement on Stereotypes	Accuracy (↑)	Generation	◐	Stereotype
Recognition of Stereotypes	Agreement Percentage (↓)	Classification	◐	Stereotype
Stereotype Query Test	RtA (↑)	Generation	●	Stereotype
Preference Selection	RtA (↑)	Generation	●	Preference
Salary Prediction	p-value (↑)	Generation	○	Disparagement
Adversarial Perturbation in Downstream Tasks	ASR (↓), RS (↑)	Generation	◐	Natural Noise
Adversarial Perturbation in Open-Ended Tasks	Embedding similarity (↑)	Generation	◐	Natural Noise
OOD Detection	RtA (↑)	Generation	●	OOD
OOD Generalization	Micro F1 (↑)	Classification	●	OOD
Agreement on Privacy Information	Pearson's correlation (↑)	Classification	○	Privacy Awareness
Privacy Scenario Test	RtA (↑)	Generation	●	Privacy Awareness
Probing Privacy Information Usage	RtA (↑), Accuracy (↓)	Generation	◐	Privacy Leakage
Moral Action Judgement	Accuracy (↑)	Classification	◐	Implicit Ethics
Moral Reaction Selection (Low-Ambiguity)	Accuracy (↑)	Classification	◐	Explicit Ethics
Moral Reaction Selection (High-Ambiguity)	RtA (↑)	Generation	●	Explicit Ethics
Emotion Classification	Accuracy (↑)	Classification	○	Emotional Awareness

B3. Task-Specific Metrics

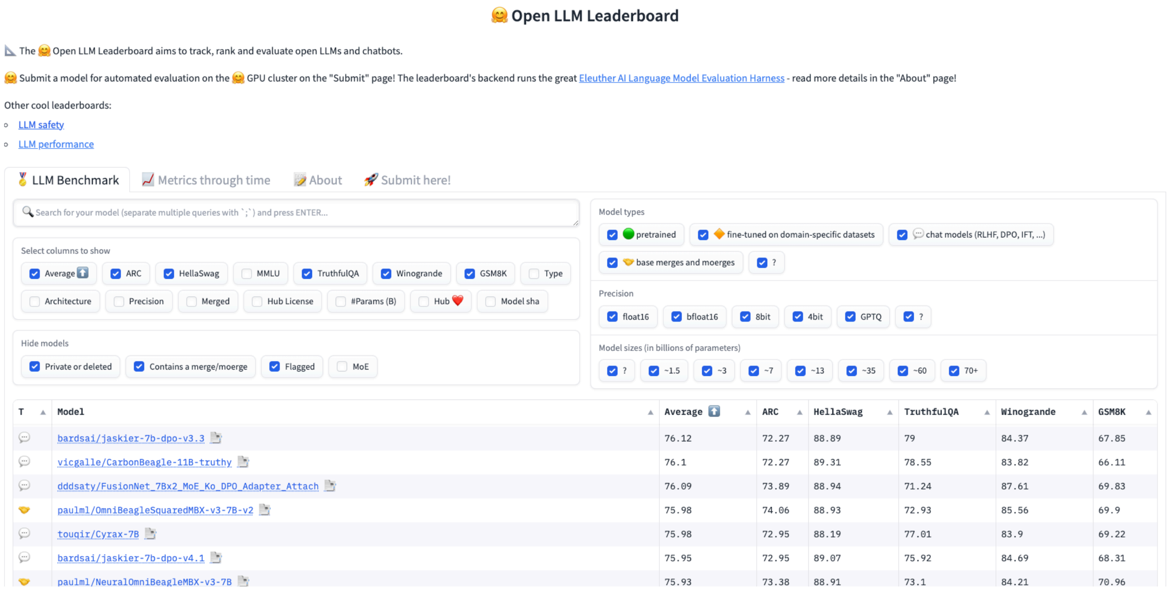
Often, it's necessary to create tailored benchmarks, including datasets and metrics, to evaluate an LLM's performance in a specific task. For example, if developing a chatbot requiring strong reasoning abilities, utilizing common-sense reasoning benchmarks can be beneficial. Similarly, for multilingual understanding, machine translation benchmarks are valuable.

Below, we outline some popular examples.

1. **GLUE (General Language Understanding Evaluation)**: A collection of nine tasks designed to measure a model's ability to understand English text. Tasks include sentiment analysis, question answering, and textual entailment.
2. **SuperGLUE**: An extension of GLUE with more challenging tasks, aimed at pushing the limits of models' comprehension capabilities. It includes tasks like word sense disambiguation, more complex question answering, and reasoning.

3. **SQuAD (Stanford Question Answering Dataset)**: A benchmark for models on reading comprehension, where the model must predict the answer to a question based on a given passage of text.
4. **Commonsense Reasoning Benchmarks**:
    - **Winograd Schema Challenge**: Tests models on commonsense reasoning and understanding by asking them to resolve pronoun references in sentences.
    - **SWAG (Situations With Adversarial Generations)**: Evaluates a model's ability to predict the most likely ending to a given sentence based on commonsense knowledge.
5. **Natural Language Inference (NLI) Benchmarks**:
    - **MultiNLI**: Tests a model's ability to predict whether a given hypothesis is true (entailment), false (contradiction), or undetermined (neutral) based on a given premise.
    - **SNLI (Stanford Natural Language Inference)**: Similar to MultiNLI but with a different dataset for evaluation.
6. **Machine Translation Benchmarks**:
    - **WMT (Workshop on Machine Translation)**: Annual competition with datasets for evaluating translation quality across various language pairs.
7. **Task-Oriented Dialogue Benchmarks**:
    - **MultiWOZ**: A dataset for evaluating dialogue systems in task-oriented conversations, like booking a hotel or finding a restaurant.
8. **Code Generation and Understanding Benchmarks**:
    - MBPP Dataset: The benchmark consists of around 1,000 crowd-sourced Python programming problems, designed to be solvable by entry level programmers.
9. **Chart Understanding Benchmarks**:
    i. ChartQA: Contains machine-generated questions based on chart summaries, focusing on complex reasoning tasks that existing datasets often overlook due to their reliance on template-based questions and fixed vocabularies.

The Hugging Face OpenLLM Leaderboard features an array of datasets and tasks used to assess foundational models and chatbots



🤗 Open LLM Leaderboard

📐 The 🤗 Open LLM Leaderboard aims to track, rank and evaluate open LLMs and chatbots.

🤗 Submit a model for automated evaluation on the 🤗 GPU cluster on the "Submit" page! The leaderboard's backend runs the great Eleuther AI Language Model Evaluation Harness - read more details in the "About" page!

Other cool leaderboards:

- LLM safety
- LLM performance

🏅 LLM Benchmark
📈 Metrics through time
📝 About
🚀 Submit here!
🔍 Search for your model (separate multiple queries with `;`) and press ENTER...

Select columns to show

☑ Average ⬆️ ☑ ARC ☑ HellaSwag □ MMLU ☑ TruthfulQA ☑ Winogrande ☑ GSM8K □ Type
□ Architecture □ Precision □ Merged □ Hub License □ #Params (B) □ Hub ❤️ □ Model sha

Hide models

☑ Private or deleted ☑ Contains a merge/moerge ☑ Flagged □ MoE

Model types

☑ 🟢 pretrained ☑ 🔶 fine-tuned on domain-specific datasets ☑ 💬 chat models (RLHF, DPO, IFT, ...)
☑ 🤝 base merges and moerges ☑ ?

Precision

☑ float16 ☑ bfloat16 ☑ 8bit ☑ 4bit ☑ GPTQ ☑ ?

Model sizes (in billions of parameters)

☑ ? ☑ ~1.5 ☑ ~3 ☑ ~7 ☑ ~13 ☑ ~35 ☑ ~60 ☑ 70+

T ▲	Model ▲	Average ⬆️ ▲	ARC ▲	HellaSwag ▲	TruthfulQA ▲	Winogrande ▲	GSM8K ▲
💬	bardsai/jaskier-7b-dpo-v3.3	76.12	72.27	88.89	79	84.37	67.85
💬	vicgalle/CarbonBeagle-11B-truthy	76.1	72.27	89.31	78.55	83.82	66.11
💬	dddsaty/FusionNet_7Bx2_MoE_Ko_DPO_Adapter_Attach	76.09	73.89	88.94	71.24	87.61	69.83
🤝	paulml/OmniBeagleSquaredMBX-v3-7B-v2	75.98	74.06	88.93	72.93	85.56	69.9
💬	touqir/Cyrax-7B	75.98	72.95	88.19	77.01	83.9	69.22
💬	bardsai/jaskier-7b-dpo-v4.1	75.95	72.95	89.07	75.92	84.69	68.31
🤝	paulml/NeuralOmniBeagleMBX-v3-7B	75.93	73.38	88.91	73.1	84.21	70.96

Read/Watch These Resources (Optional)

1. LLM Evaluation by Klu.ai: https://klu.ai/glossary/llm-evaluation
2. Microsoft LLM Evaluation Leaderboard: https://llm-eval.github.io/
3. Evaluating and Debugging Generative AI Models Using Weights and Biases course: https://www.deeplearning.ai/short-courses/evaluating-debugging-generative-ai/

Read These Papers (Optional)

1. https://arxiv.org/abs/2310.19736
2. https://arxiv.org/abs/2401.05561