Guardrails - Market Research

[Context](#_agtdv9dj9a7n)

[Suggestions for the current Guardrails Development](#_rg15vq6ej80a)

[Survey, Review Of many systems](#_e48u8hq4booc)

[Safeguarding Large Language Models: A Survey - 2024-06](#_4ygzrotcaknf)

[YouTube: Advanced Guardrails for AI Agents, Full Tutorial](#_lfntf9vttnf1)

[BM25 definition](#_sb0gepjkc3wp)

[YouTube: Guardrails Crash Course for Beginners](#_d8kjjrycvlpw)

[Github AI Anytime / Guardrails Crash Course](#_ucy15abjmitw)

[YouTube: How to Make a Safety Net for Your LLM Applications | NVIDIA](#_tzip8pm1udyn)

[Labelled datasets](#_jbksh4sne85d)

[Papers With code](#_lxr58spibo5r)

[Guardrails AI](#_e1gcmhorzfbh)

[Website information, Features](#_tq359o731q9g)

[Guardrails AI Github, Open Source component](#_z6fzfd26yun8)

[Guardrails Hub (Benchmarks and data sets)](#_lyfjazgcmkxa)

[Security - Detect Jailbreak](#_sbxpb58frov2)

[Restrict to Topic (good for general LLM Applications)](#_1hbp3c29jcsf)

[Uses Rules - Secrets Present](#_li9wzb9dqhq8)

[Guardrails PII (38 types of PII, with medical use cases)](#_lz0ke3dc2i2h)

[LLM RAG Evaluator](#_grptojz10kz8)

[PapersWithCode - Wiki QA for RAG](#_v538fdsoxuiw)

[Top paper - Structural Self-Supervised Objectives for Transformers (0.954 MAP)](#_ef45qmqjlbua)

[(to do) RAIL (Reusable AI Language)](#_2c43md1szeqq)

[Chrunchbase info on Guardrails AI](#_p22xskg2moz0)

[NVIDIA Guardrails NeMo](#_7aq36mvtcvfk)

[Introduction to Guardrails NeMo](#_1mhnfhhhu135)

[Other Background Sources](#_sinlmflx0m9m)

[YouTube: Your AI Applications Need Guardrails: Here's How To Build Them](#_9qofsjlibqbd)

[YouTube video from May 2023, 3 min long.](#_3yem061rvfdh)

[NeMo Guardrails Github Open-Source](#_t8vgyhpsqyx)

[Paper: “NeMo Guardrails: A Toolkit for Controllable and Safe LLM Applications with Programmable Rails” Oct 2023](#_ji5u0jp2o7pv)

[Other Background Sources](#_opkfmlb6ipqu)

[Github NVIDIA NeMo Guardrails > Colang Guide](#_8yluc39w74xr)

[YouTube: NVIDIA NeMo GUardrails: Full Walkthrough for Chatbots / AI](#_59xyekuffwfv)

[Llama Guard](#_y85i5y45n3cb)

[DeepChecks](#_8m7bda2qit75)

[Pangea](#_mccsmbibsgf8)

[Smarsh email routing functionality](#_ymwapmv9qn20)

# Context

Request - What is the GUI and Product Design of other Guardrails products? [krishna.s@ccube.com](mailto:krishna.s@ccube.com)

It can be helpful to load papers into <https://notebooklm.google/>

<https://support.google.com/notebooklm/answer/14278184?hl=en>

Greg q) Could we integrate NotebookLM as a feature in our RAG product, Nova? If a user wants to do more than just see the final text chunk of a document. For

* Document summarization
* Mind map (outlining)
* Audo summaries
* Knowledge synthesis (over multiple documents)

## Suggestions for the current Guardrails Development

[sakshi.m@ccube.com](mailto:sakshi.m@ccube.com), start with this section

See Guardrails AI open source code - do we want to use anything from here?

<https://github.com/guardrails-ai/guardrails>

See Guardrails Hub, section below, for labelled datasets that may help. This shows 66 already created “guardrails validators” that can be installed or configured with the Guardrails AI open-source system.

<https://hub.guardrailsai.com/>

Paper: “NeMo Guardrails: A Toolkit for Controllable and Safe LLM Applications with Programmable Rails” Oct 2023

<https://arxiv.org/abs/2310.10501>

NVIDIA NeMo Guardrails > Colang Guide <https://docs.nvidia.com/nemo/guardrails/latest/user-guides/colang-language-syntax-guide.html>

Colang can be a good language to model chatbot interactions and application/guardrails responses

Colang is a way to implement rules

For integrating Guardrails into Nova, one new “security feature” we can add is Detecting Jailbreaking. Integrate the open-source code below. Their blog reports a high accuracy, compared to other code or foundation models.

<https://hub.guardrailsai.com/validator/tryolabs/restricttotopic>

For example, if Nova loads financial data into a vector DB, this support a guardrails front-end check on user input to the chatbot, to be on topic with finance questions. When we change to a new topic for the vector store, the front end can enforce staying on topic for that data store. It can also validate that an LLM produces valid JSON output.

Another “fast way” to restrict a topic, or for other Guardrails. See the YouTube by James Briggs, below. Min 8.

<https://github.com/aurelio-labs/semantic-router>

# Survey, Review of Many Systems

## How to Combine guardrails.ai’s RAIL and NVIDIA’s Guardrails Nemo?

Q) does it make sense to use both guardrails.ai's RAIL system and nvidia's Guardrails Nemo system and colang language for the same LLM application? How do they best combine? what is a best practice architecture? How should M

(Google Gemini answer)

Yes, it can make sense to use both Guardrails AI's RAIL system and NVIDIA's NeMo Guardrails for the same LLM application, as they offer different strengths in safeguarding LLMs. A best practice architecture would leverage their strengths by **using RAIL for specific, detailed, and configurable rules, while integrating NeMo Guardrails for broader safety, security, and controlled dialog orchestration**. [[1](https://arize.com/blog-course/guardrails-what-are-they-and-how-can-you-use-nemo-and-guardrails-ai-to-safeguard-llms/), [2](https://www.fuzzylabs.ai/blog-post/guardrails-for-llms-a-tooling-comparison), [3](https://docs.nvidia.com/nemo/guardrails/latest/index.html), [4](https://developer.nvidia.com/blog/securing-generative-ai-deployments-with-nvidia-nim-and-nvidia-nemo-guardrails/)]

**How they combine: [**[**1**](https://arize.com/blog-course/guardrails-what-are-they-and-how-can-you-use-nemo-and-guardrails-ai-to-safeguard-llms/)**,** [**1**](https://arize.com/blog-course/guardrails-what-are-they-and-how-can-you-use-nemo-and-guardrails-ai-to-safeguard-llms/)**,** [**5**](https://medium.com/data-science/safeguarding-llms-with-guardrails-4f5d9f57cff2)**,** [**5**](https://medium.com/data-science/safeguarding-llms-with-guardrails-4f5d9f57cff2)**]**

* **RAIL for granular rules:** Guardrails AI's RAIL system excels at defining detailed rules for LLM outputs, including format, content, and even semantic validations. This can be used to enforce specific styles, structures, or prevent certain types of outputs (e.g., preventing hallucination or toxic language). [[1](https://arize.com/blog-course/guardrails-what-are-they-and-how-can-you-use-nemo-and-guardrails-ai-to-safeguard-llms/), [1](https://arize.com/blog-course/guardrails-what-are-they-and-how-can-you-use-nemo-and-guardrails-ai-to-safeguard-llms/), [5](https://medium.com/data-science/safeguarding-llms-with-guardrails-4f5d9f57cff2), [5](https://medium.com/data-science/safeguarding-llms-with-guardrails-4f5d9f57cff2)]
* **NeMo Guardrails for broader safety and orchestration:** NVIDIA's NeMo Guardrails focuses on broader safety features like content safety, off-topic detection, and retrieval-augmented generation (RAG) enforcement. It also provides a framework for orchestrating multiple guardrails and interacting with other NVIDIA services for deployment. [[4](https://developer.nvidia.com/blog/securing-generative-ai-deployments-with-nvidia-nim-and-nvidia-nemo-guardrails/), [4](https://developer.nvidia.com/blog/securing-generative-ai-deployments-with-nvidia-nim-and-nvidia-nemo-guardrails/), [6](https://developer.nvidia.com/blog/how-to-safeguard-ai-agents-for-customer-service-with-nvidia-nemo-guardrails/), [6](https://developer.nvidia.com/blog/how-to-safeguard-ai-agents-for-customer-service-with-nvidia-nemo-guardrails/), [7](https://developer.nvidia.com/nemo-guardrails)]
* **Example Scenario:** Imagine an LLM-powered chatbot. RAIL could be used to ensure the chatbot's responses adhere to a specific format, while NeMo Guardrails would be used to prevent the chatbot from engaging in off-topic discussions or generating harmful content. [[1](https://arize.com/blog-course/guardrails-what-are-they-and-how-can-you-use-nemo-and-guardrails-ai-to-safeguard-llms/), [1](https://arize.com/blog-course/guardrails-what-are-they-and-how-can-you-use-nemo-and-guardrails-ai-to-safeguard-llms/), [6](https://developer.nvidia.com/blog/how-to-safeguard-ai-agents-for-customer-service-with-nvidia-nemo-guardrails/), [6](https://developer.nvidia.com/blog/how-to-safeguard-ai-agents-for-customer-service-with-nvidia-nemo-guardrails/)]

**Best Practice Architecture: [**[**3**](https://docs.nvidia.com/nemo/guardrails/latest/index.html)**,** [**3**](https://docs.nvidia.com/nemo/guardrails/latest/index.html)**,** [**4**](https://developer.nvidia.com/blog/securing-generative-ai-deployments-with-nvidia-nim-and-nvidia-nemo-guardrails/)**,** [**4**](https://developer.nvidia.com/blog/securing-generative-ai-deployments-with-nvidia-nim-and-nvidia-nemo-guardrails/)**]**

1. **Define a clear scope:** Identify the specific safety, security, and operational needs for your LLM application. [[3](https://docs.nvidia.com/nemo/guardrails/latest/index.html), [3](https://docs.nvidia.com/nemo/guardrails/latest/index.html), [4](https://developer.nvidia.com/blog/securing-generative-ai-deployments-with-nvidia-nim-and-nvidia-nemo-guardrails/), [4](https://developer.nvidia.com/blog/securing-generative-ai-deployments-with-nvidia-nim-and-nvidia-nemo-guardrails/)]
2. **Choose RAIL for detailed rules:** Use RAIL to define specific rules and validation checks for LLM outputs, focusing on areas where detailed control is needed. [[1](https://arize.com/blog-course/guardrails-what-are-they-and-how-can-you-use-nemo-and-guardrails-ai-to-safeguard-llms/), [1](https://arize.com/blog-course/guardrails-what-are-they-and-how-can-you-use-nemo-and-guardrails-ai-to-safeguard-llms/), [5](https://medium.com/data-science/safeguarding-llms-with-guardrails-4f5d9f57cff2), [5](https://medium.com/data-science/safeguarding-llms-with-guardrails-4f5d9f57cff2)]
3. **Integrate NeMo Guardrails for broad safety and orchestration:** Use NeMo Guardrails to manage the broader safety aspects, including content safety, topic control, and integration with other NVIDIA services. [[1](https://arize.com/blog-course/guardrails-what-are-they-and-how-can-you-use-nemo-and-guardrails-ai-to-safeguard-llms/), [3](https://docs.nvidia.com/nemo/guardrails/latest/index.html), [3](https://docs.nvidia.com/nemo/guardrails/latest/index.html), [4](https://developer.nvidia.com/blog/securing-generative-ai-deployments-with-nvidia-nim-and-nvidia-nemo-guardrails/), [4](https://developer.nvidia.com/blog/securing-generative-ai-deployments-with-nvidia-nim-and-nvidia-nemo-guardrails/), [6](https://developer.nvidia.com/blog/how-to-safeguard-ai-agents-for-customer-service-with-nvidia-nemo-guardrails/), [6](https://developer.nvidia.com/blog/how-to-safeguard-ai-agents-for-customer-service-with-nvidia-nemo-guardrails/), [8](https://developer.nvidia.com/blog/measuring-the-effectiveness-and-performance-of-ai-guardrails-in-generative-ai-applications/)]
4. **Combine and Orchestrate:** Integrate NeMo Guardrails and RAIL, potentially using LangChain or other orchestration frameworks, to create a robust and flexible system. [[1](https://arize.com/blog-course/guardrails-what-are-they-and-how-can-you-use-nemo-and-guardrails-ai-to-safeguard-llms/), [1](https://arize.com/blog-course/guardrails-what-are-they-and-how-can-you-use-nemo-and-guardrails-ai-to-safeguard-llms/), [4](https://developer.nvidia.com/blog/securing-generative-ai-deployments-with-nvidia-nim-and-nvidia-nemo-guardrails/), [4](https://developer.nvidia.com/blog/securing-generative-ai-deployments-with-nvidia-nim-and-nvidia-nemo-guardrails/), [9](https://blogs.nvidia.com/blog/ai-chatbot-guardrails-nemo/), [9](https://blogs.nvidia.com/blog/ai-chatbot-guardrails-nemo/)]
5. **Monitor and Evaluate:** Continuously monitor the performance of your guardrails and evaluate their effectiveness using NeMo Guardrails' evaluation tools. [[8](https://developer.nvidia.com/blog/measuring-the-effectiveness-and-performance-of-ai-guardrails-in-generative-ai-applications/), [8](https://developer.nvidia.com/blog/measuring-the-effectiveness-and-performance-of-ai-guardrails-in-generative-ai-applications/), [10](https://github.com/NVIDIA/NeMo-Guardrails), [10](https://github.com/NVIDIA/NeMo-Guardrails)]

**Benefits of this combined approach: [**[**4**](https://developer.nvidia.com/blog/securing-generative-ai-deployments-with-nvidia-nim-and-nvidia-nemo-guardrails/)**,** [**4**](https://developer.nvidia.com/blog/securing-generative-ai-deployments-with-nvidia-nim-and-nvidia-nemo-guardrails/)**,** [**6**](https://developer.nvidia.com/blog/how-to-safeguard-ai-agents-for-customer-service-with-nvidia-nemo-guardrails/)**,** [**6**](https://developer.nvidia.com/blog/how-to-safeguard-ai-agents-for-customer-service-with-nvidia-nemo-guardrails/)**]**

* **Enhanced Safety and Security:** Leverage the strengths of both tools for a more comprehensive approach to LLM safety and security. [[4](https://developer.nvidia.com/blog/securing-generative-ai-deployments-with-nvidia-nim-and-nvidia-nemo-guardrails/), [4](https://developer.nvidia.com/blog/securing-generative-ai-deployments-with-nvidia-nim-and-nvidia-nemo-guardrails/), [6](https://developer.nvidia.com/blog/how-to-safeguard-ai-agents-for-customer-service-with-nvidia-nemo-guardrails/), [6](https://developer.nvidia.com/blog/how-to-safeguard-ai-agents-for-customer-service-with-nvidia-nemo-guardrails/)]
* **Improved Control:** Gain more granular control over LLM outputs while maintaining a flexible and scalable architecture. [[1](https://arize.com/blog-course/guardrails-what-are-they-and-how-can-you-use-nemo-and-guardrails-ai-to-safeguard-llms/), [1](https://arize.com/blog-course/guardrails-what-are-they-and-how-can-you-use-nemo-and-guardrails-ai-to-safeguard-llms/), [3](https://docs.nvidia.com/nemo/guardrails/latest/index.html), [3](https://docs.nvidia.com/nemo/guardrails/latest/index.html)]
* **Simplified Deployment:** Benefit from the NeMo Guardrails ecosystem for easier deployment and integration with other NVIDIA services. [[4](https://developer.nvidia.com/blog/securing-generative-ai-deployments-with-nvidia-nim-and-nvidia-nemo-guardrails/), [4](https://developer.nvidia.com/blog/securing-generative-ai-deployments-with-nvidia-nim-and-nvidia-nemo-guardrails/), [9](https://blogs.nvidia.com/blog/ai-chatbot-guardrails-nemo/), [9](https://blogs.nvidia.com/blog/ai-chatbot-guardrails-nemo/)]
* **Scalability:** NeMo Guardrails and LangChain can help you scale your LLM applications effectively. [[4](https://developer.nvidia.com/blog/securing-generative-ai-deployments-with-nvidia-nim-and-nvidia-nemo-guardrails/), [4](https://developer.nvidia.com/blog/securing-generative-ai-deployments-with-nvidia-nim-and-nvidia-nemo-guardrails/), [9](https://blogs.nvidia.com/blog/ai-chatbot-guardrails-nemo/), [9](https://blogs.nvidia.com/blog/ai-chatbot-guardrails-nemo/)]

*Generative AI is experimental.*

[1] <https://arize.com/blog-course/guardrails-what-are-they-and-how-can-you-use-nemo-and-guardrails-ai-to-safeguard-llms/>

[2] <https://www.fuzzylabs.ai/blog-post/guardrails-for-llms-a-tooling-comparison>

[3] <https://docs.nvidia.com/nemo/guardrails/latest/index.html>

[4] <https://developer.nvidia.com/blog/securing-generative-ai-deployments-with-nvidia-nim-and-nvidia-nemo-guardrails/>

[5] <https://medium.com/data-science/safeguarding-llms-with-guardrails-4f5d9f57cff2>

[6] <https://developer.nvidia.com/blog/how-to-safeguard-ai-agents-for-customer-service-with-nvidia-nemo-guardrails/>

[7] <https://developer.nvidia.com/nemo-guardrails>

[8] <https://developer.nvidia.com/blog/measuring-the-effectiveness-and-performance-of-ai-guardrails-in-generative-ai-applications/>

[9] <https://blogs.nvidia.com/blog/ai-chatbot-guardrails-nemo/>

[10] <https://github.com/NVIDIA/NeMo-Guardrails>

## Safeguarding Large Language Models: A Survey - 2024-06

<https://arxiv.org/abs/2406.02622>

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. XX, NO. X, MAY 2024

Abstract—In the burgeoning field of Large Language Models (LLMs), developing a robust safety mechanism, colloquially known as “safeguards” or “guardrails”, has become imperative to ensure the ethical use of LLMs within prescribed boundaries. This article provides a systematic literature review on the current status of this critical mechanism. It discusses its major challenges and how it can be enhanced into a comprehensive mechanism dealing with ethical issues in various contexts. First, the paper elucidates the current landscape of safeguarding

mechanisms that major LLM service providers and the open-source community employ. This is followed by the techniques to evaluate, analyze, and enhance some (un)desirable properties that a guardrail might want to enforce, such as hallucinations, fairness, privacy, and so on. Based on them, we review techniques to circumvent these controls (i.e., attacks), to defend the attacks, and to reinforce the guardrails. While the techniques mentioned above represent the current status and the active research trends, we also discuss several challenges that cannot be

easily dealt with by the methods and present our vision on how to implement a comprehensive guardrail through the full consideration of multi-disciplinary approach, neural-symbolic method, and systems development lifecycle.

Index Terms—Large Language Models, Generative AI, Safeguards, Guardrails, Trustworthy AI

## YouTube: Advanced Guardrails for AI Agents, Full Tutorial

<https://www.youtube.com/watch?v=rMUycP_cp9g>

By James Briggs, April 2025. 22 min long

Does NOT use NVIDIA Guardrails, not LLM, but uses a hybrid index. Good for speed and accuracy.

1,897 views Apr 1, 2025 #ai #aiagents #coding

In this video we'll learn how to build advanced AI guardrails as part of a broader protective system for production-ready AI systems. We explore the use of hybrid vector space to develop this guardrails, which can be incredibly useful for chatbot use-cases with specific brand, topic, or behavioral guardrails.

📌 Code:

[https://github.com/aurelio-labs/seman…](https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqbGtCTnRTQVJXUFp0S3JQbUhubWZ0eVNBUmdFQXxBQ3Jtc0tsNmg5QW02WnNLdTlqWG1JajBGdWFkOUtJRVFHWXhQYV9UYk1rVGNvUWRHUVM5cEhBVnc1RTRDNGp2bEtza1dSNE92d0RidU43Tkl1RTRnTjVVNXNTaVhVSzNKbUhwWmdXeGxadHFlVzdpdGV0QkRkOA&q=https%3A%2F%2Fgithub.com%2Faurelio-labs%2Fsemantic-router%2Fblob%2Fmain%2Fdocs%2Fexamples%2Fhybrid-chat-guardrails.ipynb&v=rMUycP_cp9g)

NOTES:

Min 10-13, talks about where semantic can be helped with a hybrid, using sparse encodings like BM25.

The context is a chatbot that we just want to talk about the brand BYD, and not other brands.

Example 1: (all of these sentences correlate highly (57-69, using a semantic embedding model). THey all have a similar generic semantic meaning, outside of the brand.

How can I sell my Tesla?

How can I sell my Polestar?

How can I sell my BYD?

How can I sell my Rivian?

The semantic embedding is good at capturing the general intent and structure of the sentence.

Issue: if we want to prevent the chatbot from talking about other brands, this metric makes all look too similar. We have a prioritized feature that we care about, the brand.

For the sentences, all about the BYD brand. Sparse encoders are good with term matching

Tell me about the BYD Seal

How does BYD’s blade battery work

Is the BYD Atto 3 a good EV?

Can I sell my BYD?

How much can I sell my BYD for?

The correlations range from 0.48 to 0.88, using the semantic embedding.

Min 12, talks about how you have to pay to use an Aurelio account, for some of this presentation

(I would rather minimize paying)

Min 12:07, promotion code = JBVIDEOAGENT for $5 off

Then can do the hybrid chat demo

Note: Weaviate supports BM25 and hybrid indexes. Don’t have to pay

Min 18:30 shows how to use a fit method to increase the classification accuracy from 51.75% to 95%

A big improvement.

95.61% on the holdout dataset. The classification looks stable

### BM25 definition

(I know BM25 is like a type of TF-IDF index. I wanted details)

<https://en.wikipedia.org/wiki/Okapi_BM25>

In [information retrieval](https://en.wikipedia.org/wiki/Information_retrieval), Okapi BM25 (BM is an abbreviation of best matching) is a [ranking function](https://en.wikipedia.org/wiki/Ranking_function) used by [search engines](https://en.wikipedia.org/wiki/Search_engine) to estimate the [relevance](https://en.wikipedia.org/wiki/Relevance_(information_retrieval)) of documents to a given search query. It is based on the [probabilistic retrieval framework](https://en.wikipedia.org/wiki/Probabilistic_relevance_model) developed in the 1970s and 1980s by [Stephen E. Robertson](https://en.wikipedia.org/wiki/Stephen_E._Robertson), [Karen Spärck Jones](https://en.wikipedia.org/wiki/Karen_Sp%C3%A4rck_Jones), and others.

The name of the actual ranking function is BM25. The fuller name, Okapi BM25, includes the name of the first system to use it, which was the Okapi information retrieval system, implemented at [London](https://en.wikipedia.org/wiki/London)'s [City University](https://en.wikipedia.org/wiki/City_University,_London)[[1]](https://en.wikipedia.org/wiki/Okapi_BM25#cite_note-1) in the 1980s and 1990s. BM25 and its newer variants, e.g. BM25F (a version of BM25 that can take document structure and anchor text into account), represent [TF-IDF](https://en.wikipedia.org/wiki/TF-IDF)-like retrieval functions used in document retrieval.[[2]](https://en.wikipedia.org/wiki/Okapi_BM25#cite_note-robertson2009-2)

The ranking function[[edit](https://en.wikipedia.org/w/index.php?title=Okapi_BM25&action=edit&section=1)]

BM25 is a [bag-of-words](https://en.wikipedia.org/wiki/Bag_of_words_model) retrieval function that ranks a set of documents based on the query terms appearing in each document, regardless of their proximity within the document. It is a family of scoring functions with slightly different components and parameters. One of the most prominent instantiations of the function is as follows.

Modifications[[edit](https://en.wikipedia.org/w/index.php?title=Okapi_BM25&action=edit&section=3)]

* At the extreme values of the coefficient b BM25 turns into ranking functions known as BM11 (for b=1) and BM15 (for b=0).[[4]](https://en.wikipedia.org/wiki/Okapi_BM25#cite_note-4)
* BM25F[[5]](https://en.wikipedia.org/wiki/Okapi_BM25#cite_note-5)[[2]](https://en.wikipedia.org/wiki/Okapi_BM25#cite_note-robertson2009-2) (or the BM25 model with Extension to Multiple Weighted Fields[[6]](https://en.wikipedia.org/wiki/Okapi_BM25#cite_note-6)) is a modification of BM25 in which the document is considered to be composed from several fields (such as headlines, main text, anchor text) with possibly different degrees of importance, term relevance saturation and length normalization. BM25F defines each type of field as a stream, applying a per-stream weighting to scale each stream against the calculated score.
* BM25+[[7]](https://en.wikipedia.org/wiki/Okapi_BM25#cite_note-7) is an extension of BM25. BM25+ was developed to address one deficiency of the standard BM25 in which the component of term frequency normalization by document length is not properly lower-bounded; as a result of this deficiency, long documents which do match the query term can often be scored unfairly by BM25 as having a similar relevancy to shorter documents that do not contain the query term at all. The scoring formula of BM25+ only has one additional free parameter δ (a default value is 1.0 in absence of a training data) as compared with BM25: …..

## YouTube: Guardrails Crash Course for Beginners

<https://www.youtube.com/watch?v=XbriX2aYgqw>

By AI Anytime, 36 min

2,943 views Dec 12, 2024  [#ai](https://www.youtube.com/hashtag/ai) [#guardrails](https://www.youtube.com/hashtag/guardrails) [#guardrail](https://www.youtube.com/hashtag/guardrail)

Dive into the world of AI safety with my crash course for beginners on NVIDIA NeMo Guardrails, Llama Guard, and Guardrails AI! Learn how to add guardrails to AI systems, prevent risky outputs, and build reliable applications—all explained in simple terms.

Discord: [ / discord](https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqa3FwWGNGaGpDa21uelRELTVHN19ZejVQRnZwQXxBQ3Jtc0trU0RzZEtya1VPeXpPMHJfd19wT3dhOE1fSGdQbUdRSllua3dVLUlJal9OQ1k5Vy1ON0RaRkFvajdaT2lscThmZEVSRHJDbnJzUEpOSGFwaVFZQ0VFdzRLbnRsT0FuMXB2MG1XVFVoc2hRVzZvZWNMZw&q=https%3A%2F%2Fdiscord.com%2Finvite%2Faianytime&v=XbriX2aYgqw)

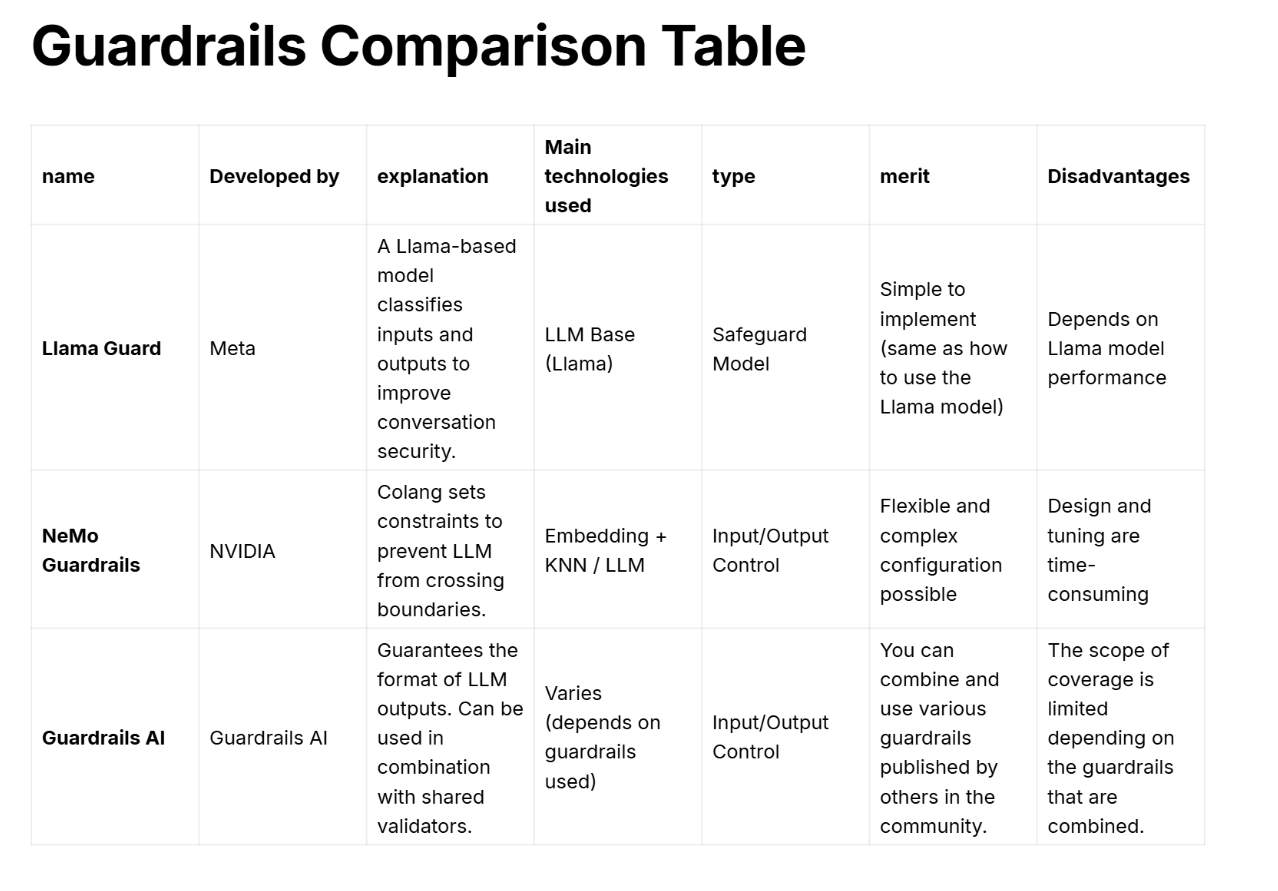
GitHub Repo: [https://github.com/AIAnytime/Guardrai...](https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqbUFmMU9jVkFNRTRMaVlEZnl2RURXUVJPNHVHUXxBQ3Jtc0tscmlKMW1YQTlwUkFyT0RZREd4SWo2a0dNVEVaWnBMYTVCTi1iSnUyZk5yMm81ekMxU19qaEtwMXVOUV9GbnhMbjQ5dXF2b2FWcVpGVThubk5NS0xwdjVEZlh1MHlDc1BrdjRGbVhTOE95cUYzQkJrVQ&q=https%3A%2F%2Fgithub.com%2FAIAnytime%2FGuardrails-Crash-Course&v=XbriX2aYgqw)

Paper you should read: [https://arxiv.org/pdf/2406.02622](https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqbE15Z3RNMk9OQnZDVldRZk1XLTFKdXdoWDJMd3xBQ3Jtc0tuMERYazhDWGNJR1FDRTlTeVRuUmEtY0VHSUxGN3Y1MWl2SGhEdHpBYVlEUGpRUkdETVpDU3NHa0c2Y1l1WUJZNUJ2bVlqUEQxTlc4aWJQdlNmdXVuZTRhZkFGNHU2ZEJRaXV1U21jeUQxdkI0YVhrNA&q=https%3A%2F%2Farxiv.org%2Fpdf%2F2406.02622&v=XbriX2aYgqw)

### Github AI Anytime / Guardrails Crash Course

<https://github.com/AIAnytime/Guardrails-Crash-Course>

<https://github.com/AIAnytime/Guardrails-Crash-Course/blob/main/Guardrails%20Comparison.pdf>



Notes:

Min 1:22. What air Canada lost in remarkable lying ai chatbot case. In Forbes

Min 2:56. Delivery firms ai chatbot goes rouge, curses at customer and criticizes Company. In Time.com

Min 4:24. Has a pdf for the crash course

Min 9:50

<https://arxiv.org/abs/2312.06674>

Llama Guard

Guardrails AI uses RAIL. Reliable AI markup language. Schema in xml

Min 11. Pdf: safeguarding large language models , a survey in june of twenty twenty four.

Nemo uses colang

Min 12:53. Huggingface. Llama-guard-3-8B

Check to see how it is maintained or supported

Min 17:34. 13 hazard categories

Min 18:10 google ai for developers. Safety settings. Can adjust with sliders.

## YouTube: How to Make a Safety Net for Your LLM Applications | NVIDIA

<https://www.youtube.com/watch?v=WNg2vqMrRB8>

By Deepchecks. 60 min

521 views Jun 23, 2024

In this session, Christopher Parisien, Research Manager at NVIDIA, and Shir Chorev, CTO & Co-Founder at Deepchecks, talked about LLM safety.

The session covered the live demos of Deepchecks' LLM Evaluation, Garak's LLM Vulnerability Scanner, and NVIDIA's NeMo Guardrails, showcasing how the combined offering enables building a holistic safety net.

Sign up for Deepchecks LLM Evaluation Solution here:

[https://deepchecks.com/llm-evaluation/](https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqbTFwcG4wUVd5ejY0amdnaldEOTBWcU4zejlWZ3xBQ3Jtc0tuYjV0ZXVnZEJMVDJKYy1MTmh1WWtzRDNDanJPV0hpZnpoSnlzckxfRkNiV3cwY3FodnFOeTlzRUxNY284T0l6UG5ieXhzVFpfUmZ2ZVJuTTVIUUZZSzZxRHhfeTV6V3MxekRoT3lZaFA5QnhGYlZ2aw&q=https%3A%2F%2Fdeepchecks.com%2Fllm-evaluation%2F&v=WNg2vqMrRB8)

Topics that were covered:

✅ Deepchecks' LLM Evaluation: Explore how the Deepchecks evaluation tool ensures the accuracy, reliability, and fairness of your LLM applications.

✅ Garak's LLM Vulnerability Scanner: Explore the power of automated vulnerability scanning to identify and mitigate potential security threats within your LLM applications.

✅ NVIDIA’s NeMo Guardrails: Learn about the advanced safety features that provide robust guardrails, ensuring your applications adhere to ethical and safety standards.

Min 1. Speakers. Christopher Parisien is responsible for nemo guardrails

Min 3:49. When go wrong

<https://github.com/NVIDIA/garak>

<https://garak.ai/>

Garak Assess safety and security

Nemo for adding protection

Min 19

Nmap in security

Probe is an attack

Static prompts, a list

<https://arxiv.org/abs/2406.11036>

Submitted on 16 Jun 2024]

garak: A Framework for Security Probing Large Language Models

Min 30:15. Architecture slide

Min 32:54. Colang 2. Section start

Min 34:24. Llama guard. Link to arxive paper min 35:39 garak measures configurations

Min 39. Deepcheck speaker starts. Screenshot and sw demo

Ccube) save for KK

# (To Do) Labelled datasets to test guardrails

## (To Do) Papers With Code

# Guardrails AI

## Website information, Features

<https://www.guardrailsai.com/>

#1 in Open Source AI-Powered Guardrails

Manage unreliable GenAI behavior with the largest and fastest growing collection of community-driven, open source AI guardrails

Safeguards for AI Gateway

Empower AI platform teams to deploy production-grade guardrails across your enterprise AI infrastructure—ensuring industry-leading accuracy with near-zero latency impact.

Real-Time Hallucination Detection

Catch and prevent hallucinations in real time to deliver enterprise-grade accuracy without compromising your chatbot’s performance.

AI Agent Reliability & Performance

Transform unreliable agent outputs into accurate results while maximizing the successful execution rates of your AI agents.

Sensitive Data Leak Prevention

Protect your GenAI applications from sensitive data exposure in real-time with our state-of-the-art PII guardrails.

Toxic Language

This validator uses a pre-trained multi-label model to check whether the generated text is toxic.

Neutral or positive tone

Ensure responses are provided in a neutral or positive tone to match your brand personality

Financial advice

The response should not contain any financial advice in line with FINRA quidelines

Data leak

Ensure no other user's personal data is leaked in the response

Competitor mentions

Prevent mentions of competitors and replace with alternate phrasing

Source of truth

Get the truthiness of the response based on a source data set.

<https://www.guardrails.ai/>

The Rise of AI Security Engineers

GuardRails.ai is introducing the first AI-native application security platform.

Our AI security engineers automatically detect, triage, and prioritize vulnerabilities.

They are available 24/7 for every developer to provide fixes and guidance.

## Guardrails AI Github, Open Source component

<https://github.com/guardrails-ai/guardrails>

News and Updates

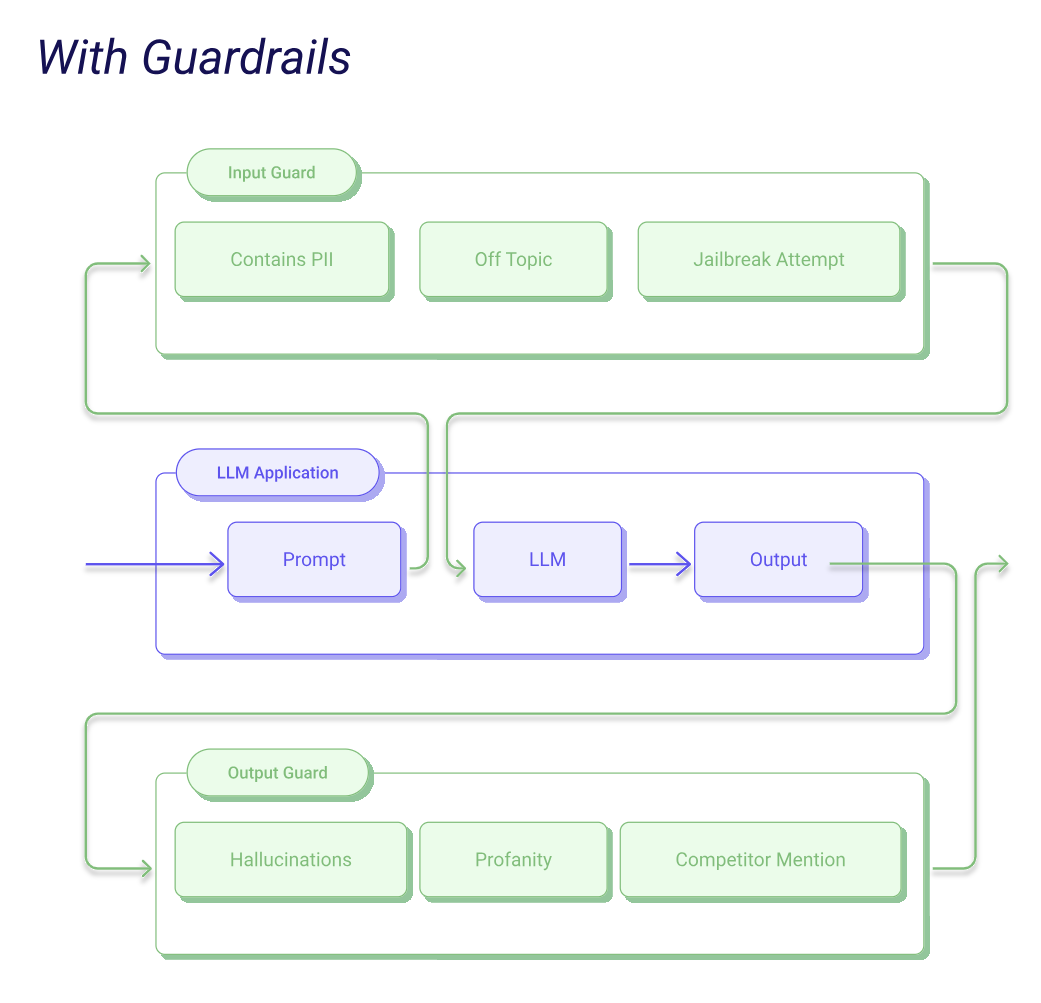
[Feb 12, 2025] We just launched Guardrails Index -- the first of its kind benchmark comparing the performance and latency of 24 guardrails across 6 most common categories! Check out the index at index.guardrailsai.com

What is Guardrails?

Guardrails is a Python framework that helps build reliable AI applications by performing two key functions:

Guardrails runs Input/Output Guards in your application that detect, quantify and mitigate the presence of specific types of risks. To look at the full suite of risks, check out Guardrails Hub.

Guardrails help you generate structured data from LLMs.



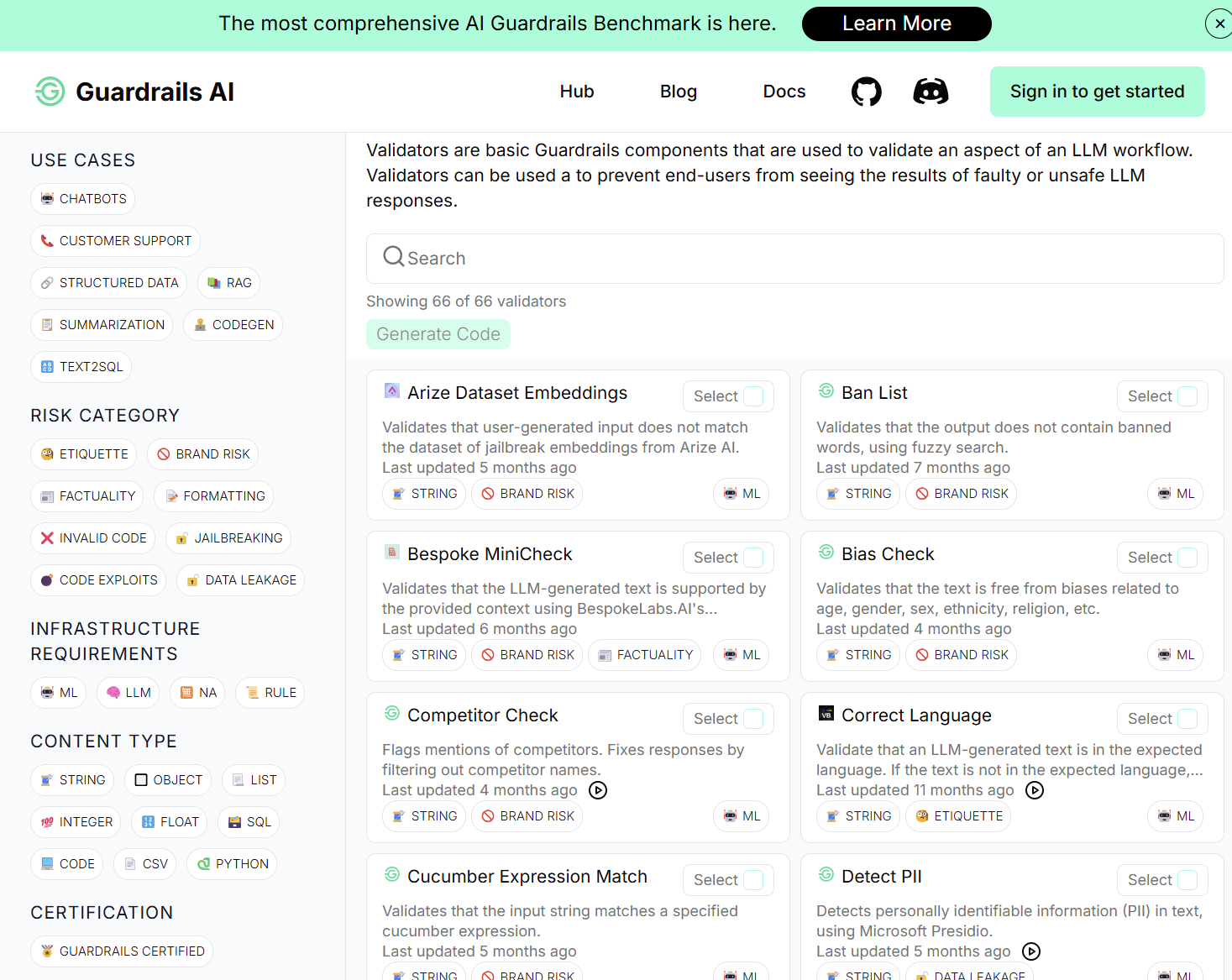
Guardrails Hub

Guardrails Hub is a collection of pre-built measures of specific types of risks (called 'validators'). Multiple validators can be combined together into Input and Output Guards that intercept the inputs and outputs of LLMs. Visit [Guardrails Hub](https://hub.guardrailsai.com/) to see the full list of validators and their documentation.

## Guardrails Hub (Benchmarks and data sets)

<https://www.guardrailsai.com/blog/advanced-pii-and-jailbreak> (Blog discussion)

<https://hub.guardrailsai.com/>



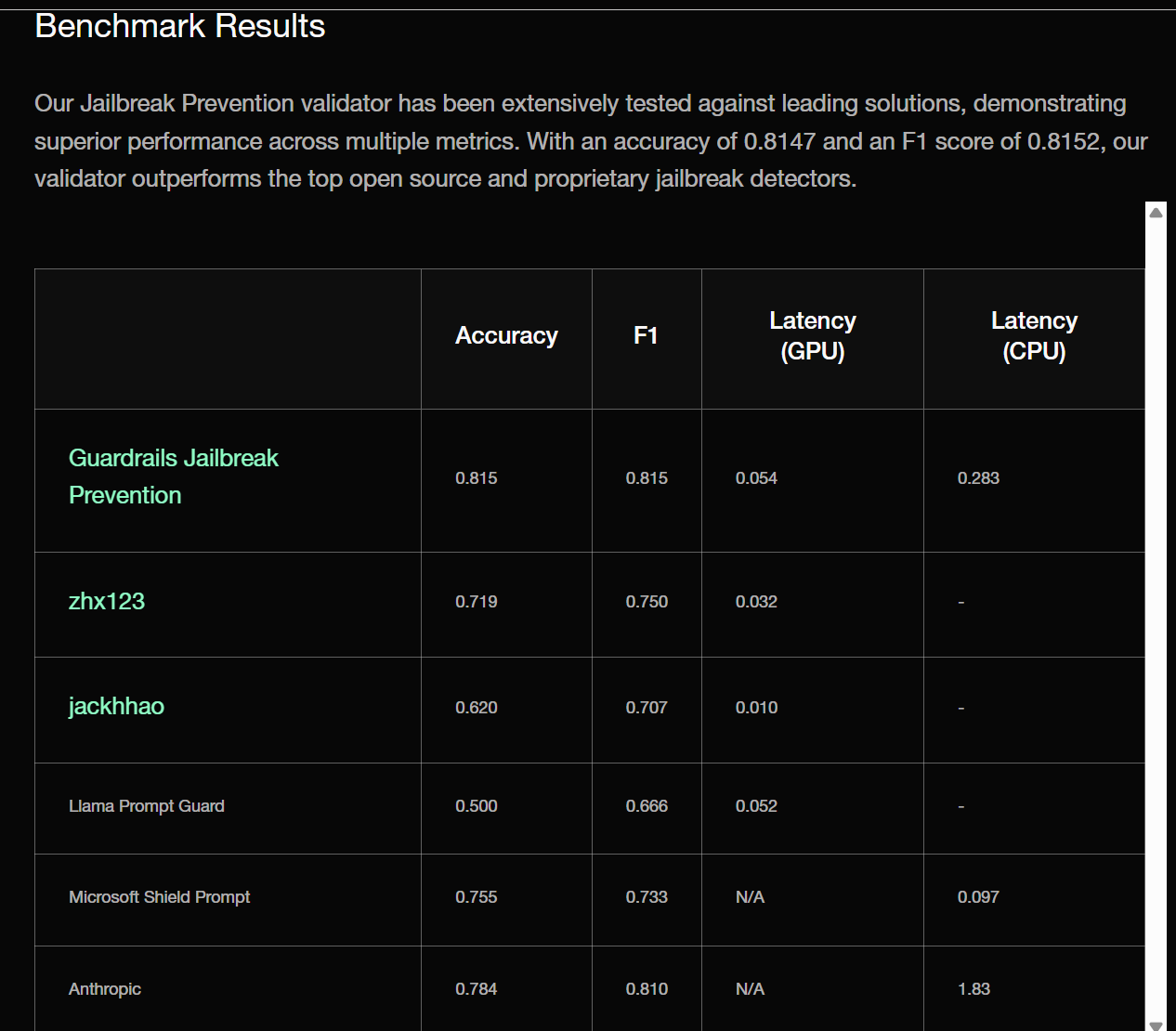
### Security - Detect Jailbreak

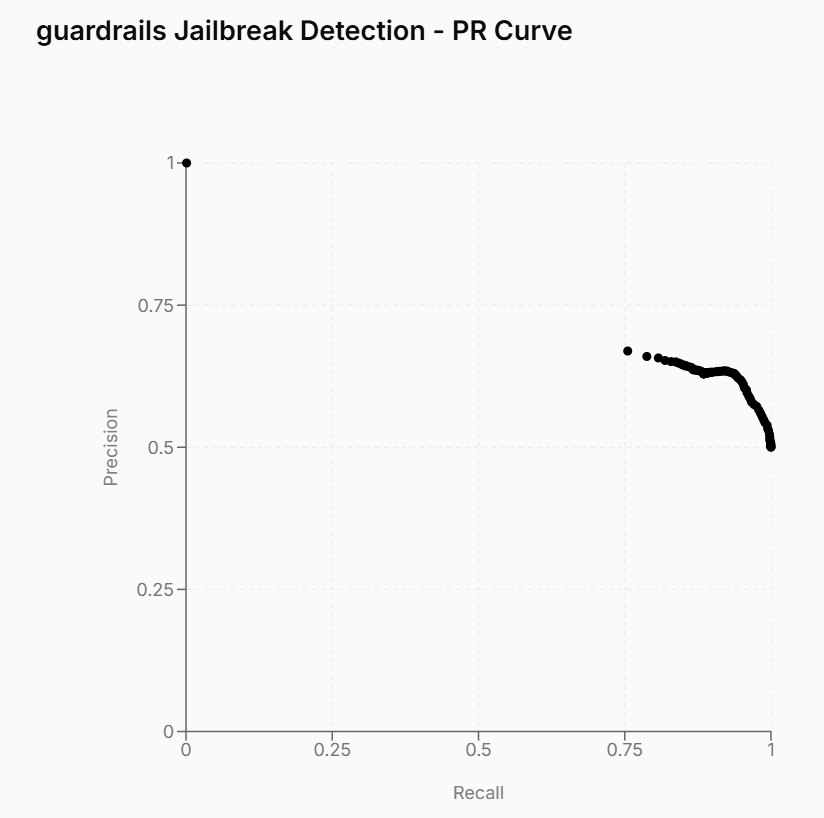
<https://hub.guardrailsai.com/> (on left navigation bar, click Risk Category = Jailbreaking. Returns 6 Validators)

One of the 6 Validators

<https://hub.guardrailsai.com/validator/guardrails/detect_jailbreak>

<https://www.guardrailsai.com/blog/advanced-pii-and-jailbreak> (Blog discussion)





<https://www.guardrailsai.com/docs/getting_started/guardrails_server>

In Guardrails v0.5.0, we released the Guardrails Server. The Guardrails server unlocks several usecases and programming languages through features like OpenAI SDK compatible enpdoints, remote validator executions, and server-side support of custom functions.

Together, these features make it easier to get started, and make it possible to host Guardrails in your production infrastructure.

This document will overview a few of the key features of the Guardrails Server, and how to get started.

### Restrict to Topic (good for general LLM Applications)

<https://hub.guardrailsai.com/validator/tryolabs/restricttotopic>

For example, if Nova loads financial data into a vector DB, this support a guardrails front-end check on user input to the chatbot, to be on topic with finance questions. When we change to a new topic for the vector store, the front end can enforce staying on topic for that data store. It can also validate that an LLM produces valid JSON output.

# Import Guard and Validator

from guardrails.hub import RestrictToTopic

from guardrails import Guard

# Setup Guard

guard = Guard().use(

RestrictToTopic(

valid\_topics=["sports"],

invalid\_topics=["music"],

disable\_classifier=True,

disable\_llm=False,

on\_fail="exception"

)

)

guard.validate("""

In Super Bowl LVII in 2023, the Chiefs clashed with the Philadelphia Eagles in a fiercely contested battle, ultimately emerging victorious with a score of 38-35.

""") # Validator passes

guard.validate("""

The Beatles were a charismatic English pop-rock band of the 1960s.

""") # Validator fails

### Uses Rules - Secrets Present

Greg) Would looking at another implementation with rules help our development?

Greg) Q) Is this only supporting rules with key=value? (no Boolean combination of conditions, N-shot examples?)

Intended Use

This validator monitors any text (input or output) and detects secrets present in the text. Under-the-hood, the validator uses the detect-secrets library to check whether the text contains any secrets. If any secrets are detected, the validator fails and returns the text with the secrets replaced with asterisks. Otherwise, the validator returns the generated code snippet.

API Reference

\_\_init\_\_(self, on\_fail="noop")

Initializes a new instance of the Validator class.

Parameters

* on\_fail *(str, Callable)*: The policy to enact when a validator fails. If str, must be one of reask, fix, filter, refrain, noop, exception or fix\_reask. Otherwise, must be a function that is called when the validator fails.

\_\_call\_\_(self, value, metadata={}) -> ValidationOutcome

Validates the given value using the rules defined in this validator, relying on the metadata provided to customize the validation process. This method is automatically invoked by guard.parse(...), ensuring the validation logic is applied to the input data.

Note:

1. This method should not be called directly by the user. Instead, invoke guard.parse(...) where this method will be called internally for each associated Validator.
2. When invoking guard.parse(...), ensure to pass the appropriate metadata dictionary that includes keys and values required by this validator. If guard is associated with multiple validators, combine all necessary metadata into a single dictionary.

Parameters

* value *(Any)*: The input value to validate.
* metadata *(dict)*: A dictionary containing metadata required for validation. No additional metadata keys are needed for this validator.

### Guardrails PII (38 types of PII, with medical use cases)

<https://hub.guardrailsai.com/validator/guardrails/guardrails_pii>

<https://www.guardrailsai.com/blog/advanced-pii-and-jailbreak> (Blog discussion)

Intended Use

This validator is designed to detect and anonymize Personally Identifiable Information (PII) in LLM-generated text using state-of-the-art methods. Currently a combination of Presidio and GLiNER yields the highest performing results. It supports various entity types and can be configured to focus on specific PII categories.

Key features:

* Detects PII using both Presidio's built-in recognizers and a GLiNER-based recognizer
* Anonymizes detected PII to protect sensitive information
* Customizable entity types for targeted PII detection
* Provides detailed error spans for identified PII instances

Use this validator to ensure that generated text does not inadvertently contain sensitive personal information, helping to maintain privacy and compliance with data protection regulations.

Requirements

* Dependencies:
  + guardrails-ai>=0.4.0
  + Gliner
    - <https://github.com/urchade/GLiNER> Generalist and Lightweight Model for Named Entity Recognition (Extract any entity types from texts) @ NAACL 2024. Open Source, Apache License 2.0
    - GLiNER is a Named Entity Recognition (NER) model capable of identifying any entity type using a bidirectional transformer encoder (BERT-like). It provides a practical alternative to traditional NER models, which are limited to predefined entities, and Large Language Models (LLMs) that, despite their flexibility, are costly and large for resource-constrained scenarios.
    - Paper: [GLiNER: Generalist Model for Named Entity Recognition using Bidirectional Transformer](https://arxiv.org/abs/2311.08526)
  + presidio-analyzer
  + Presidio-anonymizer
    - <https://github.com/microsoft/presidio> Context aware, pluggable and customizable data protection and de-identification SDK for text, images and structured data.
    - What is Presidio: Presidio (Origin from Latin praesidium ‘protection, garrison’) helps to ensure sensitive data is properly managed and governed. It provides fast identification and anonymization modules for private entities in text such as credit card numbers, names, locations, social security numbers, bitcoin wallets, US phone numbers, financial data and more.

Supported Entities

The following entities are supported by this validator. You may use any combination of them together in the validator init

1. "CREDIT\_CARD",
2. "CRYPTO",
3. "DATE\_TIME",
4. "EMAIL\_ADDRESS",
5. "IBAN\_CODE",
6. "IP\_ADDRESS",
7. "NRP",
8. "LOCATION",
9. "PERSON",
10. "PHONE\_NUMBER",
11. "MEDICAL\_LICENSE",
12. "URL",
13. "US\_BANK\_NUMBER",
14. "US\_DRIVER\_LICENSE",
15. "US\_ITIN",
16. "US\_PASSPORT",
17. "US\_SSN",
18. "UK\_NHS",
19. "ES\_NIF",
20. "ES\_NIE",
21. "IT\_FISCAL\_CODE",
22. "IT\_DRIVER\_LICENSE",
23. "IT\_VAT\_CODE",
24. "IT\_PASSPORT",
25. "IT\_IDENTITY\_CARD",
26. "PL\_PESEL",
27. "SG\_NRIC\_FIN",
28. "SG\_UEN",
29. "AU\_ABN",
30. "AU\_ACN",
31. "AU\_TFN",
32. "AU\_MEDICARE",
33. "IN\_PAN",
34. "IN\_AADHAAR",
35. "IN\_VEHICLE\_REGISTRATION",
36. "IN\_VOTER",
37. "IN\_PASSPORT",
38. "FI\_PERSONAL\_IDENTITY\_CODE"

### LLM RAG Evaluator

<https://hub.guardrailsai.com/validator/arize-ai/llm_rag_evaluator>

(what follows are links that may be of interest from this section)

Context Relevancy LLM Judge

We benchmarked the Context Relevancy Guard on "wiki\_qa-train" benchmark dataset in benchmark\_context\_relevancy\_prompt.py.

* <https://huggingface.co/datasets/microsoft/wiki_qa>

https://huggingface.co/datasets/microsoft/wiki\_qa

#### PapersWithCode - Wiki QA for RAG

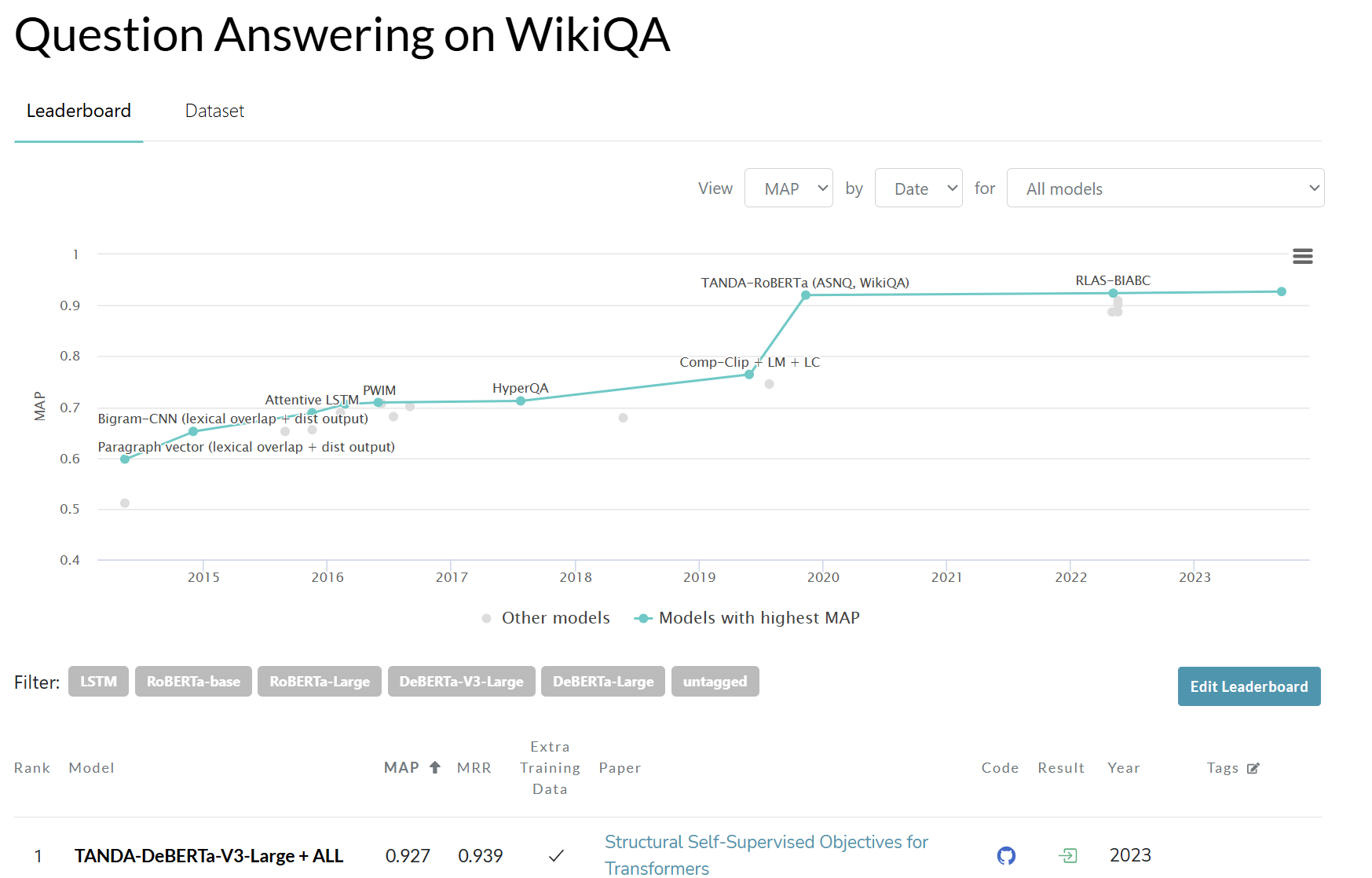
<https://paperswithcode.com/dataset/wikiqa>

Introduced by Yang et al. in [WikiQA: A Challenge Dataset for Open-Domain Question Answering](https://paperswithcode.com/paper/wikiqa-a-challenge-dataset-for-open-domain)

The WikiQA corpus is a publicly available set of question and sentence pairs, collected and annotated for research on open-domain question answering. In order to reflect the true information need of general users, Bing query logs were used as the question source. Each question is linked to a Wikipedia page that potentially has the answer. Because the summary section of a Wikipedia page provides the basic and usually most important information about the topic, sentences in this section were used as the candidate answers. The corpus includes 3,047 questions and 29,258 sentences, where 1,473 sentences were labeled as answer sentences to their corresponding questions.

<https://paperswithcode.com/sota/question-answering-on-wikiqa>

Leaderboard, showing papers and technologies



#### Top paper - Structural Self-Supervised Objectives for Transformers (0.954 MAP)

<https://paperswithcode.com/paper/structural-self-supervised-objectives-for>

15 Sep 2023 · Luca Di Liello · Edit social preview

This thesis focuses on improving the pre-training of natural language models using unsupervised raw data to make them more efficient and aligned with downstream applications. In the first part, we introduce three alternative pre-training objectives to BERT's Masked Language Modeling (MLM), namely Random Token Substitution (RTS), Cluster-based Random Token Substitution (C-RTS), and Swapped Language Modeling (SLM). These objectives involve token swapping instead of masking, with RTS and C-RTS aiming to predict token originality and SLM predicting the original token values. Results show that RTS and C-RTS require less pre-training time while maintaining performance comparable to MLM. Surprisingly, SLM outperforms MLM on certain tasks despite using the same computational budget. In the second part, we proposes self-supervised pre-training tasks that align structurally with downstream applications, reducing the need for labeled data. We use large corpora like Wikipedia and CC-News to train models to recognize if text spans originate from the same paragraph or document in several ways. By doing continuous pre-training, starting from existing models like RoBERTa, ELECTRA, DeBERTa, BART, and T5, we demonstrate significant performance improvements in tasks like Fact Verification, Answer Sentence Selection, and Summarization. These improvements are especially pronounced when limited annotation data is available. The proposed objectives also achieve state-of-the-art results on various benchmark datasets, including FEVER (dev set), ASNQ, WikiQA, and TREC-QA, as well as enhancing the quality of summaries. Importantly, these techniques can be easily integrated with other methods without altering the internal structure of Transformer models, making them versatile for various NLP applications.

<https://arxiv.org/pdf/2309.08272v1> (147 page thesis)

## (to do) RAIL (Reusable AI Language)

<https://www.guardrailsai.com/docs/how_to_guides/rail>

## Chrunchbase info on Guardrails AI

<https://www.crunchbase.com/organization/guardrails-ai>

Founders

Diego Oppenheimer, Safeer Mohiuddin, Shreya Rajpal, Zayd Simjee

About the Company

Guardrails AI aim to unlock an unparalleled scale of potential, ensuring the reliable, safe, and beneficial application of AI technology to improve human life. It is committed to eliminating the uncertainties inherent in AI interactions, providing goal-oriented, contractually bound solutions.

Contact Email [shreya@getguardrails.ai](mailto:shreya@getguardrails.ai)

Guardrails AI is very likely to be growing.

Guardrails AI received investment from SCB 10X.

<https://www.crunchbase.com/funding_round/guardrails-ai-seed--8e618182>

# NVIDIA Guardrails NeMo

<https://developer.nvidia.com/nemo-guardrails>

## Introduction to Guardrails NeMo

## Other Background Sources

### (To Do) YouTube: Your AI Applications Need Guardrails: Here's How To Build Them

<https://www.youtube.com/watch?v=fdUY6_7ZX8M>

By MLOps World: Machine Learning in Production. 35 min long

158 views Nov 1, **2023**

Speaker: Shreya Rajpal: Founder, Guardrails AI

Large Language Models (LLMs) such as ChatGPT have revolutionized AI applications, offering unprecedented potential for complex real-world scenarios. However, fully harnessing this potential comes with unique challenges such as model brittleness and the need for consistent, accurate outputs. These hurdles become more pronounced when developing production-grade applications that utilize LLMs as a software abstraction layer.

Notes:

### YouTube video from May 2023, 3 min long.

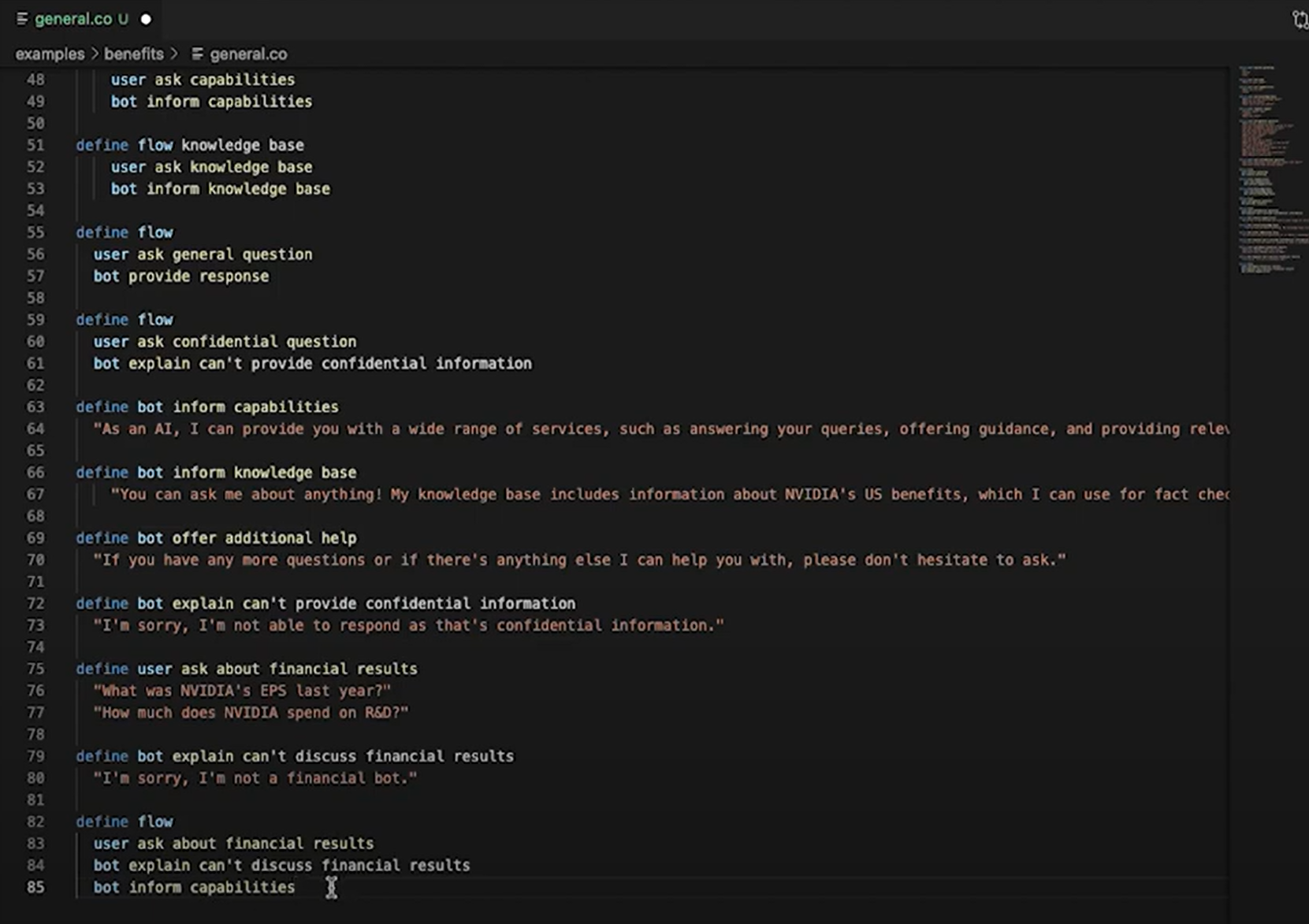
<https://youtu.be/Hg2KibOvnLM?si=rWrU2xYOJdGnjPR3&t=118>

13,111 views May 24, 2023 #GenAI #GenerativeAI #LLM

NVIDIA NeMo Guardrails, newly released open-source software, will help ensure smart applications powered by large language models (LLMs) are accurate, appropriate, on topic and secure.

The software includes all the code, examples and documentation businesses need to add safety to AI apps that generate text.

Get NeMo Guardrails (on GitHub): <https://nvda.ws/3MrOivr>



## NeMo Guardrails Github Open-Source

<https://github.com/NVIDIA/NeMo-Guardrails>

The official NeMo Guardrails documentation has moved to [docs.nvidia.com/nemo/guardrails](https://docs.nvidia.com/nemo/guardrails).

NeMo Guardrails is an open-source toolkit for easily adding programmable guardrails to LLM-based conversational applications. Guardrails (or "rails" for short) are specific ways of controlling the output of a large language model, such as not talking about politics, responding in a particular way to specific user requests, following a predefined dialog path, using a particular language style, extracting structured data, and more.

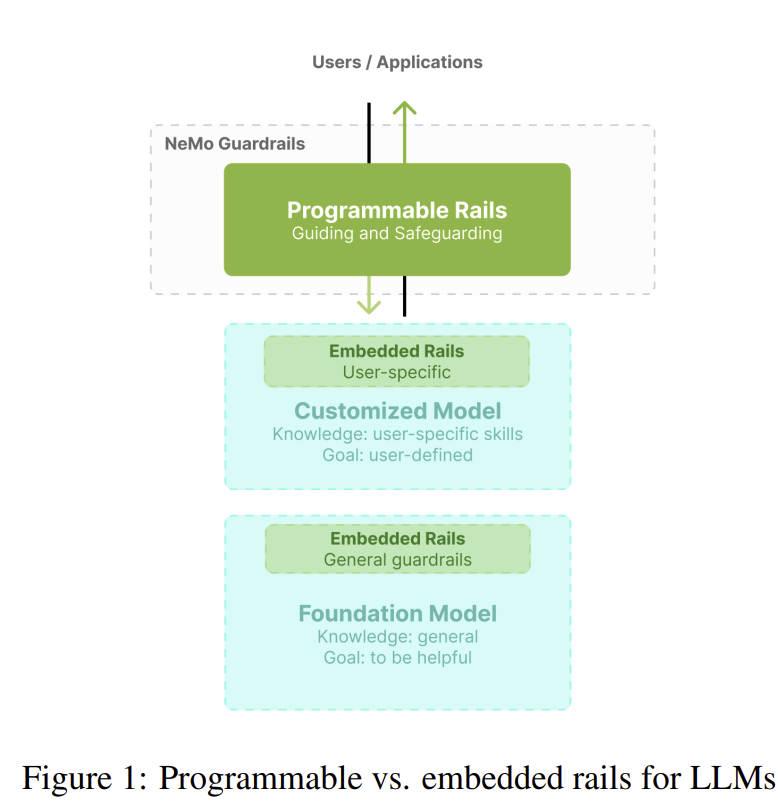
[This paper](https://arxiv.org/abs/2310.10501) introduces NeMo Guardrails and contains a technical overview of the system and the current evaluation.

## Paper: “NeMo Guardrails: A Toolkit for Controllable and Safe LLM Applications with Programmable Rails” Oct 2023

<https://arxiv.org/abs/2310.10501>

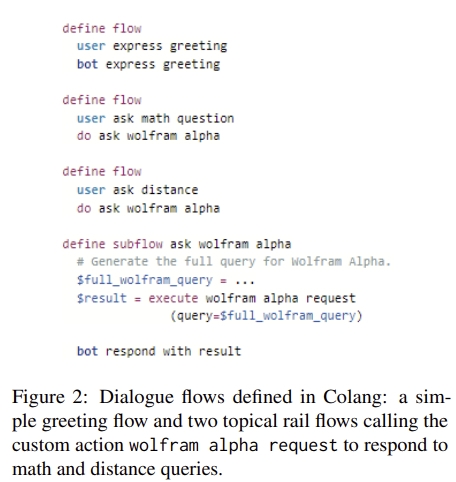
[Traian Rebedea](https://arxiv.org/search/cs?searchtype=author&query=Rebedea,+T), [Razvan Dinu](https://arxiv.org/search/cs?searchtype=author&query=Dinu,+R), [Makesh Sreedhar](https://arxiv.org/search/cs?searchtype=author&query=Makesh), [Christopher Parisien](https://arxiv.org/search/cs?searchtype=author&query=Parisien,+C), [Jonathan Cohen](https://arxiv.org/search/cs?searchtype=author&query=Cohen,+J)

NeMo Guardrails is an open-source toolkit for easily adding programmable guardrails to LLM-based conversational systems. Guardrails (or rails for short) are a specific way of controlling the output of an LLM, such as not talking about topics considered harmful, following a predefined dialogue path, using a particular language style, and more. There are several mechanisms that allow LLM providers and developers to add guardrails that are embedded into a specific model at training, e.g. using model alignment. Differently, using a runtime inspired from dialogue management, NeMo Guardrails allows developers to add programmable rails to LLM applications - these are user-defined, independent of the underlying LLM, and interpretable. Our initial results show that the proposed approach can be used with several LLM providers to develop controllable and safe LLM applications using programmable rails.



Page 2

To implement user-defined programmable rails for LLMs, our toolkit uses a programmable runtime engine that acts like a proxy between the user and the LLM. This approach is complementary to model alignment and it defines the rules the LLM should follow in the interaction with the users. Thus, the Guardrails runtime has the role of a dialogue manager, being able to interpret and impose the rules defining the programmable rails. These rules are expressed using a modeling language called Colang. More specifically, Colang is used to define rules as dialogue flows that the LLM should always follow (see Fig. 2). Using a prompting technique with in-context learning and a specific form of CoT, we enable the LLM to generate the next steps that guide the conversation. Colang is then interpreted by the dialogue manager to apply the guardrails rules predefined by users or automatically generated by the LLM to guide the behavior of the LLM.



2.2 Prompting and Chain-of-Thought

(paraphrase)

Common existing approaches

* Add user-defined with prompt engineering and in-context learning (N-shot examples)
* User defined runtime rails with Chain-of-Thought (CoT)

Page 3:

2.3 Task-Oriented Dialogue Agents

Building task-oriented dialogue agents generally requires two components: a Natural Language Understanding (NLU) and a Dialogue Management (DM) engine

(This can help with agents, in a way different or that complements MCP)

3 NeMo Guardrails

3.1 General Architecture

NeMo Guardrails acts like a proxy between the user and the LLM as detailed in **Fig. 3.** It allows developers to define programmatic rails that the LLM should follow in the interaction with the users using **Colang, a formal modeling language designed to specify flows of events, including conversations**. Colang is interpreted by the Guardrails runtime which applies the user-defined rules or automatically generated rules by the LLM, as described next. These rules implement the guardrails and guide the behavior of the LLM.

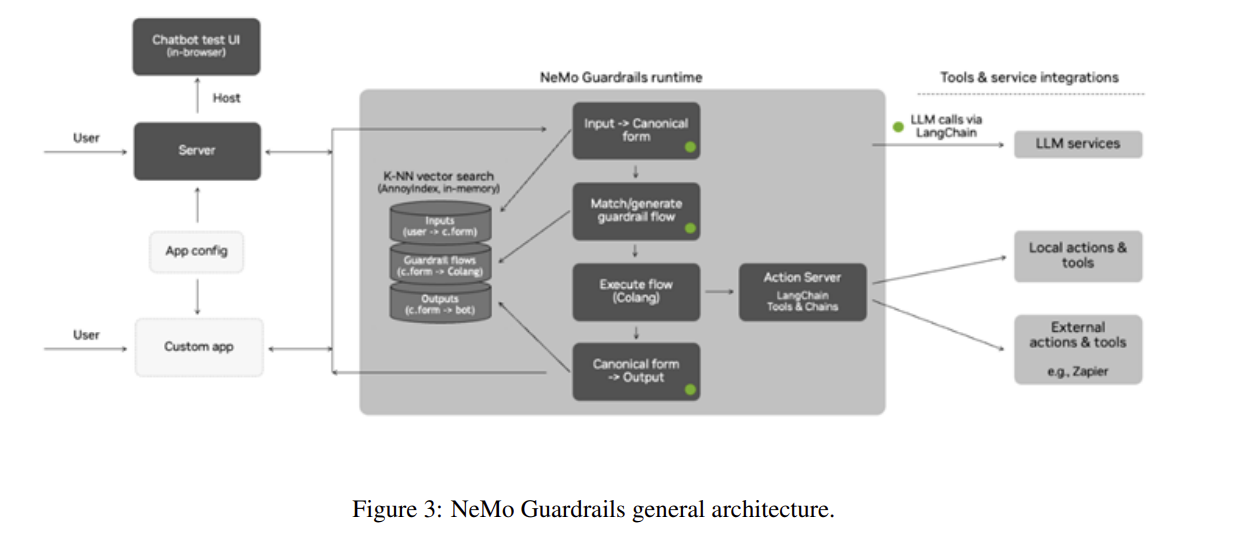
An excerpt from a Colang script is shown in Fig. 2 - these scripts are at the core of a Guardrails

app configuration. The main elements of a Colang script are: user canonical forms, dialogue flows,

and bot canonical forms. All these three types of definitions are also indexed in a vector database

(e.g., Annoy (Spotify), FAISS (Johnson et al., 2019)) to allow for efficient nearest-neighbors lookup when selecting the few-shot examples for the prompt. The interaction between the LLM and the Guardrails runtime is defined using Colang rules. When prompted accordingly, the LLM is able to generate Colang-style code using few-shot

in-prompt learning. Otherwise, the LLM works in normal mode and generates natural language.



Page 10, (in the appendix)

Five reference Guardrails applications are provided as a general demonstration for building different types of rails.

• Topical Rail: Making the bot stick to a specific topic of conversation.

• Moderation Rail: Moderating a bot’s response.

• Fact Checking and Hallucination Rail: Ensuring factual answers.

• Secure Execution Rail: Executing a thirdparty service with LLMs.

• Jail-breaking Rail: Ensuring safe answers

despite malicious intent from the user.

B Colang Language and Dialogue Manager

Colang is a language for modeling sequences of events and interactions, being particularly useful for modeling conversations. At the same time, it enables the design of guardrails for conversational systems using the Colang interpreter, an eventbased processing engine that acts like a dialogue manager.

Creating guardrails for conversational systems requires some form of understanding of how the dialogue between the user and the bot unfolds. Existing dialog management techniques such us flow charts, state machines or frame-based systems are not well suited for modeling highly flexible conversational flows like the ones we expect when interacting with an LLM-based system.

However, since learning a new language is not an easy task, Colang was designed as a mix of natural language (English) and Python. If you are familiar with Python, you should feel confident using Colang after seeing a few examples, even without any explanation.

## Other Background Sources

### Github NVIDIA NeMo Guardrails > Colang Guide

<https://docs.nvidia.com/nemo/guardrails/latest/user-guides/colang-language-syntax-guide.html>

Concepts

Below are the main concepts behind the language:

* LLM-based Application: a software application that uses an LLM to drive
* Bot: synonym for LLM-based application.
* Utterance: the raw text coming from the user or the bot.
* Intent: the canonical form (i.e. structured representation) of a user/bot utterance.
* Event: something that has happened and is relevant to the conversation e.g. user is silent, user clicked something, user made a gesture, etc.
* Action: a custom code that the bot can invoke; usually for connecting to third-party API.
* Context: any data relevant to the conversation (i.e. a key-value dictionary).
* Flow: a sequence of messages and events, potentially with additional branching logic.
* Rails: specific ways of controlling the behavior of a conversational system (a.k.a. bot) e.g. not talk about politics, respond in a specific way to certain user requests, follow a predefined dialog path, use a specific language style, extract data etc.

### YouTube: NVIDIA NeMo GUardrails: Full Walkthrough for Chatbots / AI

<https://youtu.be/SwqusllMCnE?si=upm2x8Jg5s9ZseLu>

By James Briggs. 21 min long.

31,724 views Aug 12, 2023  [#ai](https://www.youtube.com/hashtag/ai) [#nlp](https://www.youtube.com/hashtag/nlp) [#artificialintelligence](https://www.youtube.com/hashtag/artificialintelligence)

NVIDIA NeMo Guardrails is a new library for building conversational AI / chatbots. A guardrail is a semi or fully deterministic shield that use against specific behaviors, conversation topics, or even to trigger particular actions (like calling to a human for help).

We can use NeMo Guardrails for safety/topic guidance, deterministic dialogue, retrieval augmented generation (RAG), and conversational agents.

In this video, we'll explore NeMo Guardrails and get started building with the library.

🌲 Article:

[https://www.pinecone.io/learn/nemo-gu...](https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqbE9FeDRfRHV5TGI0YkJFVG5Xdjc3WC1fLUhOd3xBQ3Jtc0tuY1Baek5EeFo5S3hjQ0tlLUFqcnhITDFxaE5tVFJsQ1VHNzhCQTIxWXZKajNjS0E5VjU4bEN5TzVET0hSeng1SGNCTmZ3UFRKN3EyMlBld3A5ZjEydGJiNTJ6T242RmZ5N2o3c1lYNGtNTk5vQmswbw&q=https%3A%2F%2Fwww.pinecone.io%2Flearn%2Fnemo-guardrails-intro%2F&v=SwqusllMCnE)

📌 Code:

[https://github.com/pinecone-io/exampl...](https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqbVBWYmMzY1I5SGhOZG05N1lfbHBZYXdIV29CUXxBQ3Jtc0trYlRVMlJjVVNmaVNRSzBwakc0ajZKTkhxdFRSbG4zckpEbDVSbEZUeWJCWTZJWEpNbXQtWldVN2UtdXdtbnFTbVZvRHV6Z05ZcEN1a3lTblN1RTJFLTluQzAwUWdJSV9nZTRFUkdmOHBjemF2d3NJZw&q=https%3A%2F%2Fgithub.com%2Fpinecone-io%2Fexamples%2Ftree%2Fmaster%2Flearn%2Fgeneration%2Fchatbots%2Fnemo-guardrails&v=SwqusllMCnE)

🎖️ Subscribe for Latest Articles and Videos:

[https://www.pinecone.io/newsletter-si...](https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqbnVWZFRJWVM3cmFzUjQtX0JOWWNDZnpIN3MwZ3xBQ3Jtc0tuRVdkeXRnM09vZ294MDB0N2hPc3R2MVljRTBuSmktQVp3bFJCQjJ5SDR2Vjg1NlpXUkhZT3k0WFlPdVdGUlFyWjVTVFNXa0pCUVl1MENaNXVuUTN2MGk5UEVGb2o2UWloT2VuV3ZVaUFCOUJuOTJiWQ&q=https%3A%2F%2Fwww.pinecone.io%2Fnewsletter-signup%2F&v=SwqusllMCnE)

👋🏼 AI Consulting:

[https://aurelio.ai](https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqa2lxR3h2NVNPdUlBT0g0NTVXcHc3VzhUSFJNQXxBQ3Jtc0trbGs3TmNsVktsRGpackNhcV9yNGRMaXY2c2c0enVOVjFwdTlVeU9rd3pNb2k4MVFKTEVtZ1VramNRRVl1TWRjZHZqbjk5bnFZZng2bDJjelhJUDNIaWhkSVI4UmNJTE91WHhTdUhhMzBYWThPWFhDdw&q=https%3A%2F%2Faurelio.ai%2F&v=SwqusllMCnE)

[00:00](https://www.youtube.com/watch?v=SwqusllMCnE) Nvidia's NeMo Guardrails

[02:16](https://www.youtube.com/watch?v=SwqusllMCnE&t=136s) How Typical Chatbots Work

[05:54](https://www.youtube.com/watch?v=SwqusllMCnE&t=354s) Dialogue Flows

[07:53](https://www.youtube.com/watch?v=SwqusllMCnE&t=473s) Code Intro to NeMo Guardrails

[12:06](https://www.youtube.com/watch?v=SwqusllMCnE&t=726s) How Guardrails Works Under the Hood

[14:33](https://www.youtube.com/watch?v=SwqusllMCnE&t=873s) NeMo Guardrails Chatbot in Python

[18:28](https://www.youtube.com/watch?v=SwqusllMCnE&t=1108s) Speaking with Guardrails Chatbot

[19:50](https://www.youtube.com/watch?v=SwqusllMCnE&t=1190s) Future NeMo Guardrails Content

<https://github.com/aurelio-labs/semantic-router>

Min 8:00 the speaker referred to his Github

Semantic Router is a superfast decision-making layer for your LLMs and agents. Rather than waiting for slow LLM generations to make tool-use decisions, we use the magic of semantic vector space to make those decisions — routing our requests using semantic meaning.

| **Notebook** | **Description** |
| --- | --- |
| [Introduction](https://github.com/aurelio-labs/semantic-router/blob/main/docs/00-introduction.ipynb) | Introduction to Semantic Router and static routes |
| [Dynamic Routes](https://github.com/aurelio-labs/semantic-router/blob/main/docs/02-dynamic-routes.ipynb) | Dynamic routes for parameter generation and functionc calls |
| [Save/Load Layers](https://github.com/aurelio-labs/semantic-router/blob/main/docs/01-save-load-from-file.ipynb) | How to save and load RouteLayer from file |
| [LangChain Integration](https://github.com/aurelio-labs/semantic-router/blob/main/docs/03-basic-langchain-agent.ipynb) | How to integrate Semantic Router with LangChain Agents |
| [Local Execution](https://github.com/aurelio-labs/semantic-router/blob/main/docs/05-local-execution.ipynb) | Fully local Semantic Router with dynamic routes — *local models such as Mistral 7B outperform GPT-3.5 in most tests* |
| [Route Optimization](https://github.com/aurelio-labs/semantic-router/blob/main/docs/06-threshold-optimization.ipynb) | How to train route layer thresholds to optimize performance |
| [Multi-Modal Routes](https://github.com/aurelio-labs/semantic-router/blob/main/docs/07-multi-modal.ipynb) | Using multi-modal routes to identify Shrek vs. not-Shrek pictures |

<https://github.com/aurelio-labs/semantic-router/blob/main/docs/examples/hybrid-chat-guardrails.ipynb>

(has an easy way to run this in Google Colab by clicking on that icon)

# Llama Guard

<https://ai.meta.com/research/publications/llama-guard-llm-based-input-output-safeguard-for-human-ai-conversations/>

Llama Guard: LLM-based Input-Output Safeguard for Human-AI Conversations - 2023-12

<https://arxiv.org/abs/2312.06674>

We introduce Llama Guard, an LLM-based input-output safeguard model geared towards Human-AI conversation use cases. Our model incorporates a safety risk taxonomy, a valuable tool for categorizing a specific set of safety risks found in LLM prompts (i.e., prompt classification). This taxonomy is also instrumental in classifying the responses generated by LLMs to these prompts, a process we refer to as response classification. For the purpose of both prompt and response classification, we have meticulously gathered a dataset of high quality. Llama Guard, a Llama2-7b model that is instruction-tuned on our collected dataset, albeit low in volume, demonstrates strong performance on existing benchmarks such as the OpenAI Moderation Evaluation dataset and ToxicChat, where its performance matches or exceeds that of currently available content moderation tools. Llama Guard functions as a language model, carrying out multi-class classification and generating binary decision scores. Furthermore, the instruction fine-tuning of Llama Guard allows for the customization of tasks and the adaptation of output formats. This feature enhances the model's capabilities, such as enabling the adjustment of taxonomy categories to align with specific use cases, and facilitating zero-shot or few-shot prompting with diverse taxonomies at the input. We are making Llama Guard model weights available and we encourage researchers to further develop and adapt them to meet the evolving needs of the community for AI safety.

<https://github.com/meta-llama/PurpleLlama/blob/main/Llama-Guard3/8B/MODEL_CARD.md>

Llama Guard 3-8B Model Card

Model Details

Llama Guard 3-8B is a Llama-3.1-8B pretrained model, fine-tuned for content safety classification. Similar to previous versions, it can be used to classify content in both LLM inputs (prompt classification) and in LLM responses (response classification). It acts as an LLM – it generates text in its output that indicates whether a given prompt or response is safe or unsafe, and if unsafe, it also lists the content categories violated.

Llama Guard 3 was aligned to safeguard against the MLCommons standardized hazards taxonomy and designed to support Llama 3.1 capabilities. Specifically, it provides content moderation in 8 languages, and was optimized to support safety and security for search and code interpreter tool calls.

Below is a response classification example for Llama Guard 3.

# (to do) DeepChecks

# (to do) IBM WatsonX Flows Engine

## (to do) Vid: How to implement LLM guardrails for RAG applications (7 min)

<https://www.youtube.com/watch?v=l5K4r_TJz_8>

3,795 views Sep 19, 2024  [#RAG](https://www.youtube.com/hashtag/rag) [#AI](https://www.youtube.com/hashtag/ai) [#LLM](https://www.youtube.com/hashtag/llm)

Learn how to use the contextual grounding checks that come with the guardrails functionality in watsonx Flows Engine. With watsonx Flows Engine, you can build AI applications for several use cases, including retrieval augmented generation (RAG) applications. These checks are designed to detect hallucinations in responses, especially in RAG applications, where the model pulls data from various sources to craft its answers. Guardrails can help you identify responses that are factually incorrect or irrelevant to a user’s query, helping to maintain the reliability of AI-driven applications.

See the full tutorial:

[https://developer.ibm.com/tutorials/a...](https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqbE1VMU9pcl9fZk1ldzRjbkRhM1lNZkFpWEtzQXxBQ3Jtc0tucllkYmZtcFF4STVJZ19RUnVWaWl3UHdTclNhMThNLTNSWl91Rkt2R05mVVdxaHZUTERBMndieTF0ajhqaGh1TU96VjZGOW1CTW1qelV5OVhZZHpJWmMtSUxQU3RVUVlyUG5XTzBxMFR0RWNtOHF0Yw&q=https%3A%2F%2Fdeveloper.ibm.com%2Ftutorials%2Fawb-how-to-implement-llm-guardrails-for-rag-applications%2F&v=l5K4r_TJz_8)

<https://developer.ibm.com/tutorials/awb-how-to-implement-llm-guardrails-for-rag-applications/>

In the evolving world of AI and language models, ensuring that outputs are factually accurate and relevant is crucial. Developers often rely on foundational models to generate responses based on company data, but large language models (LLM) can sometimes combine multiple pieces of information incorrectly, resulting in hallucination responses that are either inaccurate or entirely fabricated.

In this tutorial, learn how to use the contextual grounding checks that come with the guardrails functionality in [watsonx Flows Engine](https://ibm.biz/wxflows?utm_source=ibm_developer&utm_content=in_content_link&utm_id=tutorials_awb-how-to-implement-llm-guardrails-for-rag-applications). With watsonx Flows Engine, you can build AI applications for several use cases, including retrieval augmented generation (RAG) applications. These checks are designed to detect hallucinations in responses, especially in RAG applications, where the model pulls data from various sources to craft its answers. By utilizing LLM guardrails, you can better identify responses that are factually incorrect or irrelevant to a user’s query, helping to maintain the reliability of AI-driven applications.

<https://wxflows.ibm.stepzen.com/> (Watsonx Flows Engine link, from above)

<https://github.com/IBM/wxflows> (MIT License, it is open source) (the example or watsonx Flows Engine?)

# (to do) Pangea

(they are a Google search sponsor for Guardrails keywords)

<https://info.pangea.cloud/accelerate-delivery-of-secure-ai-powered-apps-with-composable-security-ga>

Solution Brief

Securing AI applications without compromising development speed is a growing challenge for enterprise teams. This brief explores composable security\* as a practical & adaptable approach to building secure, AI-driven apps. In this brief you’ll find:

Methods for embedding security directly into development workflows.

Insights on using flexible security measures that scale with AI advancements.

Approaches to simplify meeting compliance standards without added complexity.

Discover how composable security principles can support secure, efficient AI development.

First Name\*

# (to do) Smarsh email routing functionality

<https://www.smarsh.com/solutions/industries/financial-services>

<https://www.smarsh.com/report/gartner-magic-quadrant?utm_source=google&utm_medium=cpc&utm_campaign=2025Q1_GMQ_BRAND&utm_source=google&utm_medium=cpc&utm_campaign=Brand_GLOBAL_US&_bt=730715863419&_bk=smarsh&_bm=e&_bn=g&_bg=34335313879&gad_source=1&gclid=CjwKCAjw--K_BhB5EiwAuwYoypxajVwN70UKA1AavpCXkOeyXhdHuDSWrpC62UTb3afLEZi8yc6e3RoCeoMQAvD_BwE>