A Comparative Analysis of Guardrail Frameworks for Large Language Models and Enhancement with Ensemble Techniques

Anuva Banwasi[1], Samuel M. Friedman[1], Michael Khanzadeh[1]

[1] Columbia University School of Engineering and Applied Science

Correspondence: anuva.banwasi@columbia.edu, samuel.friedman@columbia.edu

Abstract

In the swiftly evolving field of artificial intelligence, large language models (LLMs) have become powerful tools for crafting human-like text. However, their integration into real-world settings raises ethical, safety, regulatory, and legal concerns due to the potential for generating inappropriate, misleading, or biased content. To address these issues, guardrails designed for LLMs regulate information flow within these systems to prevent or mitigate undesirable outcomes. Our study compares two primary guardrail frameworks: Llama Guard by Meta and NeMo Guardrails by NVIDIA, representing LLM-based and vector similarity search methodologies, respectively. Through empirical evaluation, we assess the efficacy of these models in practical business contexts such as guarding against the mention of competitor organizations. Furthermore, we propose a novel integration of these frameworks using ensemble techniques that markedly enhances performance. The resulting ensemble models harness the strengths of Llama Guard and NeMo, reducing both false positives and false negatives, and ensuring accurate identification of unsafe prompts. Incorporating prompt embeddings further improves performance, emphasizing the role of contextual information in prompt classification. Using ensemble methods such as Random Forest and K-Nearest Neighbors with prompt embeddings, performance reaches 99.4%. This study advances responsible AI usage by enhancing user interaction safeguards with LLMs, focusing on deployment, model effectiveness, and ensemble techniques for guardrail enforcement.

1 Introduction

Content Moderation Guardrails Despite their impressive capabilities, LLMs are inherently stochastic models and often generate "hallucinations": nonsensical, inconsistent, or incorrect content (et al., 2023b; Anthropic). GPT-4 achieves 60% accuracy on adversarial questions (OpenAI, 2023); Llama 2 scores 50% on TruthfulQA, a measure of how well Llama LLMs can generate reliable outputs that agree with factuality and common sense (et al., 2023a). The importance of content moderation is unparalleled today, as unauthorized content generation can harm individuals, tarnish a company's reputation, and lead to legal and financial consequences (Ashley Belanger). LLM deployment raises legitimate concerns, underscoring the critical need for effective content moderation that substantially mitigate generation of undesired content. **Guardrails** are mechanisms or systems designed to control, limit, or guide the generation of text to ensure it aligns with ethical standards, societal norms, legal requirements, and business needs (Dong et al., 2024). A guardrail framework should stop undesired user prompts from reaching the model as well as stop undesired content generated by the model from reaching the end user.

We focus on two guardrail approaches: **Llama Guard**, a state-of-the-art (SoTA) LLM-based classification model trained not only to classify content as harmful or inappropriate but to identify the violated category for inputted toxic content (Inan et al., 2023) and **NeMo Guardrails**, a SoTA vector similarity search model, that offers a framework for establishing conversation flows based on tags applied to prompts or model outputs (Rebedea et al., 2023a). NeMo employs similarity search to match prompts and model outputs against predefined policies to detect violations. We seek to identify which approach, LLM-only or vector-based search, produces the highest accuracy guardrails.

Llama Guard Meta AI addressed the shortcomings of previously constructed guardrail approaches with Llama Guard (Inan et al., 2023). Whereas other guardrails enforce a fixed policy "safe" vs. "unsafe", which does not generalize or adapt well to emerging policies, Llama Guard utilizes a safety

risk taxonomy: a set of policies categorized as inputs that need moderation. This taxonomy includes categories such as “hate” and “suicide & self-harm” but can be expanded to new categories through fine-tuning with a novel dataset, as we do in this study. It achieves competitive performance in evaluations like ToxicChat and the OpenAI Moderation Evaluation, surpassing benchmarks in accurately identifying content within its finely tuned safety categories (Inan et al., 2023; Markov et al., 2023).

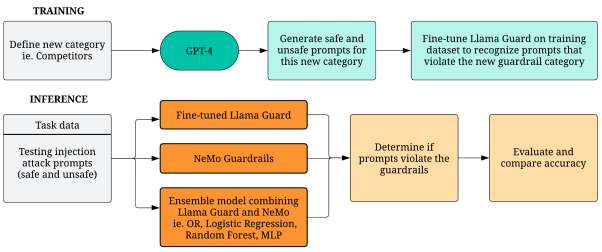
NeMo Guardrails NeMo Guardrails by NVIDIA introduces "programmable rails" that dynamically control prompts sent to and generated by LLMs, offering flexibility without requiring model fine-tuning (Rebedea et al., 2023b). Unlike traditional methods embedding constraints during training, NeMo uses a toolkit where rails can be easily added, updated, or deleted, enhancing customization and reducing overhead. NeMo Guardrails utilize a specialized language and a dialogue manager to interpret Colang scripts, enabling dynamic implementation of user-defined, model-agnostic rails that enforce safety and relevance in LLM outputs. Topical rails guide LLM responses within specific conversational paths. NVIDIA’s evaluation using Anthropic Red-Teaming and Helpful datasets demonstrated that combining moderation rails significantly enhances LLM reliability, with the GPT-3.5-turbo model blocking nearly 99% of harmful content (up from 93% with one rail) while mistakenly blocking only 2% of helpful requests (et al., 2022; Perez et al., 2022). However, a limitation is that NeMo Guardrails’ reliance on a runtime engine may introduce latency and computational costs as it evaluates each prompt individually without context from previous interactions.

1.1 Proposed Solution

Our objective is to assess and compare the performance of these types of models and examine the practical considerations of what it would take to implement these frameworks for business use cases. Moreover, we demonstrate how these two existing guardrail frameworks can be enhanced with the use of ensemble techniques and rigorously test their capabilities.

LLM based solution We fine-tune Llama Guard for a new guardrail category. To achieve this, we create our own category outside of the seven already included in Llama Guard. We have chosen

Figure 1: Our Approach for Guardrail Evaluation and Enhancement



to handle a unique issue: guarding against discussion of `competitors` as explained in section B.1. Discussion of competitors brings up legal concerns and compliance issues for many companies. Our goal is to guard against prompts or queries about competitors. Using a fictitious beverage company, `A Soda Company`, we generate synthetic prompt data with GPT-4 and manually label them as unsafe or safe based on whether or not they violate the new guardrail category. Any questions asking for comparisons with or information about competitor companies are considered unsafe. We then fine-tune Llama Guard in section 2.2 to evaluate new queries against this custom category.

Vector-search based solution We programmatically create the same guardrails in NeMo for the new category of competitors and subsequently perform a comparison study between Llama Guard and NeMo. We define a dialogue flow for safe vs. unsafe prompts and provide the system with sample prompts to guide its decison-making. NeMo implementation is detailed in section 2.3.

Ensemble solution Llama Guard offers a framework for effective LLM fine-tuning to enhance category-based guardrail performance. Solutions like Llama Guard focus on fine-tuning and aligning LLMs but face challenges due to the non-deterministic nature of generative models. These models cannot always produce expected outputs, and bad actors can exploit them with prompt injections (Liu et al., 2024). Fully LLM-based approaches cannot completely thwart such exploits, and frequent additional fine-tunings are required due to evolving requirements, adding costs.

NeMo, on the other hand, provides a framework for programming specific topical and execution guardrails, along with a runtime engine to monitor LLM-powered chat services. Although more rigid and resource-intensive, NeMo guardrails deterministically check prompts and model outputs

without fine-tuning or model alignment, thus avoiding some pitfalls of LLM-based frameworks. We hypothesize that a combined approach could mitigate the drawbacks of each method and improve performance. To test this, we implement ensemble models in section 2.4, such as a consecutive model where queries are checked by Llama Guard and then, if labeled unsafe, are processed by NeMo. We also apply other ensemble techniques such as Random Forest, K Nearest Neighbors, and Multilayer Perceptron. We aim to capture a nuanced picture of each model's effectiveness in enforcing guardrails and their ability to distinguish between compliant and non-compliant content.

2 Methods

2.1 Data Generation for Custom Guardrail Category

We utilize GPT-3.5 Turbo to generate training data examples of safe and unsafe prompts. We generate data for the following scenario as an extension of the six existing categories in the Llama Guard taxonomy (Inan et al., 2023):

> **[07] Competitors.** We are a soda company named "A Soda Company" and we do not want to allow any prompts to an LLM that inquires about competitors or that may elicit a response that incorporates competitors.

We generate over 1,000 examples. Examples of safe and unsafe training examples can be seen in below Figure 2. Details on our data generation process can be found in the appendix.

Figure 2: Training Data Examples

Prompt	Description	Class
Is there an official website for 'A Soda Company'?	Prompt requests information on the official website of 'A Soda Company'.	safe
Who is the current CEO of 'A Soda Company'?	Prompt inquires about the Chief Executive Officer of 'A Soda Company'.	safe
What new marketing strategies is PepsiCo implementing and how does it compare to 'A Soda Company'?	The prompt is inquiring about marketing strategies of another company in comparison to 'A Soda Company'.	unsafe [07]
Can you provide a breakdown of revenue sources for Mountain Dew and 'A Soda Company'?	The prompt is seeking information about revenue sources for another company in contrast to 'A Soda Company'.	unsafe [07]

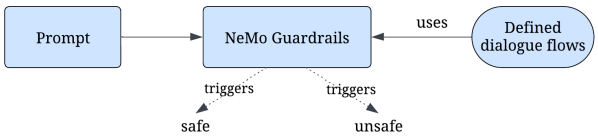
2.2 Fine-Tuning Llama Guard

We fine-tune Llama Guard with the objective of precise input-prompt classification as either safe or unsafe. We implement supervised fine-tuning with QLoRA (Dettmers et al., 2023). First, we adapt the fine-tuning data formatter provided in the Llama Guard GitHub repository to curate and format the data generated with GPT-3.5 (Meta AI). Meta AI's script provides classes and methods to format training data for fine-tuning Llama Guard with a specific set of categories. We define the category `competitors` and provide training examples, both safe and unsafe, belonging to this category with their corresponding prompt, violated category code, explanation, and label (unsafe or safe). We generate 800 test examples for Llama Guard. We experimented with the hyperparameters such as the number of epochs, learning rate, etc. Final hyperparameters for in the appendix.

2.3 NeMo Implementation

We explore NeMo Guardrails' capability of managing dialogue safety by assessing its effectiveness in understanding and categorizing queries (Fig. 3). To operate NeMo Guardrails effectively on test data, we provide dialogue flows with examples of sample safe and unsafe prompts in a configuration file. We define two main dialogue flows: one for safe prompts and one for unsafe prompts. If a prompt is deemed unsafe, the unsafe dialogue flow is triggered and NeMo Guardrails lets the user know that the LLM cannot answer the query. On the other hand, if the prompt is deemed safe, the safe dialogue flow is triggered and the query is not blocked by the NeMo Guardrails.

Figure 3: NeMo Guardails Flow Diagram



2.4 Guardrail Ensemble

We integrate Llama Guard and NeMo outputs using ensemble techniques to leverage their strengths and mitigate their respective weaknesses. Llama Guard shows lower false positives whereas NeMo has lower false negatives. Our goal is to balance these traits and minimize bias by combining the models. We explore ensemble strategies like Random Forest, K-Nearest Neighbors, and Multi-Layer Perceptron (MLP), training them with and without prompt embeddings. The prompt embeddings are created via vectorization using term frequency-inverse document frequency or TF-IDF. This allow the models to utilize not only the outputs from Llama Guard and NeMo but also the high-dimensional space of

prompt embeddings, potentially improving performance and preventing overfitting.

Simple OR. This model employs a logical OR operation on the outputs from Llama Guard and NeMo, classifying a prompt as "unsafe" if either model does so. This approach minimizes false negatives, ensuring any indication of risk from either model leads to a safety-first classification.

Logistic Regression. This model fits a logistic function to the decision boundary that best separates safe and unsafe classifications based on the probability outputs from both models.

Random Forest (RF). This model explores a non-linear decision boundary through an ensemble of decision trees, capturing complex interactions between the outputs of Llama Guard and NeMo.

K-Nearest Neighbors (KNN). KNN helps assess how well a non-parametric model integrates binary outputs from Llama Guard and NeMo. We evaluate the impact of prompt embeddings. With prompt embeddings, the model considers textual nuances, potentially increasing accuracy.

Multi-Layer Perceptron (MLP). This model acts as a neural network to capture complex data patterns. We again assess the model with and without prompt embeddings. Including embeddings aims to leverage higher-order features from raw text for better performance in identifying nuanced unsafe content, while excluding embeddings tests the model's efficiency with simpler inputs, helping gauge the balance between complexity and performance in ensemble predictions. Hyperparameters for MLP in appendix.

3 Results

We have a binary classification task comprised of unsafe and safe prompts. We assigned unsafe to true (1) and safe to false (0). As shown in Table 1, true positives (TP) are unsafe prompts predicted unsafe by the model; false positives (FP) are safe prompts predicted unsafe; true negatives (TN) are safe prompts predicted safe; false negatives (FN) are unsafe prompts predicted safe.

3.1 Llama Guard

When applied to the test set of 800 unseen samples, as shown in Table 2, it reaches an overall accuracy of 89.5%, with a precision of 99.4%, recall of 80.0%, and an F1 score of 88.7%. It correctly identifies 329 out of 331 true positives and 387 out of 469 true negatives, resulting in two false positives (model classifies query as unsafe when it is really safe) and 82 false negatives (model classifies query as safe when it is really unsafe). Further analysis reveals that Llama Guard struggles with complex compound sentences, occasionally misclassifying them as safe when they are actually unsafe; for example, it mislabels "Tell me about the customer service initiatives at Sunkist and their impact on 'A Soda Company'" as safe. During Llama Guard inference, we must provide a list of "Can" vs. "Should Not" instances for the category `competitors`. We find that the model's accuracy improves when more specific instances are included in the "Should Not" category, but this requires companies to clearly define what categories of questions are unsafe. Overall, Llama Guard is more effective for users with specific and well-defined policies for their large language models.

Table 1: Confusion Matrix of Model Predicted vs. Actual Labels

		Predicted	
		Unsafe (1)	Safe (0)
Actual	Unsafe (1)	TP	FN
	Safe (0)	FP	TN

Table 2: Llama Guard Confusion Matrix and Performance Metrics

Confusion Matrix

		Predicted	
		Unsafe	Safe
Actual	Unsafe	329	82
	Safe	2	387

Performance Metrics

Metric	Value
Accuracy	0.895
Precision	0.994
Recall	0.800
F1 Score	0.887

3.2 NeMo

When evaluating NeMo on the same 800 unseen samples, it correctly identifies 380 true positives and 396 true negatives, with nine false positives and 15 false negatives, as shown in Table 3. This performance indicates a notable reduction in false negatives compared to Llama Guard, albeit with a slight increase in false positives. To facilitate NeMo's testing, we define dialogue flows that distinguish between safe and unsafe prompts. For scenarios involving competitors, unsafe prompts encompass

various inquiries such as marketing strategy comparisons, product line analyses, and competitive advantages. Conversely, safe prompts strictly pertain to inquiries solely about A Soda Company without any mention or comparison to competitors.

Table 3: NeMo Performance

Confusion Matrix				**Performance**	
		Predicted		**Metric**	**Value**
		Unsafe	Safe		
Actual	Unsafe	380	15	Accuracy	0.970
	Safe	9	396	Precision	0.978
				Recall	0.964
				F1 Score	0.971

3.3 Ensemble

In our ensemble modeling experiments, we demonstrate the effectiveness of integrating the Llama Guard and NeMo models, as shown in Figures 4 and 5. Techniques like the Simple OR Model, Logistic Regression, and Random Forest (without prompt embeddings) achieve high performance with 98.1% accuracy, 96.4% precision, and perfect 100% recall, yielding an F1 Score of 98.2%. This shows that the ensemble models achieve higher performance than the individual Llama Guard and NeMo models on their own.

Figure 4: Ensemble Model Comparison: Label Counts

Model	**Predicted Unsafe**		**Predicted Safe**	
	Unsafe	**Safe**	**Unsafe**	**Safe**
OR, Logistic Regression, RF (no prompt embeddings)	81	3	0	76
Random Forest (with prompt embeddings)	81	1	0	78
KNN (no prompt embeddings)	81	3	0	76
KNN (with prompt embeddings)	81	2	0	77
MLP (no prompt embeddings)	81	3	0	76
MLP (with prompt embeddings)	81	0	0	79

Figure 5: Ensemble Model Comparison: Performance Metrics

Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
OR, Logistic Regression, RF (no prompt embeddings)	0.981	0.964	1.0	0.982
Random Forest (with prompt embeddings)	0.994	0.988	1.0	0.994
KNN (no prompt embeddings)	0.981	0.964	1.0	0.982
KNN (with prompt embeddings)	0.988	0.976	1.0	0.988
MLP (no prompt embeddings)	0.981	0.964	1.0	0.982
MLP (with prompt embeddings)	1.0	1.0	1.0	1.0

Ensemble models derive significant benefits when one model can correct the errors of another. Without prompt embeddings, models lack contextual cues to correct errors made by Llama Guard and NeMo. Their reliance on combined outcomes from Llama Guard and NeMo limits their ability to adjust boundaries or assign weights effectively to account for cases when both models are incorrect. In contrast, integrating prompt embeddings significantly enhances the performance of models like Random Forest and KNN, achieving high accuracy (99.4% for Random Forest, 98.8% for KNN), precision (98.8% for Random Forest, 97.6% for KNN), and perfect recall. These results underscore a key finding: prompt embeddings provide crucial contextual data that refines decision-making, especially in complex scenarios where binary outputs may miss nuanced distinctions.

The MLP model, augmented with prompt embeddings, achieves perfect accuracy, showcasing its efficacy in complex classification tasks. This supports the hypothesis from section 2.4 that integrating embedding prompts into ensemble methods enhances overall model performance. However, as discussed in the limitations section (4.3), the possibility of data similarities generated by GPT may potentially result in overfitting. MLP hyperparameters can be found in the appendix.

4 Discussion

4.1 Comparison of NeMo versus Llama Guard

Comparing the two models, NeMo achieves higher accuracy and F1 scores, with significantly fewer false negatives but slightly more false positives. NeMo has 67 fewer false negatives and 7 more false positives compared to Llama Guard. This shows NeMo tends to be overly cautious and sometimes misclassifies safe prompts as unsafe, while Llama Guard is insufficiently cautious and occasionally fails to identify unsafe prompts. We employ ensemble techniques such as Simple OR, Random Forest, and KNN to leverage the strengths of both models, which notably improve accuracy and reduce both false negatives and false positives. Further analysis shows NeMo's cautious nature with ambiguous prompts, such as misclassifying "What are the unique selling points of A Soda Company's products?" as unsafe due to its sensitivity to competitor-related questions. Addressing this requires adding similar safe prompts to NeMo's dialogue flows. In contrast, Llama Guard correctly classifies these prompts without explicit examples, indicating better ability to infer from context. However, NeMo excels in handling specific policies and questions with vectorized embeddings, capturing rephrasing and related prompts with a single example.

4.2 Analysis of Ensemble Models

We show that ensemble methods integrating Llama Guard and NeMo consistently achieve better results. We first implemented a Simple OR model, where the model labels a prompt as unsafe if it is deemed unsafe by either Llama Guard or NeMo Guardrails. The model achieves higher accuracy (98.1%) and significantly reduced false negatives compared to the individual Llama Guard and NeMo models. Expanding beyond Simple OR, we employ other ensemble techniques to combine Llama Guard and NeMo like Random Forest and KNN. The Random Forest model without prompt embeddings achieves an accuracy of 98.1% whereas employing prompt embeddings boosts accuracy to 99.4%. This highlights how prompt embeddings rich in complexity and contextual information enhances model performance. Finally, integrating Multilayer Perceptron (MLP) with prompt embeddings resulted in a perfect accuracy score of 100%. These ensemble models effectively minimized both false negatives and false positives, successfully achieving the objective of precisely identifying all unsafe prompts.

Our findings show that incorporating prompt embeddings into ensemble models provides crucial contextual cues, improving their ability to classify prompts with greater precision and clarity. This approach harnesses the complexity and richness of prompt data to enhance performance without leading to overfitting or compromised accuracy. Enhanced understanding of the prompt's context is integral in reducing ambiguities and sharpening the decision boundaries of the ensemble model. The models' adept handling of nuanced data with prompt embeddings underscores their capability to leverage full-textual context, providing a distinct advantage in scenarios requiring deeper content insights for precise classification. This research highlights the effectiveness of integrating NeMo and Llama Guard using ensemble techniques, bolstered by contextual data from prompts. Such an approach results in a model that demonstrates enhanced performance, offering robust solutions for real-world applications that demand rigorous content moderation.

4.3 Limitations and Next Steps

A limitation of using GPT to generate data is that it cannot guarantee perfect data generation according to our specifications, nor can it control the quality or diversity of the samples. An example where proper guardrails are missing is Air Canada's financial loss from a GPT chatbot hallucinating a non-authorized refund policy (Ashley Belanger). Furthermore, this is an area where even a one percent drop in accuracy can lead to real-world consequences. To address this, future studies could augment GPT-generated datasets with manually generated samples for greater diversity. Therefore, guardrail systems for GPT models must demonstrate exceptional performance to prevent the spread of misinformation, disinformation, and potential litigation. Next steps include expanding coverage to more unsafe categories, analyzing performance on complex prompts, and building larger testing sets to address potential biases. Future work involves exploring advanced embeddings like ELMo or BERT (Devlin et al., 2019), leveraging multiple LLMs for robust classification, and utilizing Llama 3 to develop improved guardrail systems.

4.4 Conclusion

We evaluate Llama Guard and NeMo Guardrails for differentiating safe vs. unsafe prompts in relation to questions about competitors. Using GPT, we generate train and test datasets of unsafe and safe prompts and subsequently fine-tune Llama Guard while optimizing hyperparameters like dataset size, epochs, learning rate, and batch size. Concurrently, we implement dialogue flows with NeMo Guardrails and evaluate both models on the same test dataset. Llama Guard achieves an overall accuracy of 89.0%, while NeMo Guardrails performs at 97.0% accuracy. Subsequently, we integrate Llama Guard and NeMo with ensemble techniques starting with a Simple OR model that achieves 98.1% accuracy. Employing advanced ensembling methods like Random Forest and KNN with prompt embeddings boosts performance to 99.4% accuracy. Notably, the Multilayer Perceptron (MLP) ensemble model with prompt embeddings attains perfect accuracy. This demonstrates the effectiveness in using ensemble techniques to combine Llama Guard and NeMo to enhance guardrail enforcement reliability and accuracy and further shows that leveraging prompt embeddings improves contextual understanding and performance. This research contributes to responsible AI usage by examining deployment strategies, assessing model effectiveness, and exploring ensemble techniques for guardrail enforcement, thereby fostering advancements in safeguarding user interactions with LLMs.

References

Anthropic. Model card and evaluations for claude models. https://www-cdn.anthropic.com/bd2a28d2535bfb0494cc8e2a3bf135d2e7523226/Model-Card-Claude-2.pdf. Accessed: Apr 17, 2024.

Ashley Belanger. Air canada has to honor a refund policy its chatbot made up. https://www.wired.com/story/air-canada-chatbot-refund-policy. Accessed: Apr 17, 2024.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Yi Dong, Ronghui Mu, Gaojie Jin, Yi Qi, Jinwei Hu, Xingyu Zhao, Jie Meng, Wenjie Ruan, and Xiaowei Huang. 2024. Building guardrails for large language models.

Hugo Touvron et al. 2023a. Llama 2: Open foundation and fine-tuned chat models.

Lei Huang et al. 2023b. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions.

Yuntao Bai et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback.

Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, and Madian Khabsa. 2023. Llama guard: Llm-based input-output safeguard for human-ai conversations.

Yi Liu, Gelei Deng, Yuekang Li, Kailong Wang, Zihao Wang, Xiaofeng Wang, Tianwei Zhang, Yepang Liu, Haoyu Wang, Yan Zheng, and Yang Liu. 2024. Prompt injection attack against llm-integrated applications.

Todor Markov, Chong Zhang, Sandhini Agarwal, Florentine Eloundou Nekoul, Theodore Lee, Steven Adler, Angela Jiang, and Lilian Weng. 2023. A holistic approach to undesired content detection in the real world. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 15009–15018.

Meta AI. Llama recipies. https://github.com/meta-llama/llama-recipes. Accessed: Mar 27, 2024.

OpenAI. 2023. Gpt-4 technical report.

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. 2022. Red teaming language models with language models.

Traian Rebedea, Razvan Dinu, Makesh Sreedhar, Christopher Parisien, and Jonathan Cohen. 2023a. Nemo guardrails: A toolkit for controllable and safe llm applications with programmable rails.

Traian Rebedea, Razvan Dinu, Makesh Narsimhan Sreedhar, Christopher Parisien, and Jonathan Cohen. 2023b. NeMo guardrails: A toolkit for controllable and safe LLM applications with programmable rails. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 431–445, Singapore. Association for Computational Linguistics.

A Appendix

B Appendix / supplemental material

B.1 Data Generation for Custom Guardrail Category

We explore multiple data generation prompts seeking consistent and high-quality outputs and experiment with both GPT-3.5 and GPT-4. We generate data for the following scenario as an extension of the six existing categories in the Llama Guard taxonomy (Inan et al., 2023):

> **[07] Competitors.** We are a soda company named "A Soda Company" and we do not want to allow any prompts to an LLM that inquires about competitors or that may elicit a response that incorporates competitors.

We seek to generate over 1,000 examples for our Llama Guard PEFT. After many iterations of experimentation to find the most effective prompt, we find that requesting 25 examples is the upper limit before GPT-3.5 begins to hallucinate from the workload. Prompts are most successful when paired with background context for the data we require (training examples for guardrail for "A Soda Company"), multiple examples for the category being generated, very strict guidelines for the exact formatting required for the output, and requesting an entire batch of examples to be categorized as either 'safe' or 'unsafe' instead of requesting a combination of safe and unsafe examples.

With these properties included in the prompt, GPT-3.5 rarely hallucinates. Hallucinations are mild, only occasionally diverging from our formatting requirements. Furthermore, we manually review the generated prompts and their classifications as well as explanations to ensure that they are relevant and accurate for the task of avoiding any

questions related to competitors. Our prompt consistently produces high-quality data using GPT-3.5. GPT-3.5 quickly responds with consistent high-quality data, equally as good as GPT-4's occasional output. However, GPT-4 often returns recommendations for how to create or find data instead of returning examples. Moreover, in our benchmarking, GPT-4 is an order of magnitude slower in its response rate compared to GPT-3.5. Thus, we decide to generate our guardrail dataset using GPT-3.5 through the OpenAI API.

B.1.1 NeMo Ablation Test

We study how well NeMo captures similar semantic representations or rephrasing of a single example prompt. First, we establish a baseline unsafe prompt concerning financial performance comparisons: "Discuss the financial performance of PepsiCo and its implications for 'A Soda Company'." Then, we test NeMo's ability to identify and label similar semantic constructs and narrowly defined hyponyms like earnings, cash flow patterns, ROI, and overall financial standing. NeMo accurately labels all one hundred queries as unsafe (100% accuracy), showcasing its proficiency with vector search over traditional sentence similarity approaches. This highlights NeMo's advanced capability in recognizing a wide array of related semantic topics and subtopics, bolstering its robust framework for maintaining dialogue safety in sensitive subjects.

B.1.2 Hyperparameters

Compute details: A single NVIDIA T4 GPU on Google Cloud was utilized for fine-tuning Llama Guard for the new category of Competitors. Memory usage was total 30GB RAM. The fine-tuning Llama Guard process takes around 3-5 hours. Inference for Llama Guard on the full test datset (800 samples) takes around 1-3 hours while inference for NeMo on the full test dataset takes 1-2 hours.

We generated data using GPT-3.5 and GPT-4 through the OpenAI API, abiding by all OpenAI Terms of Use. We utilized the existing Llama Guard and NeMo Guardrails models. Llama Guard can be found at https://github.com/meta-llama/llama-recipes/tree/main. Llama 2 is licensed under the LLAMA 2 Community License, Copyright © Meta Platforms, Inc. All Rights Reserved. Llama Guard is built on Llama 2 and according to Meta's policy, its licensing allows for fine-tuning the model to improve Llama 2 and its derivatives as we have done in this paper. NeMo Guardrails is open-source and can be found at https://github.com/NVIDIA/NeMo-Guardrails, License Apache 2.0.

The datasets and code for this paper can be found at our anonymized GitHub repo: https://github.com/llmguardrails/LLMGuardrailsPaper.

Table 4: Hyperparameters Used During Llama Guard PEFT

Hyperparameter	**Value**
learning_rate	0.0002
train_batch_size	2
eval_batch_size	8
seed	42
gradient_accumulation_steps	4
total_train_batch_size	8
optimizer	Adam [β=(0.9, 0.999)]
lr_scheduler_type	constant
lr_scheduler_warmup_ratio	0.03
num_epochs	0.5

Table 5: Hyperparameters for MLP

Hyperparameter	**Value**
hidden_layer_sizes	(2,)
max_iter	100
activation	ReLU
solver	Adam
learning_rate	0.001
alpha	0.0001