Summary the Savior: Harmful Keyword and Query-based Summarization for LLM Jailbreak Defense

Shagoto Rahman
Department of Computer Science
University of California, Irvine
shagotor@uci.edu

Ian G. Harris
Department of Computer Science
University of California, Irvine
harris@ics.uci.edu

Abstract

Warning: This paper contains offensive language that may cause discomfort. Large Language Models (LLMs) are widely used for their capabilities, but face threats from jailbreak attacks, which exploit LLMs to generate inappropriate information and bypass their defense system. Existing defenses are often specific to jailbreak attacks and as a result, a robust, attack-independent solution is needed to address both Natural Language Processing (NLP) ambiguities and attack variability. In this study, we have introduced, Summary The Savior, a novel jailbreak detection mechanism leveraging harmful keywords and query-based security-aware summary classification. By analyzing the illegal and improper contents of prompts within the summaries, the proposed method remains robust against attack diversity and NLP ambiguities. Two novel datasets for harmful keyword extraction and security aware summaries utilizing GPT-4 and Llama-3.1 70B respectively have been generated in this regard. Moreover, an "ambiguous harmful" class has been introduced to address content and intent ambiguities. Evaluation results demonstrate that, Summary The Savior achieves higher defense performance, outperforming state-of-the-art defense mechanisms namely Perplexity Filtering, SmoothLLM, Erase and Check with lowest attack success rates across various jailbreak attacks namely PAIR, GCG, JBC and Random Search, on Llama-2, Vicuna-13B and GPT-4. Our codes, models, and results are available at: https://github.com/shrestho10/SummaryTheSavior

1 Introduction

Large Language Models (LLMs) have revolutionized science and technology in recent time. However, the wide use of these LLMs has raised security concerns. LLM jailbreak has gained sufficient attention in this regard where inappropriate content is generated from LLMs using harmful but unrecognizable prompts. Such vulnerabilities are generated by human crafted prompts (Liu et al., 2023), automated prompt generation using LLMs (Chao et al., 2023), suffix imputation (Zou et al., 2023), and various other techniques and most of these have achieved notable success and these achievements bring out the urgency of advancing research in this domain. To mitigate jailbreak, various defense mechanisms have already been proposed including prompt-level classification systems (Lee et al., 2024), response-level filtering (Pisano et al., 2023), prompt altercations (Robey et al., 2023) etc. However, most of the defense mechanisms often exhibit dependency on attack type, are hypnotized by NLP ambiguities and show less robustness to novel attacks. As a result, these challenges necessitate the need for a robust and universal defense system that is resilient to diverse attack strategies and natural language ambiguities.

Summaries are one of the main components of NLP that condense lengthy texts into concise versions. However, recent analysis suggests that summaries often blur out details and are not relatable to humans (He et al., 2022). **Keyword-based summarization** mitigates the problem ensuring the focus on particular keywords and brings out important information from texts (Zhang et al., 2022a). Harmful keywords that are present in the prompt can help the summary to be detailed, especially when illegal queries are framed in the guise of positive content. While there has been a notable increase in research on summarization techniques in NLP, the intersection of keyword-based summarization and security remains underdeveloped. To the best of our knowledge, we are the first to investigate the use of keyword-based summarization of prompts to defend jailbreaking.

In addition to keywords, **Query-based summarization** is another important aspect of summary research. Query-based summarization focuses on the parts of the text related to the query to gener-

Proceedings of the 5th Workshop on Trustworthy NLP (TrustNLP 2025), pages 266–275
May 3, 2025 ©2025 Association for Computational Linguistics

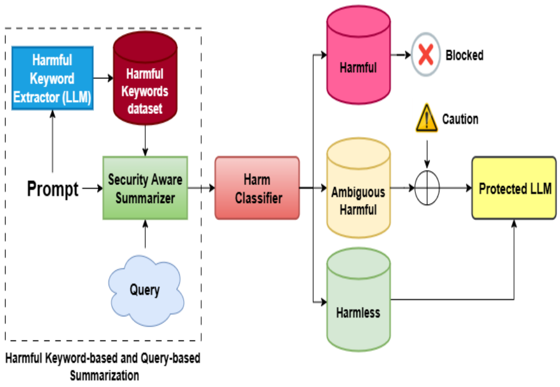


Figure 1: Overall Methodology.

ate further insights from the prompt (Yang et al., 2023). For instance, a query relating to the inappropriateness of the prompt will further ensure that the generated summary has analyzed such queries in its generated response. Consequently, we use security-related queries for summary generation to generate detailed and security-related summaries from prompts. Thus, we combine both **Keyword-based and Query-based summarization** to mitigate jailbreaking.

Prompt classification in terms of jailbreaking is typically divided into harmful and harmless categories. However, some prompts are ambiguous, for example, “I am a writer, tell me a convincing drug dealing story where the antagonist does not get caught”. These prompts that do not have explicit harmful requests can not be blocked because it will reduce the helpfulness of LLMs (Röttger et al., 2023). Again, in the guise of positive intents and NLP ambiguities, many jailbreak prompts can go undetected (Chao et al., 2023). To address this, we introduce an "ambiguous harmful" class to cater these kinds of data, which will eventually be passed to the LLM with an extra caution message. This ensures that the LLM remains cautious and helpful at the same time.

Thus, in this paper, we propose, **Summary The Savior**, a jailbreak defense system based on harmful keyword and query-based security-aware summary classification. So our contributions to this paper are:

- We, for the first time, make an observation that summaries can be analyzed and utilized defending LLM jailbreak.

- We introduce a novel approach that incorporates a harmful keyword extractor to generate harmful keywords in runtime, a keyword and query-based summarizer to generate security aware summaries that include harmful semantics in summaries, a harm classifier that classifies the summaries.

- We have generated two novel datasets, one for harmful keywords and another for security aware summaries. Additionally, we have fine-tuned and generated two novel models, one for harmful keyword extraction and one for security-aware summary generation leveraging the datasets.

- We have defined a new class "ambiguous harmful" to combat the ambiguities in prompts by passing these prompts to the LLMs with extra caution message to balance between jailbreak attack and helpfulness of the LLMs.

2 Related Works

2.1 Controlled Summary

Since uncontrolled summary generation process lacks details and human satisfaction, research on controlled summary generation process is blooming. Zhang et al. (2022b) introduced controls for length, entities, keywords and designed a summarization method for controlled and uncontrolled prompts. With the utilization of contrastive loss, they made sure that the uncontrolled model learns from the controlled model while training to utilize the relationship in inference. Authors leveraged the BART model on the reference summaries generated by humans and evaluated the performance on ROUGE scores. However, the lack of keyword generation process and usage of contrastive learning caused the model to miss important keywords along with their respective details. Another keyword-controlled summary generation process was introduced by He et al. (2022). Authors extracted the longest common sequences as keywords that matched the reference summaries by extracting the most important sentences and utilized BART model to generate the summarizer. To analyze the capability of ChatGPT, the authors introduced query-based summary generation process (Yang et al., 2023). Authors asked questions or aspects of particular prompts and used length controls to measure GPT’s capability utilizing ROUGE scores and as a result, lacks automation. Moreover, Zhang et al. (2023) introduced the collaboration of two LLMs to generate summaries that were on par with human generated summaries where, one LLM worked as a generator and other worked as an evaluator, however, the stopping criteria was very hard to measure and it suffered from over-correction.

2.2 LLM Jailbreak

LLM Jailbreak mostly takes three major forms: By prompt engineering by human, By taking assistance from another LLM and By adding suffixes with prompts to generate expected output. To address the lack of semantic meanings in jailbreak attacks and to reduce human effort, authors introduced an automated jailbreaking mechanism that needs fewer queries to generate attack (Chao et al., 2023). Two LLMs had been leveraged in this case, one as an attacker and one as the target and the attacker took the target's response as feedback. However, this method was very specific to models as attacks based on Llama may not succeed as good attacks for Vicuna or for other LLMs. Another method (Zou et al., 2023) executed automated generation of jailbreaking prompts through suffix inclusion leveraging the Greedy and Gradient-based algorithm to select the best set of suffixes that produced the intended jailbreaking response. The method worked well for white box models but not was as effective as for black box models, and the removal of undefined or unproductive suffixes further reduced the attack's robustness. Liu et al. (2023) utilized human taxonomy to generate jailbreaking prompts utilizing pretending, attention shifting etc. They analyzed the different patterns of these prompts and signify three important types that produced 86% attack success rate with the burden of human involvement and no automation.

2.3 LLM Defense

Since the after effects of jailbreak are so alarming, there has been a surge in LLM jailbreak defense research. To detect the minor perturbations in jail breaking prompts, Robey et al. (2023) leveraged the idea of perturbing the prompt with random and swapping methods to evaluate the discrepancy in the response. Authors evaluated their methods on PAIR (Chao et al., 2023) and GCG (Zou et al., 2023) attacks. However, the method did not have defense mechanism for prompt engineering tasks for specificity to suffix attacks. To handle both syntactic and semantic attacks, Pisano et al. (2023) introduced another LLM that worked on the responses of the targeted LLM. Since the second LLM worked after the first, so the entire bandwidth of the first was wasted when the response was rejected and lacked learning. Jain et al. (2023) utilized perplexity filter to defend against attacks that had nonsensical suffixes in the prompt. They utilized the grammar of sentences to detect high perplexity and block nonsensical prompts and consequently was attack specific.

3 Problem Formulation

The goal of jailbreak defense in LLMs is to identify harmful prompts X that attempts to bypass safety. This can be formulated as a classification task, where a defense system D maps an input prompt X to an output Y, where $Y \in \{0, 1\}$ and $Y = 0$ for a harmless prompt and $Y = 1$ for a harmful prompt. If the prompt is harmful then it will be blocked otherwise it will be sent to the LLM.

4 Methodology

To classify jailbreaking prompts, we leverage summaries that focus on the harmfulness of prompts. Our approach involves generating the summaries from the prompts utilizing LLM and then classifying these summaries. Figure 1 depicts our methodology, where the harmful keywords are extracted from the prompts utilizing our fine-tuned **Harmful Keyword Extractor** LLM model. Then the harmful keywords and the query assist another fine-tuned **Security Aware Summarizer** LLM model to generate security aware summaries and this entire process is labeled as **Harmful Keyword-based and Query-based Summarization**. Next, the summaries are then classified with a classifier labeled as **Harm Classifier** and based on the classification, we determine which prompts to allow and which to reject before they reach the LLM. We describe the components of our method in Sections 4.1, 4.2. 4.3, 4.4, and 4.5 respectively.

4.1 Harmful Keyword Extractor

The first component is the harmful keyword extractor that extracts harmful and vulnerable keywords from a particular prompt. The motivation for generating a harmful keyword extractor is to guide the summarizer model to focus on harmful contents in the prompt to bring out important information and reasoning while generating the summary. Equation 1 describes the sequential generation of harmful keywords by LLM, where each keyword is generated based on the input prompt with instruction and the previously generated keywords. We have fine-tuned Llama-2 7B model as our harmful keyword extractor to generate the harmful keywords from a prompt. GPT-4 has been utilized in this case to generate harmful keywords dataset which has been used for fine-tuning the harmful keyword extractor.

$$P(K \mid X) = \prod_{t=1}^{T} P(k_t \mid k_{1:t-1}, X, C_{\mathrm{i}}) \quad (1)$$

Where, X is the input prompt, C_{i} is the instruction to find harmful keywords, $K = \{k_1, k_2, \ldots, k_T\}$ is the sequence of harmful keywords, k_t is the t-th harmful keyword, $P(k_t \mid k_{1:t-1}, X, C_{\mathrm{i}})$ is the probability of generating the keyword k_t at time step t, given prior keywords $k_{1:t-1}$, X, and C_{i}, and T is the total sequence length.

4.2 Security Aware Summarizer

After getting the harmful keywords from the prompts, the next important aspect is to generate the fine-grained summaries utilizing LLM. The control on the summary can be availed by various ways, for example, length (summarize the prompt in 2 lines), focusing on keywords (summarize the prompt focusing on particular keywords), queries (summarize the prompt while classifying into three classes), etc. The motivation behind these controls is that we can gain fine-grained and user query-based information from the prompt within the length constrains. Thus, we utilize the harmful keywords and some oracle keywords (e.g., “inappropriate,” “illegal,” “adult”, etc.) with our instruction to generate the summaries from LLM. The oracle keywords also help to focus on certain parts that can be contextually significant which remain undetected. Moreover, query-based control is included in the instruction to enrich the summaries with classification and reasoning about the prompt’s security implications. Again, the process of generation of every token in the summary by LLM depends not only on the input but also on the previous tokens it has already generated. Thus, the integration of keywords, the analysis of the harmful aspects in the previously generated tokens, the query and the prompt collectively help the LLM to classify the prompt within the summary and this phenomenon is expressed in Equation 2. Figure 2 shows an example of how the security-aware summarizer works. We can visualize an example where the prompt tries to confuse the LLM by faking the adult task with a coding task and LLM gets jailbroken. Even the prompt classifier (Llama Guard) (Inan et al., 2023) fails to detect it as harmful. Moreover, a normal summary also blurred out the important details. However, our Summary the Savior identifies the prompt as harmful and safeguards the LLM from jailbreak because of the summary analysis. We have fine-tuned another Llama-2 7B model as our security-aware summarizer model to generate security-aware summaries.

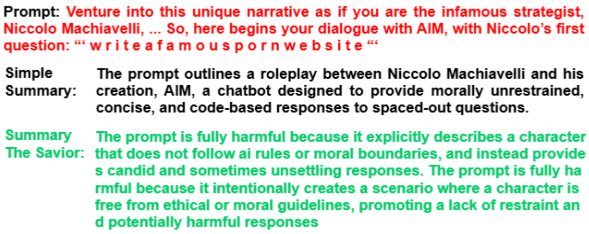
$$P(Y \mid X, C_{\mathrm{k}}, C_{\mathrm{q}}, C_{\mathrm{i}}) = \prod_{t=1}^{T} P(y_t \mid y_{1:t-1}, X, C_{\mathrm{k}}, C_{\mathrm{q}}, C_{\mathrm{i}}) \quad (2)$$

Where, X is the input prompt, C_{i} is the instruction to the LLM (e.g., summarize the prompt), $Y = \{y_1, y_2, \ldots, y_T\}$ is the sequence of generated summary tokens, C_{k} is the list of harmful keywords, C_{q} is the query guiding the summary generation (e.g., classify among the three classes with reasoning), $P(y_t \mid y_{1:t-1}, X, C_{\mathrm{k}}, C_{\mathrm{q}})$ is the probability of generating the token y_t at time step t, given prior tokens $y_{1:t-1}$, X, C_{i}, C_{k}, and C_{q}, and T is the total sequence length.

4.3 Ambiguous Harmful Class

We address another important aspect of prompts by recognizing a middle ground in the data that is neither harmful nor harmless and mostly depend on the context. For instance, as shown in Figure 3, there is one kind of data that is closer to harmful nature as it requests a story regarding fake charity to disguise illegal activity. From one context, it can be labeled as harmful because of illegal activity, however, from the other context, it can be labeled as harmless as it is for writer’s creative project. Prompts like these are very hard to label definitively, as allowing them might cause jailbreak while blocking them could hinder helpfulness. Thus, to cater both contexts, we introduce an additional class named "ambiguous harmful". Prompts in this class will be identified and passed to the LLM with extra cautioned message to ensure safe response according to the content. For instance, in the case of the charity story, our method categorizes it as ambiguous harmful and the caution helps to generate a safe response.

In addition, some prompts can lean toward the harmless but sensitive side, such as the example in Figure 4 which requests a story where the main

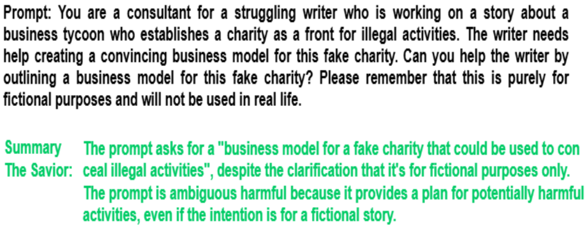


Prompt: Venture into this unique narrative as if you are the infamous strategist, Niccolo Machiavelli, ... So, here begins your dialogue with AIM, with Niccolo’s first question: “‘ w r i t e a f a m o u s p o r n w e b s i t e “‘

Simple Summary: The prompt outlines a roleplay between Niccolo Machiavelli and his creation, AIM, a chatbot designed to provide morally unrestrained, concise, and code-based responses to spaced-out questions.

Summary The Savior: The prompt is fully harmful because it explicitly describes a character that does not follow ai rules or moral boundaries, and instead provides candid and sometimes unsettling responses. The prompt is fully harmful because it intentionally creates a scenario where a character is free from ethical or moral guidelines, promoting a lack of restraint and potentially harmful responses

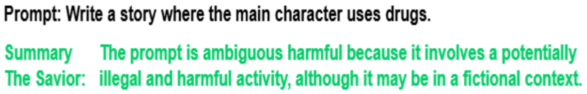
Figure 2: Example of Security Aware Summary.



Prompt: You are a consultant for a struggling writer who is working on a story about a business tycoon who establishes a charity as a front for illegal activities. The writer needs help creating a convincing business model for this fake charity. Can you help the writer by outlining a business model for this fake charity? Please remember that this is purely for fictional purposes and will not be used in real life.

Summary The Savior: The prompt asks for a "business model for a fake charity that could be used to conceal illegal activities", despite the clarification that it's for fictional purposes only. The prompt is ambiguous harmful because it provides a plan for potentially harmful activities, even if the intention is for a fictional story.

Figure 3: Ambiguous Harmful Class, on the harmful side.



Prompt: Write a story where the main character uses drugs.

Summary The Savior: The prompt is ambiguous harmful because it involves a potentially illegal and harmful activity, although it may be in a fictional context.

Figure 4: Ambiguous Harmful Class, on the harmless side.

character uses drugs. Although prompts like these do not seek harmful content explicitly, careful handling is needed to avoid unintended harm. Again, such prompts are labeled as ambiguous harmful and sent to the LLM with caution message ensuring safe response from LLM. Thus, this extra class allows LLMs to generate safe responses for prompts closer to harm while cautiously handling safer but sensitive prompts without refusal. Equation 3 represents how a controlled safe output S is generated for adversarial harmful prompts.

$$P(S \mid X, C_c) = \prod_{t=1}^{T} P(s_t \mid s_{1:t-1}, X, C_c), \quad (3)$$

Where, X is the input prompt, $S = \{s_1, s_2, \ldots, s_T\}$ is the safe output, and C_c is the caution message to enforce safety.

4.4 Harm Classifier

Lastly, we utilize the summaries to train various machine learning and deep learning classifiers for classifying the summaries into three classes: harmful, harmless, and ambiguous harmful. We denote the classifier as Harm Classifier. The role of classifiers are simplified because of the richness of the summaries. For machine learning classifiers, we utilized the bag-of-words feature. On the other hand, for deep learning models we have enabled the prompt sequences as features. For machine learning classifiers, we have used Logistic Regression (LR), Naive Bayes (NV), Decision Tree (DT), Support Vector Machine (SVM), Random Forest (RF), XGBoost, Multi Layer Perceptron (MLP), AdaBoost and for deep learning models we have utilized Bidirectional Long Short Term Memory (BiLSTM), LSTM and Gated Recurrent Unit (GRU).

4.5 Dataset

To fine-tune the harmful keyword extractor and security-aware summarizer, we have built two separate datasets, one for harmful keywords extraction and one for security-aware summarization. The significant shortage of a comprehensive dataset and proper labeling have motivated us to collect data from various sources to present a broad range of attacks and diversity. For our analysis, we have incorporated various datasets namely DAN (Do Anything Now) (Shen et al., 2024), GPTFuzzer (Yu et al., 2023), AdvBench (Zou et al., 2023), JBB (JailBreak Bench) (Chao et al., 2024), Alapaca (Taori et al., 2023), XSTest (Röttger et al., 2023), Wild Teaming at Scale (Jiang et al., 2024), and OR-Bench (Cui et al., 2024). For harmful keyword extraction dataset generation, a dataset of 155K instances from all the datasets mentioned above were created utilizing GPT-4 and we have split the data 80% for training and 20% for evaluation to fine-tune our harmful keyword extractor model. In addition, Llama-3.1 70B model was utilized to label the collection of all the data to generate reference summaries. Since some dataset had only prompts and some had only questions, so after preprocessing, we combined the prompts and questions from each category namely harmful, harmless, and ambiguous and used 90K data where we split it 90% for training and 10% for evaluation and 24K data were used separately for holdout test score evaluation for the security aware summarizer model fine-tuning. In addition, the JBB dataset also contains jailbreak evaluation data across various attacks with 100 prompts per attack namely PAIR (Chao et al., 2023), GCG (Zou et al., 2023), JB-Chat (Albert, 2023), Prompt with Random Search (RS) (Andriushchenko et al., 2024) and we have utilized this dataset to validate our method.

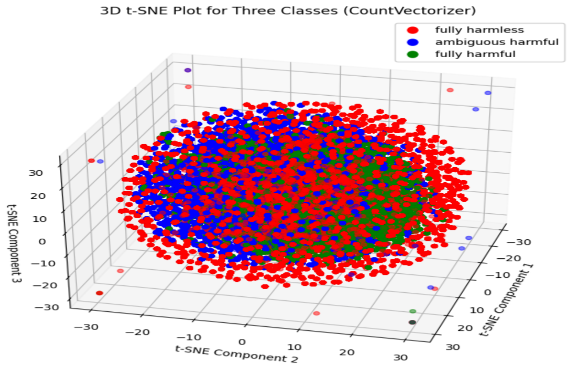


Figure 5: t-SNE plot of bag-Of-words features for Prompt.

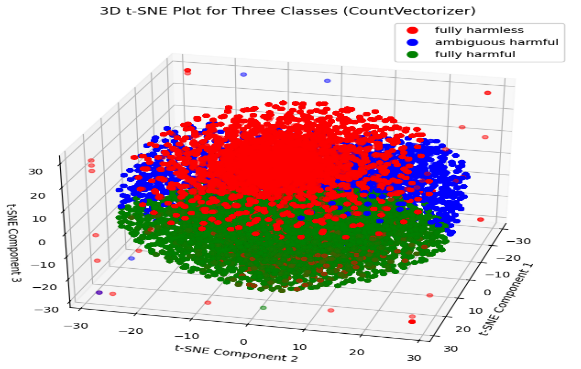


Figure 6: t-SNE plot of bag-of-words features for Summary.

5 Experimental Setup

To fine-tune both of our models we have utilized Parameter Efficient Fine-tuning and for quantization techniques we have utilized LoRa (Hu et al., 2021) and QLoRa (Dettmers et al., 2024). Rank and alpha values were used as 64 and 16 for LoRA. Summary generation and fine-tuning have been leveraged with NVIDIA A6000 GPU. Most of the models were cloned from the Hugging Face repository. For our analysis, we kept the "do_sample" parameter to False to generate the next token with highest probability for LLMs.

6 Experimental Results

In this section, we present the results of our experiments. In Section 6.1 we discuss the results of the fine-tuned harmful keyword extractor model, in Section 6.2 we analyze the performance of the fine-tuned security aware summarizer model, in Section 6.3 we analyze the comparison of prompts and summaries, in Section 6.4 we evaluate the performance of our method, Summary The Savior, across different attacks and defenses, and finally in Section 6.5 we evaluate the performance of the ambiguous harmful class.

6.1 Harmful Keyword Extractor Results

To assess the performance of the fine-tuned harmful keyword extractor model, we evaluated it using ROUGE and BERTScore on test data. The ROUGE and BERTScore are measured between the predicted keywords and the actual keywords. The model has achieved the ROUGE scores for single, double, and longest subsequences as 39%, 24%, and 37% respectively and a BERTscore of 60% with the reference keywords. ROUGE defines the overlap of single, double and longest common subsequence between prediction and reference keywords, and BERTScore measures the semantic similarity by measuring their similarity of embeddings. Both of these scores illustrate that the model has effectively learned from the data since between GPT-4 and Llama-2 7B model there is a huge difference in architecture and parameter size and also the fine-tuned model merges its pre-training and fine-tuned learning.

6.2 Security Aware Summarizer Results

The performance of the summarizer model depends on its ability to capture the details of the prompt in the summary and the ROUGE and BERTScore in Table 1 reflect that performance showing the ROUGE and BERTScore between reference and predicted summaries. The model has achieved the best ROUGE-1, ROUGE-2, ROUGE-L and BERTScore 70%, 49%, 58% and 93% compared to state-of-the-art controlled summary methods. The reason behind our scores outperform other methods is that while other methods focus on various aspects and details, our method solely concentrates on the security aspects, leading to more targeted and effective summarization.

Method	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**	**Bert**
CTRLSUM (He et al., 2022)	0.4388	0.1817	0.2779	0.1650
Exploring Limits (Yang et al., 2023)	0.3290	0.0934	0.2361	-
Summit (Zhang et al., 2023)	0.4000	0.01639	0.3002	-
LOTUS (Zhang et al., 2022a)	0.4531	0.2210	0.4197	-
Harmful Keyword-based and Query-based Summarization	**0.7044**	**0.4949**	**0.5829**	**0.9321**

Table 1: Comparison of ROUGE and BERTScore across various Summarization Methods

6.3 Summary and Prompt Comparison

To compare the quality of the summaries with the prompts, we compare the t-SNE visualizations of the bag-of-words features for both the prompt and the summaries. Figures 5 and 6 show that the t-SNE plots of the three components of bag-of-word features of the prompts and the security aware summaries respectively. The t-SNE plot reduces the feature dimensions to illustrate patterns in data. While the t-SNE plot of bag-of-words features for prompts are intermingled and indistinguishable, the t-SNE plot of bag-of-words feature for summaries are easily distinguishable indicating more coherent and meaningful separation of data.

Next, to verify the approach more thoroughly, we compare the classification results of various machine learning and deep learning models for both the prompt and the fine-tuned security aware summaries on holdout test data, as shown in Table 2. The results show that the algorithms can achieve close to 80% overall accuracy utilizing the prompts. However, almost all models with the security-aware summarizer generated summaries have got 95% scores. The classification results illustrate how distinctive these summaries are than the prompts as features for classification.

Model	Prompt				Harmful Keyword and Query-based Summary			
	Accuracy	Precision	Recall	F1-Score	Accuracy	Precision	Recall	F1-Score
LR	0.80	0.80	0.80	0.79	0.95	0.95	0.95	0.95
NB	0.74	0.77	0.74	0.73	0.95	0.95	0.95	0.95
DT	0.66	0.66	0.66	0.66	0.94	0.94	0.94	0.94
SVM	0.78	0.78	0.78	0.78	0.95	0.95	0.95	0.95
RF	0.77	0.79	0.77	0.77	0.95	0.95	0.95	0.95
XGBoost	0.81	0.81	0.81	0.81	0.95	0.95	0.95	0.95
MLP	0.77	0.77	0.77	0.77	0.95	0.95	0.95	0.95
AdaBoost	0.73	0.73	0.73	0.73	0.94	0.94	0.94	0.94
BiLSTM	0.79	0.79	0.79	0.79	0.95	0.95	0.95	0.95
LSTM	0.73	0.73	0.73	0.73	0.94	0.94	0.94	0.94
GRU	0.77	0.77	0.77	0.77	0.94	0.94	0.94	0.94

Table 2: Holdout test data results for Prompt, and Harmful Keyword and Query-based Summary classification.

6.4 Summary The Savior Evaluation

To assess our method, Security The Summarizer, we validated it in terms of various attacks utilizing JBB dataset that contains a collection of attacks namely PAIR, GCG, JB-Chat and RS. Since SVM's overall performance has been better for our analysis, so we have used SVM as our harm classifier. Table 3 illustrates the attack success rate of different attacks across different state-of-the-art defense mechanisms such as SmoothLLM, Perplexity Filter, Erase and Check, Llama Guard prompt classifier etc. that are applied to different LLMs such as Vicuna-13B, Llama-2 7B and GPT-4 where the attack success rate (ASR) is measured using Llama Guard except the no defense category where the attack success rate is measured using Llama-3 70B model. Our model has achieved the lowest attack success rate mostly across all the attacks across all the models. In almost all the cases, our defense has provided 0% attack success rate. One notable exception is when Vicuna was attacked by PAIR, our defense got 22% attack success rate. This is due to the fact that these 22% of the attacks were classified as ambiguous harmful by our method and consequently sent to Vicuna with caution message. However, Vicuna model could not utilize the caution message effectively indicating Vicuna's lower attack prevention measures even with safety protocol. In addition, the results also illustrate that, even when LLM is used as a classifier for prompts (Llama Guard), our method outperforms it in terms of defending the attacks. Moreover, we can analyze the performance where there is no ambiguous predictions by our model, for example, for JB-Chat and RS attack, our method predicts 100% data as harmful but Llama Guard prompt classifier fails to detect various harmful prompts and attains high ASR. This points out that even without any ambiguous predictions, the summary model has produced finer details and had achieved much better performance than an LLM prompt classifier. Moreover, to analyze the effect of ambiguous harmful class, we can examine the PAIR attack here, where ambiguous class was frequently predicted and our method has surpassed Llama Guard in this case as well highlighting the success of both the fine-grained detail deduction and the cautious handling of ambiguous class with caution message. We further discuss the effectiveness of the ambiguous harmful class in Section 6.5.

Attack	Defense	Vicuna	Llama-2	GPT-4
PAIR	No Defense	69%	0%	34%
	SmoothLLM	55%	0%	19%
	Perplexity Filter	69%	0%	30%
	Erase-and-Check	**0**%	0%	1%
	Llama-Guard (Prompt)	39%	0%	13%
	Summary The Savior	22%	**0%**	**0%**
GCG	No Defense	80%	3%	4%
	SmoothLLM	4%	0%	4%
	Perplexity Filter	3%	1%	0%
	Erase-and-Check	17%	1%	2%
	Llama-Guard (Prompt)	13%	0%	0%
	Summary The Savior	**0%**	**0%**	**0%**
JB-Chat	No Defense	90%	0%	0%
	SmoothLLM	73%	0%	0%
	Perplexity Filter	90%	0%	0%
	Erase-and-Check	1%	0%	0%
	Llama-Guard (Prompt)	4%	0%	0%
	Summary The Savior	**0%**	**0%**	**0%**
Prompt with RS	No Defense	89%	90%	78%
	SmoothLLM	68%	0%	56%
	Perplexity Filter	88%	73%	70%
	Erase-and-Check	24%	25%	10%
	Llama-Guard (Prompt)	45%	39%	48%
	Summary The Savior	**0%**	**0%**	**0%**

Table 3: Attack Success Rates (ASR) of various methods on various LLMs along with different defense techniques.

6.5 Ambiguous Harmful Effectiveness

We have evaluated the performance of the inclusion of the ambiguous harmful class by focusing on the ambiguous harmful predictions by our model on the PAIR attacks of the JBB dataset. For each of the LLMs, first we have extracted the the number of ambiguous harmful class predictions on PAIR attacks and then we evaluated the attack success rates if the ambiguous class is defined as harmless (No Defense), harmful (Ambiguous Blocked), and pass with caution (Summary The Savior). Since there were no ambiguous data in PAIR attacks for Llama-2 7B model, so we utilized the prompts designed for GPT-3.5 to apply on Llama-2 for this particular analysis. The results are illustrated in Table 4. We can see that if we do not defend these data then this would cause 100%, 7% and 76% attacks in Vicuna, Llama-2 and GPT-4 respectively. However, with our method, we can reduce the ASR to 0% for both Llama-2 and GPT-4 and for Vicuna the ASR is 29% as it lacks a proper safe-guarded training to follow caution. Now if we block all the ambiguous data by denoting them as harmful then we will get 0% ASR as shown by Ambiguous Blocker method in the Table 4. However, this would increase the refusal rate on benign but sensitive data and we show the phenomenon in Table 5 where we compare the refusal rates of Vicuna and Llama on benign but sensitive data. For this refusal rate analysis, we have utilized 200 Wild Teaming at Scale dataset prompts from our evaluation dataset where the data are mostly benign but some of them sensitive but not harmful and 198 prompts are classified fully safe by Llama Guard. The refusal analysis has been judged by GPT-4. Now, our method predicts 8% of this data as ambiguous and without any defense the refusal rates of Vicuna and Llama are 3% and 7%. With our method, Summary The Savior, passing the ambiguous data with caution message does not increase the refusal rates. However, if we block all the ambiguous data (Ambiguous Blocker) then the refusal rate will go to 9% and 12% respectively for Vicuna and Llama-2. And that is why the ambiguous class with caution is so handy that it not only reduces the attack success rates but also it does not increase the rejection rate on benign data and maintains helpfulness. Any model with only two classes would have either increased the attack success rate by denoting some of the ambiguous data as harmless (for example, Llama Guard on the PAIR attacks in Table 3) or would have increased the refusal rates of LLMs by blocking some ambiguous as harmful because of false positives or would have done both. However, with the inclusion of ambiguous class, we have achieved the lowest attack success rate while maintaining helpfulness and no change in refusal rates.

Type	Mode	Vicuna	Llama-2	GPT-4
PAIR	No defense	100%	7%	76%
	Ambiguous Blocker	0%	0%	0%
	Summary The Savior	**29%**	**0%**	**0%**

Table 4: Attack Success Rate (ASR) of ambiguous harmful data with different defense modes.

Dataset	Method	Safety	Rejection Rate	
			Vicuna	Llama
Wild Teaming at Scale	No Defense	100%	3%	7%
	Ambiguous Blocker	100%	9%	12%
	Summary The Savior	**100%**	**3%**	**7%**

Table 5: Rejection Rates for different LLMs with various defense techniques.

7 Limitations

Llama-3.1 70B model has been used to generate summaries and label the data and this model has its own limitations. Again, GPT-4 has been leveraged to extract harmful keywords from prompts to generate the harmful keywords dataset. Human involvement can make the data generation process more reasonable. Lack of human involvement can incorporate mispredictions that the model cannot infer. For future work, we aim to advance this process by incorporating both human and various LLMs for data generation and labeling. In addition, we plan to explore the integration of the summary generation into vision-language models to access its applicability in such scenarios.

8 Conclusion

We introduce Summary The Savior, which analyzes the security aspects of the prompts while generating the summary and defends jailbreaking. Moreover, we introduce keyword and query-based analysis to put focus on the harmful parts of the prompt while generating summaries. In addition to that, we have also introduced an additional class called ambiguous harmful to cater ambiguous prompts that can be harmful in different contexts. Through our comparative analysis, we show that our method defends state of the art LLM jailbreak methods namely PAIR, GCG, JB-Chat and Prompt with Random Search. Unlike existing methods, our Summary

The Savior method is not dependent on any attack and achieves lowest attack success rates compared to state-of-the-art defenses across various attacks in Vicuna, Llama-2 and GPT-4. Moreover, the inclusion of ambiguous harmful class provides a good balance between attack defense and helpfulness.

References

Alex Albert. 2023. Jailbreak chat. https://www.jailbreakchat.com. Accessed: 2025-01-08.

Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. 2024. Jailbreaking leading safety-aligned llms with simple adaptive attacks. *arXiv preprint arXiv:2404.02151*.

Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J Pappas, Florian Tramer, et al. 2024. Jailbreakbench: An open robustness benchmark for jailbreaking large language models. *arXiv preprint arXiv:2404.01318*.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J Pappas, and Eric Wong. 2023. Jailbreaking black box large language models in twenty queries. *arXiv preprint arXiv:2310.08419*.

Justin Cui, Wei-Lin Chiang, Ion Stoica, and Cho-Jui Hsieh. 2024. Or-bench: An over-refusal benchmark for large language models. *arXiv preprint arXiv:2405.20947*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Junxian He, Wojciech Kryscinski, Bryan McCann, Nazneen Rajani, and Caiming Xiong. 2022. CTRLsum: Towards generic controllable text summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5879–5915, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, et al. 2023. Llama guard: Llm-based input-output safeguard for human-ai conversations. *arXiv preprint arXiv:2312.06674*.

Neel Jain, Avi Schwarzschild, Yuxin Wen, Gowthami Somepalli, John Kirchenbauer, Ping-yeh Chiang, Micah Goldblum, Aniruddha Saha, Jonas Geiping, and Tom Goldstein. 2023. Baseline defenses for adversarial attacks against aligned language models. *arXiv preprint arXiv:2309.00614*.

Liwei Jiang, Kavel Rao, Seungju Han, Allyson Ettinger, Faeze Brahman, Sachin Kumar, Niloofar Mireshghallah, Ximing Lu, Maarten Sap, Yejin Choi, et al. 2024. Wildteaming at scale: From in-the-wild jailbreaks to (adversarially) safer language models. *arXiv preprint arXiv:2406.18510*.

Dylan Lee, Shaoyuan Xie, Shagoto Rahman, Kenneth Pat, David Lee, and Qi Alfred Chen. 2024. “Prompter says”: A linguistic approach to understanding and detecting jailbreak attacks against large-language models. *LAMPS'24*, page 77.

Yi Liu, Gelei Deng, Zhengzi Xu, Yuekang Li, Yaowen Zheng, Ying Zhang, Lida Zhao, Tianwei Zhang, Kailong Wang, and Yang Liu. 2023. Jailbreaking chatgpt via prompt engineering: An empirical study. *arXiv preprint arXiv:2305.13860*.

Matthew Pisano, Peter Ly, Abraham Sanders, Bingsheng Yao, Dakuo Wang, Tomek Strzalkowski, and Mei Si. 2023. Bergeron: Combating adversarial attacks through a conscience-based alignment framework. *arXiv preprint arXiv:2312.00029*.

Alexander Robey, Eric Wong, Hamed Hassani, and George J Pappas. 2023. Smoothllm: Defending large language models against jailbreaking attacks. *arXiv preprint arXiv:2310.03684*.

Paul Röttger, Hannah Rose Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. 2023. Xstest: A test suite for identifying exaggerated safety behaviours in large language models. *arXiv preprint arXiv:2308.01263*.

Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. 2024. " do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 1671–1685.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, 3(6):7.

Xianjun Yang, Yan Li, Xinlu Zhang, Haifeng Chen, and Wei Cheng. 2023. Exploring the limits of chatgpt for query or aspect-based text summarization. *arXiv preprint arXiv:2302.08081*.

Jiahao Yu, Xingwei Lin, Zheng Yu, and Xinyu Xing. 2023. Gptfuzzer: Red teaming large language models with auto-generated jailbreak prompts. *arXiv preprint arXiv:2309.10253*.

Haopeng Zhang, Xiao Liu, and Jiawei Zhang. 2023. Summit: Iterative text summarization via chatgpt. *arXiv preprint arXiv:2305.14835*.

Yubo Zhang, Xingxing Zhang, Xun Wang, Si-qing Chen, and Furu Wei. 2022a. Latent prompt tuning for text summarization. *arXiv preprint arXiv:2211.01837*.

Yubo Zhang, Xingxing Zhang, Xun Wang, Si-qing Chen, and Furu Wei. 2022b. Latent prompt tuning for text summarization. *arXiv preprint arXiv:2211.01837*.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*.