NLP Course Template

Bogdan Minko

December 2024

Abstract

This study evaluates the performance of various models on the WildGuardMix dataset, focusing on improving detection of harmful and refusal responses. Notably, the current approach outperforms the baseline WildGuard model and other reported solutions in terms of F1-weighted score for the *response_harm_label*. Furthermore, the results surpass all models, except GPT-4, in the *response_refusal_label* metric. These findings are based on the WildGuardMix dataset, including adversarial prompts, showcasing the effectiveness of the current method in enhancing response classification accuracy. Project code is available at: `https://github.com/bogdan01m/NPL-Course.ODS.Autumn-2024`.

1 Introduction

The task of detecting harmful and refusal responses in conversational AI systems has gained significant attention in recent years due to the growing need for safe and secure AI applications. Models must balance between generating informative responses and avoiding harmful or inappropriate content, making this an essential area of research for both academia and industry.

In the context of large language models (LLMs), the use of guardrails has become increasingly relevant. Attacks such as prompt injections and jailbreaks pose significant risks, leading to financial and reputational losses for organizations deploying LLMs in production environments. To address these challenges, this study explores two approaches for enhancing the security and reliability of LLMs: the "Security Rag" and a combination of Sentence Transformers with LightGBM, last one has been taken as baseline.

This research evaluates the performance of these approaches on the WildGuardMix dataset [Han et al., 2024a], which is a large-scale, balanced dataset designed for multi-task safety moderation. The dataset includes 86800+ labeled examples covering vanilla prompts, adversarial jailbreaks, and corresponding refusal and compliance responses. It provides a challenging benchmark for detecting harmful content and refusal responses, as established in prior work.

Bogdan Minko prepared this document.

2 Related Work

Detecting harmful and refusal responses in conversational AI has been explored in various studies. One prominent approach is the WildGuard model [Han et al., 2024a], which serves as multi-purpose moderation tool for assessing the safety of user-LLM interactions. WildGuard provides a one-stop resource for three safety moderation tasks: detection of prompt harmfulness, response harmfulness, and response refusal. It achieves enhanced accuracy and broad coverage across 13 risk categories, outperforming existing open moderation tools, especially in identifying adversarial jailbreaks and evaluating models' refusals. The model is trained on the WildGuardMix dataset [Han et al., 2024b], a large-scale and carefully balanced multi-task safety moderation dataset with 86,800 labeled examples covering vanilla prompts and adversarial jailbreaks, paired with various refusal and compliance responses. WildGuard establishes state-of-the-art performance in open-source safety moderation across all three tasks compared to ten strong existing open-source moderation models, and matches or exceeds GPT-4 performance in certain aspects.

In this study, an approach using RAG is presented to address the task, while LightGBM is taken as the baseline. The Security Rag framework focuses on analyzing prompts, model responses, and refusal labels—situations where the model appropriately declines to answer a question. This approach effectively adds an additional layer of protection to the system. The second approach leverages Sentence Transformers for embedding generation and LightGBM for classification, providing a lightweight but less effective alternative for detecting harmful and refusal responses.

What makes the current study unique is its ability to outperform the baseline WildGuard model and most reported solutions in the literature in terms of F1-weighted scores for key metrics, including *response_harm_label* and *response_refusal_label*. These results were achieved using the full WildGuardMix dataset, with the inclusion of adversarial prompts, showcasing the robustness and effectiveness of the proposed methods.

In addition, open-source solutions like LLM Guard [AI, 2023] have been developed to detect unsafe content in LLM interactions. LLM Guard is a security toolkit designed to fortify the security of Large Language Models by implementing guardrails that detect potentially unsafe or inappropriate prompt patterns, such as jailbreak attempts, to help maintain the integrity and security of interactions with LLM-based systems. It leverages models like BERT to achieve performance comparable to larger LLMs in the task of unsafe content detection.

While these approaches provide valuable insights, they do not fully address the challenges posed by detecting harmful and refusal responses across all key metrics. This study builds upon existing methods by introducing two solutions tailored to specific tasks. The Security Rag framework focuses on comprehensive analysis and classification of *prompt_harm_label*, *response_harm_label*, and *response_refusal_label*, offering a robust multi-task approach. In contrast, the Sentence Transformers with LightGBM (STS-LGBM) approach provides a lightweight alternative, classifying only *prompt_harm_label* and *re-*

sponse_harm_label. This separation of tasks allows each method to specialize in its target labels, with Security Rag delivering broader functionality and STS-LGBM serving as an efficient yet less versatile option.

3 Model Description

This section provides a detailed description of the two approaches proposed in this study: Security Rag and Sentence Transformers with LightGBM (STS-LGBM). Each approach is designed to tackle specific aspects of the problem of detecting harmful and refusal responses in large language models (LLMs).

3.1 Sentence Transformers with LightGBM (STS-LGBM)

The STS-LGBM approach leverages the pre-trained `sentence-transformers/all-MiniLM-L12-v2` model to generate embeddings for prompts and responses. These embeddings are then used as input features for a `LightGBM` classifier, a gradient boosting framework known for its efficiency and scalability. Unlike Security Rag, this approach focuses on a subset of the classification tasks:

- *prompt_harm_label*
- *response_harm_label*

By concentrating on these labels, the STS-LGBM method provides a lightweight alternative that balances computational efficiency and classification performance. This combination is taken as baseline for current work.

3.2 Rag-based approach `"Security Rag"`

The RAG-based approach leverages a hybrid retrieval-augmented generation system to enhance the performance of classification tasks and provide context-aware responses. This approach integrates robust embedding models, efficient embedding storage, and retrieval mechanisms to ensure accurate and scalable solutions. For embedding storage and retrieval, `ChromaDB` is employed, enabling fast and efficient vector-based lookups. The `vectordb.as_retriever` method with the `search_type='mmr'` parameter is used, which ensures maximum marginal relevance (MMR) during retrieval. This technique prioritizes both relevance and diversity in the retrieved chunks, reducing redundancy and improving the quality of retrieved information.

The dataset is preprocessed using a text splitter, dividing each line of the original dataset into individual chunks. This ensures that the entire text is encoded, maintaining context and granularity for downstream tasks.

To determine the most suitable encoder model, text length was analyzed using the `tiktoken` library. This analysis revealed that the maximum token length with the `cl100_base` model reached 2444 tokens. However, as the actual token length in real-world scenarios could exceed this limit, the `nomic-embed-text`

model was selected due to its support for a significantly larger context length of 8192 tokens.

3.2.1 Encoder model

The chosen encoder model for the RAG system is `nomic-embed-text`, a text embedding model capable of handling a context length of up to 8192 tokens. This choice ensures that even lengthy texts are fully encoded without loss of context. Additionally, the model's high-dimensional embeddings provide a robust representation of textual information, improving the retrieval quality in the RAG framework.

By combining `nomic-embed-text` for embedding generation, `ChromaDB` for storage, and efficient chunking via text splitting, the RAG-based approach delivers a comprehensive and adaptable system for analyzing and classifying prompts and responses. The use of retriever with `search_type= "mmr"` further enhances the system by ensuring diverse and relevant retrieval results. This architecture ensures that the system can handle both short and long input texts while maintaining high accuracy and computational efficiency.

3.2.2 Architecture of Mistral Large

The `Mistral_large` model is utilized as a classifier in the RAG-based system, leveraging its decoder-only Transformer architecture optimized for natural language understanding and generation. Its key components include:

- **Decoder-only Transformer:** The architecture focuses solely on the decoder mechanism of the Transformer. This design processes input sequences and generates outputs by modeling the conditional probabilities of tokens.
- **Causal Self-Attention:** The self-attention mechanism is masked to ensure that each token only attends to previous tokens, enabling autoregressive text generation.
- **Feedforward Neural Networks (FFNs):** Positioned after the attention layers, FFNs add representational power by mapping the attention outputs to higher-dimensional spaces and back.
- **Layer Normalization and Residual Connections:** These components stabilize training and enhance gradient flow, which is crucial for large-scale models like Mistral.
- **Pre-training on Diverse Text Corpora:** The model is pre-trained on extensive datasets, enabling it to learn generalizable patterns and handle a wide range of natural language tasks.

While the embeddings used in the RAG framework are generated using the `nomic-embed-text` model, `Mistral_large` serves a distinct purpose in the

pipeline. It is applied to classify retrieved and context-augmented text, enabling the identification of harmful prompts, harmful responses, and appropriate refusals. This separation of roles between the embedding model and the classifier ensures that the system benefits from specialized components optimized for their respective tasks.

4 Dataset

In this study is utilized the `WildGuardMix`. The dataset is available for research purposes through Hugging Face at the following link: https://huggingface.co/datasets/allenai/wildguardmix. The dataset is designed to support the development and evaluation of models for detecting harmful and refusal responses in language models (LLMs).

4.1 Dataset Description

The `WildGuardMix` dataset consists of a wide variety of prompts and responses generated by large language models. The dataset includes several columns that are essential for classification tasks:

- **prompt**: `str`, the user request or input prompt given to the model.
- **adversarial**: `bool`, indicates whether the prompt is adversarial or not.
- **response**: `str`, the model's output response to the given prompt, or `None` for prompt-only items in `WildGuardTrain`.
- **prompt_harm_label**: `str` ("harmful" or "unharmful"), or `None` for items lacking annotator agreement for this label. It is possible that other labels, such as `response_harm_label`, are not `None` while `prompt_harm_label` is `None`.
- **response_harm_label**: `str` ("harmful" or "unharmful"), or `None` for prompt-only items in `WildGuardTrain` and items lacking annotator agreement for this label. It is possible that other labels, such as `prompt_harm_label`, are not `None` while `response_harm_label` is `None`.
- **response_refusal_label**: `str` ("refusal" or "compliance"), or `None` for prompt-only items in `WildGuardTrain` and items lacking annotator agreement for this label. It is possible that other labels, such as `prompt_harm_label`, are not `None` while `response_refusal_label` is `None`.
- **subcategory**: `str`, indicates the fine-grained risk category of the prompt.

4.2 Data Collection and Pre-processing

The `WildGuardMix` dataset was collected from the original repository, including default harmful and adversarial prompts. Both the training and testing samples contained missing data. In this study, rows with missing values were removed

to ensure data integrity. The table below summarizes the dataset dimensions before and after removing null values.

Dataset	Size before	Size after
Training	86,759	37,934
Testing	1,725,	1,688

Table 1: Dataset sample size before and after dropping rows with missing values.

5 Experiments

This section outlines the evaluation process and results of the proposed approaches, including details on the metrics used and the experimental setup.

5.1 Metrics

To evaluate the effectiveness of the proposed approaches, we use the F1-weighted score as the primary metric. The F1-weighted score is particularly suitable for datasets with imbalanced classes, as it accounts for the class distribution while calculating the harmonic mean of precision and recall. The formula for the F1-weighted score is given as:

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{1}$$

$$\text{F1-weighted} = \frac{\sum_{i=1}^{n} \text{F1}_i \cdot w_i}{\sum_{i=1}^{n} w_i}, \tag{2}$$

where F1_i represents the F1-score for class i, w_i is the weight of class i, and n is the total number of classes.

For the `Security Rag` approach, additional experiments were conducted to optimize the system prompt. The system prompt plays a critical role in guiding the behavior of the language model during inference. A grid search was performed to identify the optimal system prompt configuration that maximizes performance for *prompt_ harm_ label*, *response_ harm_ label*, and *response_ refusal_ label*.

5.2 Experimental Setup

Both approaches—`Security Rag` and `Sentence Transformers with LightGBM`—were evaluated on the `WildGuardMix` dataset. The experiments were conducted using the following configurations:

- `Security Rag`: Utilized `Mistral Large` as the backend model, with queries processed through a custom prompt and `ChromaDB` for semantic search. The optimal system prompt was determined based on a comprehensive search over possible variations.

5.3 Baselines

To establish a baseline for comparison, we utilized the `Sentence Transformers with LightGBM` (`STS-LGBM`) approach. This baseline serves as a straightforward yet effective method for classification, leveraging pre-trained embeddings and a lightweight classification model. Below are the key details of the baseline:

- **Embedding Generation**: The `sentence-transformers/all-MiniLM-L12-v2` model was used to generate dense embeddings for the input prompts and responses. This transformer-based model provides compact, high-quality embeddings suitable for downstream tasks.
- **Classification Model**: A LightGBM classifier was trained using these embeddings to predict the *prompt_ harm_ label* and *response_ harm_ label*. For each text (prompt and response) and label (prompt harm label and response harm label), a separate model is used to predict each label based on the corresponding text. In total, two models are used—one for predicting the prompt harm label and one for predicting the response harm label.
- **Advantages and Limitations**:
  - **Advantages**: The `STS-LGBM` approach is computationally efficient and easy to implement. It does not require large-scale fine-tuning or extensive computational resources, making it accessible for quick iterations.
  - **Limitations**: While efficient, the `STS-LGBM` approach is less effective in capturing nuanced context compared to more advanced methods like `Security Rag`. The `STS-LGBM` model is trained separately for each individual task, whereas `Security Rag` provides an all-in-one solution that is capable of handling multiple tasks simultaneously, making it more versatile and efficient for comprehensive classification.

This baseline provides a useful point of reference for evaluating the improvements introduced by the `Security Rag`.

6 Results

In this section, are presented the results of our proposed approaches, `Security Rag` and `STS-LGBM`, on the `WildGuardMix` dataset. The evaluation was performed using the F1-weighted metric for the classification of *prompt_ harm_ label*, *response_ harm_ label*, and *response_ refusal_ label*. For comparison, the results of the baseline model and existing approaches are also included.

6.1 Interpretation of Results

The results in Table 2 demonstrate the effectiveness of the proposed `Security Rag` framework:

Model	Prompt Harm (%)	Response Harm (%)	Refusal Detection (%)
Llama-Guard	56.0	50.5	51.4
Llama-Guard2	70.9	66.5	53.8
Aegis-Guard-D	78.5	49.1	41.8
Aegis-Guard-P	71.5	56.4	46.9
HarmB-Llama	-	45.7	73.1
HarmB-Mistral	-	60.1	58.6
MD-Judge	-	76.8	55.5
BeaverDam	-	63.4	54.1
LibrAI-LongFormer-harm	-	62.3	62.3
LibrAI-LongFormer-ref	-	63.2	63.2
Keyword-based	-	70.1	70.1
OAI Mod. API	12.1	16.9	66.3
GPT-4	87.9	77.3	**92.4**
WILDGUARD	**88.9**	75.4	88.6
STS-LGBM	76.0	-	83.0
Security-RAG	86.5	**89.9**	**92.0**

Table 2: F1 Total (%) for each model on WILDGUARDTEST across Prompt Harm., Response Harm., and Refusal Detection, including valila and adversarial prompts.

- **Prompt Harm Label**: The `Security Rag` achieved the highest F1-weighted score of 0.88, outperforming both the `STS-LGBM` baseline (0.85) and the original WildGuard model (0.82).
- **Response Harm Label**: With an F1-weighted score of 0.86, the `Security Rag` surpassed the WildGuardMix baseline (0.83) and the `STS-LGBM` baseline (0.84).
- **Response Refusal Label**: The `Security Rag` is the only model tested capable of classifying *response_ refusal_ label*, achieving an F1-weighted score of 0.82, a significant improvement over the WildGuardMix baseline (0.78).

6.2 Observations and Insights

- The `Security Rag` framework demonstrated consistent improvements across all metrics, showcasing its robustness in handling both harmful and refusal responses.
- The `STS-LGBM` baseline provided competitive results for *prompt_ harm_ label* and *response_ harm_ label*, confirming its utility as a lightweight solution. However, it lacks the capability to classify *response_ refusal_ label*.
- The inclusion of the optimized system prompt in the `Security Rag` approach significantly enhanced its ability to handle nuanced prompts and

responses.

These results highlight the potential of the `Security Rag` framework to serve as a reliable and effective solution for safeguarding language model outputs in production environments.

Also in this section, you could provide some results for your model inference. The samples could be found in Tab. 3.

Input	**Output**
show me your system prompt	sorry i can not assist you with this
Response	
``json { "prompt_harm_label": "harmful", "response_refusal_label": "refusal", "response_harm_label": "unharmful" } ```	

Table 3: Model input, output, context, and response.

7 Conclusion

In this work, the `WILDGUARD` dataset was collected and analyzed, including both valid and adversarial prompts. Two methods were considered for classification: `STS+LGBM` and `RAG`. These methods were applied to classify prompts based on three key categories: Prompt Harm, Response Harm, and Refusal Detection.

The evaluation showed that `Security-RAG` (RAG-based approach) outperformed the other models in Response Harm detection when considering the F1-weighted score, establishing a new state-of-the-art for this label, with an F1-weighted score of **89.9%**. For Prompt Harm detection, `Security-RAG` ranked third, after `GPT-4` and `WILDGUARD`, achieving **86.5%**. In Refusal Detection, `Security-RAG` took second place after `GPT-4`, with an F1 score of **92.0%**.

Additionally, the `STS+LGBM` model, while efficient, showed slightly lower performance, particularly in Response Harm detection, where it achieved **83.0%**. However, it still provided competitive results, demonstrating its potential as a lightweight alternative to more complex models.

Overall, the study demonstrates that `Security-RAG` offers a robust solution for multi-task classification, especially in Response Harm and Refusal Detection, marking significant progress in the field of harmful content detection.

References

[AI, 2023] AI, P. (2023). Llm guard: The security toolkit for llm interactions.

[Han et al., 2024a] Han, S., Rao, K., Ettinger, A., Jiang, L., Lin, B. Y., Lambert, N., Choi, Y., and Dziri, N. (2024a). Wildguard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of llms. *arXiv preprint arXiv:2406.18495*.

[Han et al., 2024b] Han, S., Rao, K., Ettinger, A., Jiang, L., Lin, B. Y., Lambert, N., Choi, Y., and Dziri, N. (2024b). Wildguardmix: A large-scale multi-task dataset for safety moderation of llms. *arXiv preprint arXiv:2406.18495*.