LLM Hallucination Index

RAG SPECIAL

Brought to you by Galileo

A Ranking & Evaluation Framework For LLM Hallucinations

July 2024

Table of Contents

LLM Hallucination Index

RAG SPECIAL

Welcome to the second installation of Galileo's LLM Hallucination Index! The LLM landscape has changed a lot since launching our first Hallucination Index in November 2023, with larger more powerful open and closed-sourced models being announced monthly. Since then, two things happened: the term "hallucinate" became Dictionary.com's Word of the Year, and Retrieval-Augmented-Generation (RAG) has become the leading method for building AI solutions. And while the parameters and context lengths of these models continue to grow, the risk of hallucinations remains.

Galileo specializes in hallucination detection. Our platform for enterprise GenAI evaluation and observability helps AI teams across the product development lifecycle, from building and iterating to monitoring and protection. To conduct testing for the Index, we used our proprietary hallucination detection techniques and models, which we have detailed in this report. We hope you enjoy!

About the Index

Goal

Our new Index ranks 22 of the leading models based on their performance in real-world scenarios. We hope this index helps AI builders make informed decisions about which LLM is best suited for their particular use case and need.

Models Tested

We tested 22 models, 10 closed-source models and 12 open-source models, from leading foundation model brands like OpenAI, Anthropic, Meta, Google, Mistral, and more.

Attributes We Tested

There were two key LLM attributes we wanted to test as part of this Index - context length and open vs. closed-source.

Context Length

First and foremost, with the rising popularity of RAG, we wanted to see how context length affects model performance. Providing an LLM with context data is akin to giving a student a cheat sheet for an open-book exam. We tested three scenarios:

Short Context

Provide the LLM with < 5k tokens of context data, or the equivalent of a few pages of information.

Medium Context

Provide the LLM with 5k - 25k tokens of context data, or the equivalent of a book chapter.

Long Context

Provide the LLM with 40k - 100k tokens of context data, or the equivalent of an entire book.

Open vs. Closed Source

The open-source vs. closed-source software debate has waged on since the Free Software Movement (FSM) in the late 1980s. This debate has reached a fever pitch during the LLM Arms Race. The assumption is closed-source LLMs, with their access to proprietary training data, will perform better, but we wanted to put this assumption to the test.

How We Tested

Our approach when testing the LLMs followed a multi-step evaluation process:

Collect data

We gathered a diverse set of datasets reflecting real-world scenarios across short, medium, and long context lengths.

Conduct experiments

We ran each model against each dataset to gather results if it’s supported by model context length.

Evaluate model performance

To evaluate a model's propensity to hallucinate, we employed Context Adherence, Galileo's evaluation model which measures factual accuracy and an LLM's reasoning abilities within provided documents and context. Context Adherence is powered by the ChainPoll, a proprietary evaluation method developed by Galileo.

Analyze results

Once we had evaluated each model across each task type, our team got to work ranking models for specific categories.

To learn more about our methodology, see the Methodology section below.

Major Trends

In the course of testing for the Index, our team observed several key trends unfolding in the LLM Arms Race:

01 Open source is quickly closing the gap

While closed-source models still offer the best performance thanks to proprietary training data, open-source models like Gemini, Llama, and Qwen continue to improve in hallucination performance without the cost barriers of their close-sourced counterparts.

02 What context length?

We were surprised to find models perform particularly well with extended context lengths without losing quality or accuracy, reflecting how far model training and architecture has come.

03 Larger is not always better

In many tests, smaller models outperformed larger models. Gemini-1.5-flash-001 outperformed larger models across numerous tests, suggesting efficient model design can outweigh size in some cases. This theme is becoming more apparent, with Open AIs release of GPT-4o-mini (released 6/18/24) and Huggingface's SmolLM (released 7/17/24).

04 Anthropic outperforms OpenAI

During testing, Anthropic's latest Claude 3.5 Sonnet and Claude 3 Opus consistently scored close to perfect scores, beating out GPT-40 and GPT-3.5, especially in shorter context scenarios.

05 A worldwide surge in LLM development

LLMs from companies like Mistral and Alibaba have rapidly advanced in capability and popularity. This surge highlights the global effort to develop high-performing language models.

With new models being released almost weekly, it will be extremely interesting to see whether these trends hold through the end of 2024!

A Short Intro to RAG

Lets have a quick look on a basic RAG system before we go into the results. RAG works by dynamically retrieving relevant context from external sources, integrating it with user queries, and feeding the retrieval-augmented prompt to an LLM for generating responses.
To build the system, we must first set up the vector database with the external data by chunking the text, embedding the chunks, and loading them into the vector database. Once this is complete, we can orchestrate the following steps in real time to generate the answer for the user:

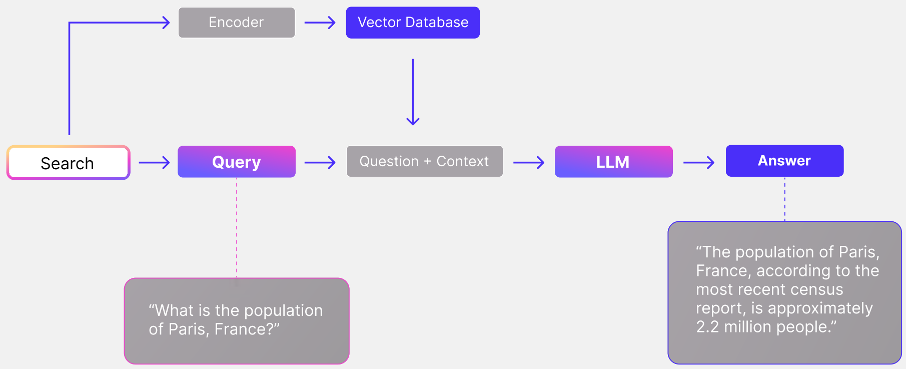
Retrieve Embedding the user query into the vector space to retrieve relevant context from an external knowledge source.

Augment Integrating the user query and the retrieved context into a prompt template.

Generate Feeding the retrieval-augmented prompt to the LLM for the final response generation.

An enterprise RAG system consists of dozens of components like storage, orchestration and observability. Each component is a large topic in itself which requires its own comprehensive blog.

How RAG Works


Encoder
Vector Database
Search
Query
Question + Context
LLM
Answer
“What is the population of Paris, France?”
“The population of Paris, France, according to the most recent census report, is approximately 2.2 million people.”

The Models

We tested 22 models from model developers like Anthropic, Google, Meta, OpenAI and more. We wanted to test the effects of context windows and open vs. closed source models. So, we selected 12 open-source and 10 closed-source models with context windows ranging from 4K to 2M tokens.

Developer	Model	Released	Size (Param.)	Open/Closed	Context Window
Anthropic	claude-3-5-sonnet-20240620	6/20/24	NA	closed	200k
Anthropic	claude-3-haiku-20240307	3/7/24	NA	closed	200k
Anthropic	claude-3-opus-20240229	2/29/24	NA	closed	200k
Cohere	command-r-plus	4/4/24	104b	closed	128k
Google	gemini-1.0-pro	12/13/23	NA	closed	32k
Google	gemini-1.5-flash-001	5/24/24	NA	closed	2000k
Google	gemini-1.5-pro-001	5/24/24	NA	closed	1000k
OpenAI	gpt-3.5-turbo-0125	1/25/23	NA	closed	16k
OpenAI	gpt-4o-2024-05-13	5/13/24	NA	closed	128k
Mistral	mistral-large-2402	2/26/24	NA	closed	32k
Databricks	dbrx-instruct	3/27/24	132B	open	32k
Google	gemma-7b-it	2/21/24	7b	open	8k
Meta	meta-llama-3-70b-instruct	4/18/24	70b	open	8k
Meta	meta-llama-3-8b-instruct	4/18/24	8b	open	8k
Mistral	mistral-7b-instruct-v0.3	5/22/24	7b	open	32k
Mistral	mixtral-8×22b-instruct-v0.1	4/10/24	176b	open	64k
Mistral	mixtral-8×7b-instruct-v0.1	1/25/24	8×7b	open	32k
Alibaba	qwen-2-1.5b-instruct	6/6/24	1.5b	open	32k
Alibaba	qwen-2-72b-instruct	6/7/24	72b	open	128k
Alibaba	qwen-2-7b-instruct	6/6/24	7b	open	128k
Snowflake	qwen1.5-32b-chat	4/6/24	32b	open	32k
Snowflake	snowflake-arctic-instruct	4/24/24	480B	open	4k

The Rankings

Overall Rankings

Our team conducted extensive testing over a two month period, with many iterations as newer models were released. Here are our winners across each category.

Top Models for RAG Applications

Overall Winners for RAG	Best model	Claude 3.5 Sonnet due to great performance on all tasks with context support up to 200k.	
	Best performance for the cost	Gemini 1.5 Flash due to great performance on all tasks with context support upto 1M.	
	Best open-source model	Qwen2-72B-Instruct due to great performance in SCR and MCR with context support upto 128k.	
Short Context RAG (<5k tokens)	Best closed source model	Claude 3.5 Sonnet	**0.97**
	Best open source model	Llama-3-70b-chat	**0.95**
	Best performance for the cost	Gemini 1.5 Flash	**0.94**
Medium Context RAG (5k to 25k tokens)	Best closed source model	Gemini 1.5 Flash	**1.0**
	Best open source model	Qwen1.5-72B	**1.0**
	Best performance for the cost	Gemini 1.5 Flash	**1.0**
Long Context RAG (40k to 100k tokens)	Best closed source model	Claude 3.5 Sonnet	**1.0**
	Best performance for the cost	Gemini 1.5 Flash	**0.92**

Overall Winners for RAG

Throughout testing, a handful of models really surprised our team. We picked three overall winners across all tests and all context lengths.

Best Model

Claude 3.5 Sonnet

We were extremely impressed by Anthropic's latest set of models. Not only was Sonnet able to perform excellently across short, medium, and long context windows, scoring an average of 0.97, 1, and 1 respectively across tasks, but the model's support of up to a 200k context window suggests it could support even larger datasets than we tested.

Best Performance for the Cost

Gemini 1.5 Flash

Gemini 1.5 Flash offered a great balance of performance and cost. It earned a 0.94, 1, and 0.92 across short, medium, and long context task types. While not as robust as other models, Gemini did this at a fraction of the cost. The $ per Million prompt tokens cost was $0.35 for Flash vs. $3 for Sonnet. Even more starkly, the $ per Million response token cost was $1.05 for Flash vs. $15 for Sonnet. For high-volume applications or use cases where some margin of error is acceptable, Flash is a great choice.

Best Open-Source Model

Qwen2-72b-instruct

A newcomer to the Index, Alibaba launched its Qwen-2 model series in June 2024. The Qwen2-72b-instruct model performed on par with Meta's Llama-3-70b-instruct model during short and medium context testing. What set Qwen2 apart from other open-source models was its supported context length of 128K tokens. For context, the next largest supported context length by an open source model was Mistral's Mixtral-8×22b model, which supports a context length of 64k tokens.

Short Context RAG (SCR)

Less than 5k tokens

The Short Context RAG aims to determine the most effective model for comprehending short contexts upto 5k tokens. It focuses on identifying any loss of information and reasoning ability within these contexts. This approach is akin to looking up information in few pages of a book, making it well suited to tasks that require domain-specific information.

Short Context Result Snapshot

Category	Provider	Description	Score
Best closed-source model	A\	Claude-3-5-sonnet Claude-3-5-Sonnet and Claude-3-opus tied with scores of 0.97 followed closely by Open AI’s GPT-4o with a score of 0.96. Sonnet was our choice as it achieved this performance at a lower cost than its close competitors.	**0.97**
Worst closed-source model		Command-r-plus Cohere's command-r-plus earned a 0.86 during short context testing. While this was better than Open AI’s gpt 3.5-turbo and Google’s Gemini which both scored a 0.84, Cohere’s price was on par with Anthropic’s Claude 3.5 Sonnet at $3/M prompt tokens and $15/M response tokens, making it our choice for the worst closed-source model.	**0.86**
Best open-source model	∞	Llama-3-70b-instruct Alibaba's qwen2-72b-instruct and Meta's llama-3-70b-instruct tied during testing with scores of 0.95. We chose llama-3-70b-instruct, as qwen2-72b-instruct has 60% longer responses, which could increase the cost.	**0.95**
Worst open-source model	G	Gemma-7b-it Google's Gemma-7b-it demonstrated the poorest performance within the 7-billion-parameter category.	**0.65**

Category	Provider	Description	Score
Best performance for the cost	G	Gemini-1.5-flash-001 As mentioned in our overall results, Gemini-1.5-flash-001 performed the best at a fraction of the cost of models like Claude-3.5-sonnet, making it our choice for this category.	**0.94**
Best small open model	∞	llama-3-8b-instruct Meta's llama-3-8b-instruct surpassed several recent large models like Snowflake's Arctic.	**0.89**

Medium Context RAG (MCR)

5k - 25k tokens

The Medium Context RAG aims to determine the most effective model for comprehending long contexts spanning from 5k to 25k tokens. It focuses on identifying any loss of information and reasoning ability within these extensive contexts. This task is akin to doing RAG on a few book chapters.

Additionally, we experimented with a prompting technique known as Chain-of-Note to improve performance as it has worked for short context. This prompting approach helped improve the performance of claude-3-haiku, mistral-7b-instruct-v0.3, and databricks/dbrx-instruct by a few points.

Interestingly, models performed extremely well across the board, showcasing that medium context lengths appear to be the sweet spot for most LLMs.

Medium Context Result Snapshot

Category	Provider	Description	Score
Best closed-source model	G	**Gemini-1.5-flash-001** During testing, many models scored a perfect 1.0 score: • Anthropic: claude-3-5-sonnet, claude-3-opus, • Cohere: command-r-plus • Google: gemini-1.5-flash-001, gemini-1.5-pro-001 • Mistral: mistral-large • OpenAI: gpt-4o-2024-05-13 We ultimately chose Gemini-1.5-flash-001 for its low cost.	**1.0**
Worst closed-source model	A\	**Claude-3-Haiku** Claude-3-haiku performed the worst but still scored a 0.96.	**0.96**
Best open-source model		**qwen2-72b-instruct** Alibaba's qwen2-72b-instruct scored a perfect 1.0 and had flawless performance up to 25k tokens. Note that llama-3-70b-instruct does not support beyond 8K context length.	**1.0**
Worst open-source model	M	**Mistral-7b-instruct-v0.3** Mistral-7b-instruct-v0.3 had good performance however when considering cost, we felt there were better options.	**0.94**

Category	Provider	Description	Score
Best performance for the cost	G	Gemini-1.5-flash-001 Gemini-1.5-flash-001 scored a perfect 1.0. Similar to our rationale for when evaluating models for small context testing, Gemini-1.5-flash-001 also performed the best at a fraction of the cost, making it our choice for this category.	**1.0**
Best small open model		qwen2-7b-instruct Alibaba's qwen2-7b-instruct scored the best among the 7b models.	**0.96**

Long Context RAG (LCR)

40k - 100k tokens

The Long Context RAG aims to determine the most effective model for comprehending the longest contexts up to 100k tokens. It focuses on identifying any loss of information and reasoning ability within these extensive contexts.
Note: Some models do not support long context length (i.e., >40k tokens). These models were not tested for the Long Context scenario. We were unable to test Qwen2-72b-instruct as part of LCR testing.

Long Context Result Snapshot

Category	Provider	Description	Score
Best closed-source model	A\	**Claude-3-5-Sonnet** During long context testing, many models scored a perfect 1.0 score, highlighting the power and quality of models from top-tier providers. Among those scoring a perfect 1.0 were: • Anthropic: claude-3-5-sonnet, claude-3-opus • Google: gemini-1.5-pro-001 • Open AI: gpt-4o Between these models, it came down to Claude-3-5-Sonnet and Gemini-1.5-pro-001, which were both comparable on performance and price. Ultimately we chose Claude-3-5-Sonnet due to its better performance across short and medium-context testing.	**1.0**
Worst closed-source model	A\	**Claude-3-Haiku** Claude-3-haiku performed the worst with a score of 0.7. During testing, the model struggled to maintain accuracy with context lengths greater than 60k tokens.	**0.7**
Best performance for the cost	G	**Gemini-1.5-flash-001** Google's gemini-1.5-flash-001 scored an impressive 0.92 and only faced issues when provided with 80000 token context length.	**0.92**

Our Methodology

The Hallucination Index is an ongoing initiative to evaluate and rank the largest and most popular LLMs propensity to hallucinate across common task types. The models were evaluated using a diverse set of datasets, chosen for their popularity and ability to challenge the models' abilities to stay on task. Below is the methodology used to create the Hallucination Index. Our RAG methodology is designed to rigorously evaluate RAG models across a variety of dimensions, ensuring both factual accuracy and contextual adherence.

1 Model Selection

The Hallucination Index evaluated the largest and most popular LLMs available today. These LLMs were chosen by surveying popular LLM repos, leaderboards, and industry surveys. The LLMs selected represent a combination of open-source and closed-source models of varying sizes. This domain is evolving, with new models being released weekly.

The Hallucination Index will be updated every two quarters. To see an LLM added to the Index, contact us here.

2 Task Type Selection

Next, LLMs were tested across three common task types to observe their performance. We selected tasks relevant to developers and end-users and tested each LLM's ability to operate with context of different lengths.

Why short, medium & long context RAG tasks?

Context length affects the design of a RAG system by influencing retrieval strategies, computational resource needs, and the balance between precision and breadth. We conducted 3 experiments to gauge the state of LLMs' performance in different contexts lengths.

For short context lengths (less than 5,000 tokens), the pros are faster responses, better precision, and simplicity. However, they can miss out on broader context and might overfit to narrow scenarios. There is also a higher reliance on vector database precision to ensure relevant information retrieval.

Medium context lengths (5,000 to 25,000 tokens) offer a balance between detail and scope, providing more nuanced answers. They rely less on the pinpoint accuracy of vector databases, as they have more room to include context. However, they come with increased complexity and higher resource usage.

Long context lengths (40,000 to 100,000 tokens) handle detailed queries well, offering rich information and comprehensive understanding. Since extensive context can be included, the reliance on vector database precision decreases even further. The downside is slower response times, high computational costs, and potential inclusion of irrelevant information.

Short Context RAG

The SCR evaluation utilizes a variety of demanding datasets to test the robustness of models in handling short contexts:

We employ Chainpoll with GPT-4o, which leverages the strong reasoning power of GPT series models. By using a chain of thought technique to poll the model multiple times, we can better judge the correctness of the responses. This not only provides a metric to quantify potential hallucinations but also offers explanations based on the provided context, a crucial feature for RAG systems.

Medium and Long Context RAG

Our methodology focuses on models' ability to comprehensively understand extensive texts in medium and long contexts.

We extract text from very recent 10k documents of a company, divide it into chunks, and designate one of these chunks as the needle chunk. Using these chunks, we construct the necessary dataset by varying the location of the needle. We create a retrieval question that can be answered using the needle. The LLM has to answer the question using the context containing the needle.

Medium context lengths - 5k, 10k, 15k, 20k, 25k

Long context lengths - 40k, 60k, 80k, 100k

We designed the task with these considerations:

- All the text in context should be of single domain.
- Response should always be correct with short context to confirm the influence of long context.
- The question cannot be answered from memory of pre-training. It should not be a general old fact.
- Measuring the influence of position requires keeping the context, information, and question the same and altering only the location of the information.
- Avoid standard dataset as there can be test leakage.

Effect of prompting technique on performance

Additionally we experimented with a prompting technique known as Chain-of-Note to improve performance as it has worked for short context.

Evaluation

Adherence to context is evaluated using a custom LLM-based assessment, checking for the relevant answer within the response.

3 Dataset Selection

Short Context Rag

The Hallucination Index assesses LLM performance by leveraging 4 popular and 2 proprietary datasets. The datasets effectively challenge each LLM's capabilities relevant to the task at hand. For this task we convert the query and document to form the input prompt with context.

DROP: Reading comprehension benchmark which requires Discrete Reasoning Over the content of Paragraphs. Answering requires resolving references in a question, perhaps to multiple input positions, and performing discrete operations over them (such as addition, counting, or sorting).

Microsoft MS Macro: A dataset containing queries and paragraphs with relevance labels

.

HotpotQA: A dataset with Wikipedia-based question-answer pairs that require finding and reasoning over multiple supporting documents to answer.

ConvFinQA: A dataset to study the chain of numerical reasoning in conversational question answering. It poses great challenge in modeling long-range, complex numerical reasoning paths in real-world conversations.

Medium Context RAG

We extract text from very recent 10k documents of a company, divide it into chunks, and designate one of these chunks as the needle chunk. Using these chunks, we construct the necessary dataset by varying the location of the needle. We keep the needle at 20 varying locations per each context length to test the performance.

For the dataset with a context length of 10k, we will create 20 samples, keeping the “info” at different positions in the context—0, 500, 1000, 1500, .., 9000, 9500.

Similarly, for the dataset with context length of 100k, we will create 20 samples where we keep the “info” at different positions in the context - 0, 5000, 10000, 15000, .., 90000, 95000.

4 Experimentation

Once LLMs, Task Types, and Datasets were selected, experimentation began. The experimentation process we used is as follows:

- We use the prompt format as per the model.
- We add the context in the prompt with a simple format for all tasks. Additionally we also conduct chain-of-note prompt experiments for MCR and LCR experiments.
- Generation: The generations are done using private APIs, Together.ai, and hosting model on HuggingFace.

5 Evaluation

Scoring

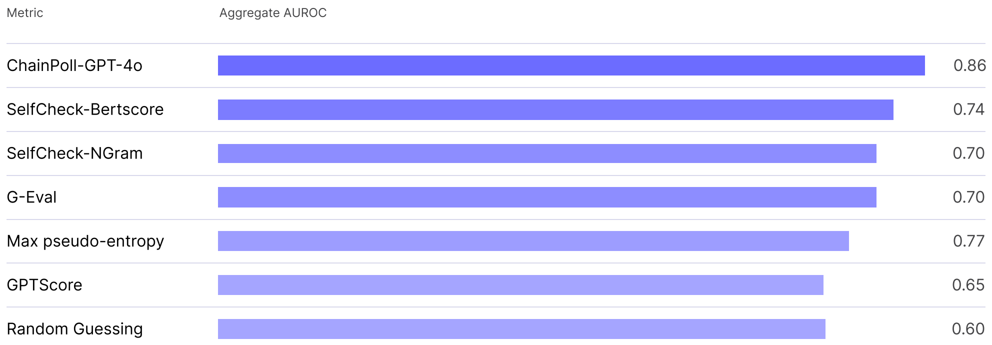
After preparing the prompts and generation for each model and dataset, they were evaluated using ChainPoll to obtain the task score. ChainPoll utilizes the strong reasoning abilities of GPTs and employs a technique of polling the model multiple times to assess the accuracy of the response. This approach not only quantifies the extent of potential errors but also provides an explanation based on the given context, particularly in the case of RAG-based systems.

Chainpoll: A High Efficacy Method for LLM Hallucination Detection

A high accuracy methodology for hallucination detection that provides an 85% correlation with human feedback - your first line of defense when evaluating model outputs.

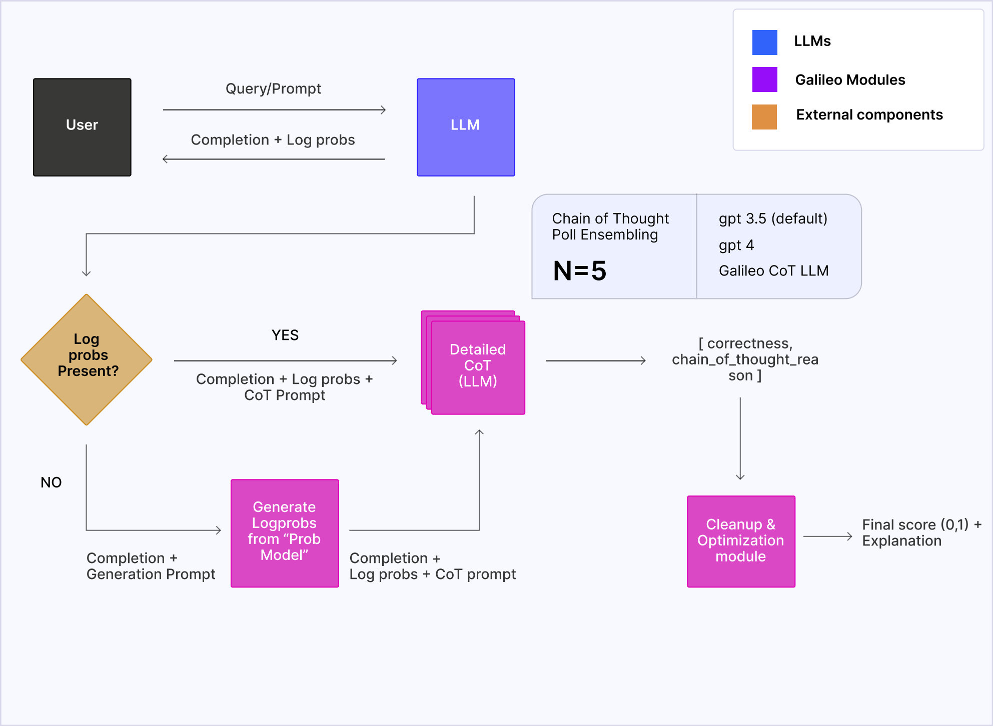
-ChainPoll: a novel approach to hallucination detection that is substantially more accurate than any metric we've encountered in the academic literature. Across a diverse range of benchmark tasks, the ChainPoll outperforms all other methods – in most cases, by a huge margin.

- ChainPoll dramatically out-performs a range of published alternatives – including SelfCheckGPT, GPTScore, G-Eval, and TRUE – in a head-to-head comparison on RealHall.
- ChainPoll is also faster and more cost-effective than most of the metrics listed above.
- Unlike all other methods considered here, ChainPoll also provides human-readable verbal justifications for the judgments
- it makes, via the chain-of-thought text produced during inference.
- Though much of the research literature concentrates on the the easier case of closed-domain hallucination detection, we show that ChainPoll is equally strong when detecting either open-domain or closed domain hallucinations. We develop versions of ChainPoll specialized to each of these cases: ChainPoll-Correctness for open-domain and ChainPoll-Adherence for closed-domain.



How does this work?

Chainpoll piggybacks on the strong reasoning power of your LLMs, but further leverages a chain of thought technique to poll the model multiple times to judge the correctness of the response. This technique not only provides a metric to quantify the degree of potential hallucinations, but also provides an explanation based on the context provided, in the case of RAG based systems.


LLMs
Galileo Modules
External components
User
Query/Prompt
Completion + Log probs
LLM
Chain of Thought Poll Ensembling
N=5
gpt 3.5 (default)
gpt 4
Galileo CoT LLM
Log probs Present?
YES
Completion + Log probs + CoT Prompt
Detailed CoT (LLM)
[correctness, chain_of_thought_rea son]
NO
Completion + Generation Prompt
Generate Logprobs from "Prob Model"
Completion + Log probs + CoT prompt
Cleanup & Optimization module
Final score (0,1) + Explanation

Evaluation

We selected an LLM-based evaluation to keep the approach scalable. ChainPoll powers the metrics used to evaluate output propensity for hallucination.

Task score

The final score shown is calculated as the mean of the score for each task dataset. The dataset score is the mean of the ChainPoll score for each sample.

Learn more

We have developed a comprehensive set of RAG metrics to cover various evaluation aspects of these models. Our documentation provides a detailed breakdown of each RAG metric and our methodologies.

About Context Adherence

Context Adherence evaluates the degree to which a model's response aligns strictly with the given context, serving as a metric to gauge closed-domain hallucinations, wherein the model generates content that deviates from the provided context.

The higher the Context Adherence score (i.e., it has a value of 1 or close to 1), the more likely the response is to contain only information from the context provided to the model.

The lower the Context Adherence score (ie., it has a value of 0 or close to 0), the response is more likely to contain information not included in the context provided to the model.

These metrics are powered by ChainPoll, a hallucination detection methodology developed by Galileo Labs. You can read more about ChainPoll here: https://arxiv.org/abs/2310.18344

6 How to use the Hallucination Index for LLM selection?

While our model ranking provides valuable insights for various tasks, we acknowledge that it does not cover all applications and domains comprehensively. To address this, we have plans to incorporate additional models and datasets in the future. To request a specific model, get in touch below. In the meantime, here's a suggested approach to refine your model selection process:

Task Alignment

Begin by identifying which of our benchmarking task types aligns most closely with your specific application.

Top 3 Model Selection

Based on your criteria, carefully select the three top-performing models for your identified task. Consider factors such as performance, cost, and privacy with your objectives.

Exploration of New Models

Extend your model pool by adding any additional models you believe could deliver strong performance in your application context. This proactive approach allows for a more comprehensive evaluation.

Data Preparation

Prepare a high-quality evaluation dataset using real-world data specific to your task. This dataset should be representative of the challenges and nuances to be faced in production.

Performance Evaluation

Execute a thorough evaluation of the selected models using your prepared dataset. Assess their performance based on relevant metrics, ensuring a comprehensive understanding of each model's strengths and weaknesses.

By following these steps, you'll gain a nuanced perspective on model suitability for your application, enabling you to make informed decisions in selecting the most appropriate model. Stay tuned for updates as we expand our model offerings to further cater to diverse applications and domains.

Appendix

SCR Task - Model Performance Data

Model	drop	hotpot_qa	convfinqa	ms_marco	model_avg	$/M prompt tokens	$/M response tokens
dataset_avg	0.84	0.85	0.86	0.91	0.86	2.18	7.67
claude-3-5-sonnet-20240620	0.98	0.96	0.99	0.96	0.97	3	15
claude-3-opus-20240229	0.96	0.96	1	0.94	0.97	15	75
gpt-4o-2024-05-13	0.97	0.93	1	0.95	0.96	5	15
gemini-1.5-pro-001	0.93	0.95	0.98	0.93	0.95	3.5	10.5
meta-llama-3-70b-instruct	0.94	0.94	0.97	0.95	0.95	0.9	0.9
mistral-large-2402	0.94	0.93	0.97	0.95	0.95	8	24
qwen2-72b-instruct	0.97	0.94	0.97	0.93	0.95	0.9	0.9
gemini-1.5-flash-001	0.92	0.92	0.98	0.95	0.94	0.35	1.05
mixtral-8×22b-instruct-v0.1	0.93	0.89	0.96	0.94	0.93	1.2	1.2
claude-3-haiku-20240307	0.93	0.87	0.97	0.93	0.92	0.25	1.25
meta-llama-3-8b-instruct	0.87	0.88	0.92	0.9	0.89	0.2	0.2
dbrx-instruct	0.86	0.85	0.89	0.9	0.88	1.2	1.2
qwen1.5-32b-chat	0.87	0.86	0.87	0.89	0.87	0.8	0.8
command-r-plus	0.81	0.88	0.88	0.89	0.86	3	15
snowflake-arctic-instruct	0.84	0.81	0.83	0.91	0.85	2.4	2.4
gemini-1.0-pro	0.78	0.76	0.89	0.91	0.84	0.5	1.5
gpt-3.5-turbo-0125	0.78	0.85	0.83	0.91	0.84	0.5	1.5
mixtral-8×7b-instruct-v0.1	0.8	0.81	0.8	0.89	0.83	0.6	0.6
mistral-7b-instruct-v0.3	0.69	0.8	0.72	0.92	0.78	0.2	0.2

Model	drop	hotpot_qa	convfinqa	ms_marco	model_avg	$/M prompt tokens	$/M response tokens
qwen2-7b-instruct	0.67	0.79	0.63	0.9	0.75	0.2	0.2
gemma-7b-it	0.6	0.68	0.49	0.84	0.65	0.2	0.2
qwen2-1.5b-instruct	0.34	0.48	0.31	0.7	0.46	0.1	0.1

MCR Task - Model Performance Data

Model	prompt_type	5000	10000	15000	20000	25000	model_avg	$/M prompt tokens	$/M response tokens
claude-3-5-sonnet-20240620	simple	1	1	1	1	1	1	3	15
claude-3-5-sonnet-20240620	with-con	1	1	1	1	1	1	3	15
claude-3-haiku-20240307	simple	1	1	0.9	0.95	0.95	0.96	0.25	1.25
claude-3-haiku-20240307	with-con	1	1	0.9	1	1	0.98	0.25	1.25
claude-3-opus-20240229	simple	1	1	1	1	1	1	15	75
claude-3-opus-20240229	with-con	1	1	1	1	1	1	15	75
command-r-plus	simple	1	1	1	1	1	1	3	15
command-r-plus	with-con	1	1	1	1	1	1	3	15
dbrx-instruct	simple	1	0.9	1	0.95	0.9	0.95	1.2	1.2
dbrx-instruct	with-con	1	1	0.95	1	0.95	0.98	1.2	1.2
gemini-1.5-flash-001	simple	1	1	1	1	1	1	0.35	1.05
gemini-1.5-flash-001	with-con	1	1	1	1	1	1	0.35	1.05
gemini-1.5-pro-001	simple	1	1	1	1	1	1	3.5	10.5
gemini-1.5-pro-001	with-con	1	1	1	1	1	1	3.5	10.5
gpt-4o-2024-05-13	simple	1	1	1	1	1	1	5	15
gpt-4o-2024-05-13	with-con	1	1	1	1	1	1	5	15
mistral-large-2402	simple	1	1	1	1	1	1	8	24
mistral-large-2402	with-con	1	1	1	1	1	1	8	24
mistral-7b-instruct-v0.3	simple	0.95	1	0.85	0.95	0.95	0.94	0.2	0.2
mistral-7b-instruct-v0.3	with-con	1	1	0.95	1	0.95	0.98	0.2	0.2
mixtral-8×22b-instruct-v0.1	simple	1	1	1	1	0.95	0.99	1.2	1.2

Model	prompt_type	5000	10000	15000	20000	25000	model_avg	$/M prompt tokens	$/M response tokens
mixtral-8×22b-instruct-v0.1	with-con	1	1	1	1	0.95	0.99	1.2	1.2
mixtral-8×7b-instruct-v0.1	simple	1	1	1	1	0.95	0.99	0.6	0.6
mixtral-8×7b-instruct-v0.1	with-con	1	1	1	1	0.95	0.99	0.6	0.6
qwen1.5-32b-chat	simple	1	1	1	1	0.95	0.99	0.8	0.8
qwen1.5-32b-chat	with-con	1	1	1	1	0.85	0.97	0.8	0.8
qwen2-72b-instruct	simple	1	1	1	1	1	1	0.9	0.9
qwen2-72b-instruct	with-con	1	1	1	1	1	1	0.9	0.9
qwen2-7b-instruct	simple	1	1	1	0.85	0.95	0.96	0.2	0.2
qwen2-7b-instruct	with-con	1	1	0.95	1	0.9	0.97	0.2	0.2

LCR Task - Model Performance Data

Model	prompt_type	4000	60000	80000	100000	model_avg	$/M prompt tokens	$/M response tokens
claude-3-5-sonnet-20240620	simple	1	1	1	1	1	3	15
claude-3-5-sonnet-20240620	with-con	1	1	1	1	1	3	15
claude-3-haiku-20240307	simple	0.85	0.95	0.55	0.45	0.7	0.25	1.25
claude-3-haiku-20240307	with-con	0.6	0.95	1	0.55	0.78	0.25	1.25
claude-3-opus-20240229	simple	1	1	1	1	1	15	75
claude-3-opus-20240229	with-con	1	1	1	1	1	15	75
command-r-plus	simple	1	0.95	0.95	0.9	0.95	3	15
command-r-plus	with-con	1	0.95	0.95	0.9	0.95	3	15
gemini-1.5-flash-001	simple	1	1	0.7	1	0.92	0.35	1.05
gemini-1.5-flash-001	with-con	1	1	0.65	1	0.91	0.35	1.05
gemini-1.5-pro-001	simple	1	1	1	1	1	3.5	10.5
gemini-1.5-pro-001	with-con	1	1	1	1	1	3.5	10.5
gpt-4o-2024-05-13	simple	1	1	0.95	1	0.99	5	15
gpt-4o-2024-05-13	simple	1	1	1	1	1	5	15

Brought to you by Galileo

www.rungalileo.io

Galileo is the leading platform for enterprise GenAI evaluation and observability. The Galileo platform, powered by Luna™ Evaluation Foundation Models (EMs), supports AI teams across the development lifecycle, from building and iterating to monitoring and protection. Galileo is used by AI teams at startups to Fortune 50 companies. Visit rungalileo.io to learn more about the Galileo suite of products.