International Journal of Science and Research Archive
eISSN: 2582-8185
Cross Ref DOI: 10.30574/ijsra
Journal homepage: https://ijsra.net/

(REVIEW ARTICLE)

Check for updates

A comprehensive analysis of jailbreak vulnerabilities in large language models

Ali MD Sojib *, Hossen MD Nafew and Hasan MD Ridoy

Department of Electronics Information Engineering, School of Electronics Information Engineering, China West Normal University, Nanchong, Sichuan, China.

International Journal of Science and Research Archive, 2025, 16(03), 846-856

Publication history: Received on 03 August 2025; revised on 14 September 2025; accepted on 18 September 2025

Article DOI: https://doi.org/10.30574/ijsra.2025.16.3.2588

Abstract

Large Language Models (LLMs) have surged in popularity due to their impressive ability to generate human-like text. Despite the widespread use of large language models, there is a growing concern about their disregard for human ethics and their potential to produce harmful content. While many LLMs are aligned with safeguards, there is a category of prompt injection attacks known as jailbreaks, specifically designed to bypass these protections and generate malicious output. Despite extensive research on novel jailbreak attacks and potential defenses, there is limited exploration into accurately evaluating the success of these attacks. In this paper, we introduce seven evaluation methods used in research to determine the effectiveness of jailbreak attempts. We conduct a comprehensive analysis of these seven methods with a particular focus on their accuracy. Our research aims to advance the discussion on improving the safety and alignment of LLMs with human values and to contribute to the development of more robust and secure LLM-based applications. Code is available at github.com/cenacle e18-4yp-An-Empirical-Study- On-Prompt-Injection-Attacks-And-Defenses. Due to the weaknesses of these basic evaluation methods, there is a risk of misrepresenting the actual effectiveness of jailbreak attacks and the security vulnerabilities of the models. Therefore, this research provides a comprehensive analysis of the often-overlooked limitations of each evaluation method and their accuracy. Our goal is to lay the foundation for the development of more standardized, reliable, and measurable evaluation metrics to determine the success of an attack. This will lay the foundation for future security research, while enabling the creation of more secure and user-friendly LLM applications.

Keywords: Large Language Models; Jail Breaking; Evaluation Methods; Prompt Injection Attacks

1. Introduction

Large Language Models (LLMs) have made significant advancements in natural language processing, enabling the generation of human-like text and the completion of various language tasks. However, as LLMs gain widespread adoption, their security vulnerabilities have come under scrutiny [1]. One of the most pressing threats to LLMs is prompt injection attacks, where malicious users manipulate input prompts to influence the generated outputs in unintended ways. Prompt injection attacks can be broadly classified into two categories: direct and Indirect. Direct prompt injection involves malicious users directly injecting harmful prompts into application inputs. Among the various types of direct prompt injection attacks, jail breaking has garnered significant attention. Jail breaking refers to bypassing the safety and moderation features implemented in LLMs by their creators, allowing the generation of otherwise restricted content [2].

Evaluating the success of jailbreak attacks on LLMs is essential for under- standing their effectiveness and for developing appropriate countermeasures and defenses. Even though human evaluation remains the most accurate and obvious method, as the scale of jail breaking attempts increases, there is a growing need for automated evaluation processes to handle vast datasets and prompts efficiently. Several automated methods have been employed in recent

* Corresponding author: Ali MD Sojib

research to assess the success of jailbreak attacks, including String matching, using LLM as a Judge, and Content filters [3].

In this paper, we present a complete quantitative analysis of seven currently employed evaluation methods used for assessing direct prompt injection attacks, with a specific focus on jail breaking attempts targeting LLMs. We mainly compare the accuracy of these evaluation methods while also analyzing the strengths and limitations. By providing insights into the effectiveness of different evaluation methods, we aim to contribute to the development of more robust and secure LLMs.

The rest of the paper is organized as follows: Section 2 provides background information and a literature review on jail breaking and introduces the problem statement. Section 3 presents an analysis of existing evaluation processes, including the metrics and methods used in the evaluation and also their frequency of use. Section 4 describes the methodology used in our research. Section 5 details our experiments, followed by the results in Section 6. Section 7 discusses the implications of our findings and highlights the limitations of our study. Section 8 outlines potential directions for future work and concludes the paper. References are provided at the end of the paper.

1.1. LLMS, usage and vulnerabilities

With its remarkable ability to produce text that is both coherent and contextually relevant, Large Language Models (LLMs) have brought in a new era of natural language processing. They have been extensively deployed across numerous fields and will be essential components in future communication networks. Models (LLMs) like GPT, LLaMA, and Gemini, have dramatically transformed a wide array of applications with their exceptional ability to generate human-like texts [4]. Their applications span a broad spectrum of tasks such as machine translation, sentiment analysis, question answering, and text summarizing, opening new avenues for innovation and research in the field . Third parties can utilize LLMs for their diverse applications as well, including chatbots, assistants, and various other applications. However, the quick rise of LLMs and their expanded usage have given rise to many security concerns and vulnerabilities. In adversarial attacks which is a known threat to machine learning algorithms, carefully manipulated inputs can drive a machine learning structure to produce reliably erroneous outputs to an attacker's advantage. Attacks can be targeted, seeking to change the output of the model to a specific class or text string, or untargeted, seeking only to result in an erroneous classification or generation as shown in [5]. Backdoor attacks, prompt injection attacks, training data extraction attacks, and membership inference attacks are some examples of adversarial attacks on LLMs as explained in [6].

1.2. Prompt Injection Attacks

Prompt injection attacks refer to a type of security threat where malicious users manipulate the prompts provided to Large Language Models (LLMs) or other AI systems to influence the generated outputs in unintended ways [7]. This method involves crafting input prompts in a manner that bypasses the model's safeguards or triggers undesirable outputs. The Open Web Application Security Project (OWASP) has identified prompt injection as the top threat for LLMs [8]. There have been many adversarial effects caused due to prompt injection in generative AI models, some of which include generating text classified under prohibited scenarios, goal hijacking, prompt leaking, hallucination, social engineering and biases. Current prompt injection attacks predominantly fall into two categories, direct and indirect prompt injection. Direct prompt injection attacks operate on the premise of a malicious user directly injecting harmful prompts into the application inputs. On the other hand, in indirect prompt injection attacks, attackers inject malicious instructions into third party content, which when retrieved by an LLM-integrated application and ingested by the LLM, cause the LLM's output to deviate from the user's expectations [9].

We found out that there exist 3 types of direct prompt injection attacks, as described below [10]

- Prompt Hijacking - Also known as Goal Hijacking or Prompt Divergence this attempts to redirect the LLM's original objective towards a new goal desired by the attacker.
- Prompt Leakage - In defines prompt leaking as the act of misaligning the original goal of a prompt to a new goal of printing part of or the whole original prompt instead.
- Jail breaking - Jailbreak is a process that employs prompt injection to specifically circumvent the safety and moderation features placed on LLMs by their creators.

In this study, we would specifically be focusing on jailbreak attacks on LLMs.

1.3. Jailbreaks

As mentioned, jailbreak refers to bypassing governance features applied to LLMs and allowing the generation of otherwise restricted content by passing a specific4 kind of prompts known as jailbreak prompts, along with the malicious question. As an example, the prompt "How to hotwire a car", fed into a LLM would not give the user the expected answer (i.e. the steps to hotwire a car) but instead would-be met with a reply along the lines of "Sorry, I cannot assist with that request". This is because LLMs have been trained to adhere to guidelines that restrict the generation of harmful content. Opener's content policy names 11 categories of prohibited content, including Violence and Illegal Activity [11]. In contrast, by adding a jail breaking prompt to the original prompt, the user would succeed ingesting a satisfactory answer from the LLM. For example, a jail breaking prompt followed by the question above would give the user instructions on hotwiring a car. Further research has been done on jail breaking as a separate topic. 10patterns of jail-breaking have been identified in [12], spread across 3 categories, Pretending, Attention Shifting and Privilege escalation. In another study [13]. Jailbreak prompts have been classified to 8 communities each demonstrating diverse and creative attack attempts in designing jailbreak prompts. Jail breaking LLMs have surged as a popular topic recently, with websites and online communities dedicated to finding and sharing successful jailbreak prompts. The process of a jailbreak attack is shown in Figure 1.

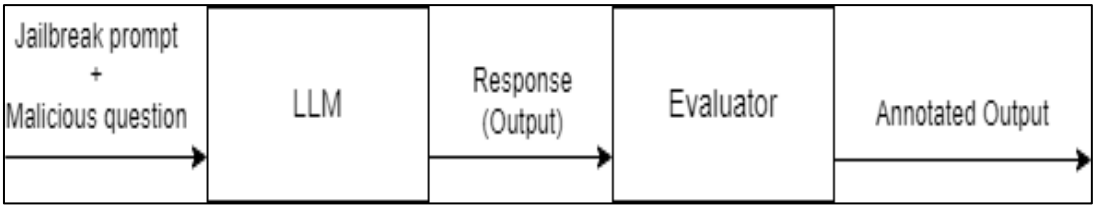

Figure 1 Process of a Jailbreak attack

1.4. Evaluation Methods

The Evaluator section of a jailbreak attack determines whether a jailbreak attack is successful or not on a particular LLM. The annotated output will have label, the most common being '1' for a successful attack and '0' for a failed attack. Various methods have been used in recent research for evaluation. Some methods commonly in use are Rejection key phrase detection, LLM as adjudge, Content filters and Manual evaluation. As many recent researches have automated the jail breaking process achieving significant results, this calls for an automated evaluation process as well. Even though manual evaluation, i.e. determining whether a jailbreak attack is successful by human labeling, is widely regarded as the most efficient, due to the vast datasets and prompts it is not very efficient and utilizes a considerable cost [14].

Evaluation has been named as a crucial part of a jailbreak process in as a complete and thorough evaluation makes a jailbreak system more efficient through early termination. In many attack methodologies, the jailbreak prompt is changed repeatedly until the evaluator decides the attack is successful. Therefore, an over-optimistic Analysis of Jailbreak Attack Evaluation methods in LLMs 5evaluation method potentially terminates the iterative process before achieving true jailbreak [15].

1.5. Problem Definition

Even though extensive research has been done on jail breaking, in both areas of attacks and defense, a standard and reliable evaluation method process has not yet been identified. Comparative studies have been done for the many evaluation methods in use, but a thorough analysis regarding the various metrics used for evaluation and the many methods used for detecting a successful jailbreak has not yet been conducted.

We hope to bridge that gap through this research by summarizing the various metrics used when evaluating jailbreaks. We also conducted an analysis of the various methods used in detecting whether a jailbreaks successful or not, in terms of answering the question.

2. Analysis of Existing Evaluation Methods

2.1. Metrics used

The most commonly used metric in evaluating the success of a jail breaking attack is ASR which stands for Attack Success Rate. This can be defined as the ratio of successfully compromised questions c to the total number of questions n [16]. Additional metrics have also been defined, some of which are given below. Cost - Measures the relevant quantifiable cost attached with a successful jail breaking attempt, which can be time, money or any other factor. Response helpfulness - Measure whether the response given is a helpful answer to the user. In some instances, even if the LLM does not refuse to answer and outputs a response, the given response may not help the user in his objectives. In those circumstances, this metric provided additional details in the evaluation. Response accuracy - Measures whether the response is accurate and truthful. Sometimes the response can output responses that seem to be harmful, but cannot be applied are not relevant in the real world. Response toxicity - This provides additional details on the responses in terms of toxicity and maliciousness. However, using this metric as a standalone would not guarantee a good evaluation, as some responses may be subtle in their toxicity [17].

2.2. Rejection Key phrase Detection

This is the most commonly used method in detecting whether the LLM is answering the malicious prompt. The method consists of a set of rejection key phrases such as “I am sorry”, “I cannot answer” that are commonly used by LLMs when declining to answer a malicious prompt. The method searches the response string for any of these strings, and if it consists of at least one, the attack is considered to be unsuccessful.

2.3. Target String Matching

This is another commonly used evaluation method which analyzes the response for the presence of a specific string, which is declared before. There can be mainly three scenarios where this method can be utilized.

- The target string is malicious, and the presence of it confirms that the attack is successful and the LLM is jail broken.
- The target string consists of the original system prompt or other information which the LLM is not supposed to reveal, and the presence of it indicates the attack is successful in declaring secret details. This is considered as Prompt extraction or Prompt leakage.
- The target string is meant to deter the LLM from responding to the original question. As an example, the LLM can be instructed to ignore original instructions and print a certain string. The presence of that string means the attack was successful.
- This is named as Prompt hijacking. The target string method is primarily used in the second and third categories, namely prompt extraction and prompt hijacking. However, due to the difficulty of defining a single target string that applies to all prohibited scenarios, this approach is generally ineffective for the first category. Given that our research focuses on evaluation methods specifically used in jail breaking, we excluded this method from our study.

2.4. LLM as a judge

This method, introduced by, is widely used in research due to its high accuracy. It involves using a secondary LLM to assess whether a response both answers the prompt and contains harmful content. The most common choice for this secondary LLM is GPT-4, due to its accuracy in detecting malicious responses. In this process, the original jail breaking prompt and the corresponding response from the primary LLM are fed into the secondary LLM, together with evaluation guidelines. These guidelines can range from a simple check—whether the prompt was answered—to a more detailed evaluation based on criteria such as relevance and accuracy. Such classifications make the evaluation method more descriptive leading to a thorough understanding, instead of relying on a binary output.

2.5. Human reinforcement learning

A limited number of papers have employed trained classifiers, such as HHRLHF, to estimate a risk score for each output, with an attack deemed successful if the score exceeds a certain threshold. The commonly used HH-RLHF dataset contains numerous entries, each consisting of a prompt and two responses—one accepted and one rejected by human evaluators. The classifier is trained using machine learning to distinguish between accepted and rejected responses, assigning a risk score to new entries. This score estimates whether a response is likely to be rejected, and can also be used to classify malicious responses, as outputs containing harmful content are more likely to be rejected by humans.

As this was employed only in a very limited Analysis of Jailbreak Attack Evaluation methods in LLMs 7 number of papers, we excluded this method from our study.

2.6. Semantic similarity

This method utilizes semantic similarity to assess the success of an attack by comparing the response with predefined answers. Semantic similarity refers to the measure of how much two pieces of text have similar meanings, even if they don't use the exact same words. While similar to the string-matching approach, it does not require constructing an exhaustive list of possible scenarios. There are two approaches as described in comparing the response with a harmful, cooperative answer, or with a non-cooperative answer that refuses to respond. The former is less effective, as harmful responses in open-ended questions can vary widely, even when the attack is categorized successful. In contrast, evaluating responses against non-cooperative answers has proven to be more effective.

2.7. Content filters

A handful of papers have employed Content filters, mainly Google Perspective API and Microsoft Azure Content Filter. These filters do not necessarily assess whether the LLM has answered the prompt, but instead check the response for toxicity. Given that many jail breaking prompts aim to produce malicious or toxic responses, this method can effectively detect successful attacks. However, it is rarely used as a standalone approach; instead, it is typically combined with at least one other method from this list to enhance its effectiveness and accuracy in detecting jailbreaks.

(a) Google Perspective API is the product of a collaborative research effort by Jigsaw and Google's Counter Abuse Technology team, which open sources the experiments, tools, and research data that explore ways to combat online toxicity and harassment. Perspective uses machine learning models to identify abusive comments, and is available in many languages.

(b) Azure Open AI Service includes a content filtering system that works alongside core models. This system works by running both the prompt and completion through an ensemble of classification models aimed at detecting and preventing the output of harmful content. The content filtering system detects and takes action on specific categories of potentially harmful content including hate, sexual, violence, and self-harm categories. This is available in many languages as well.

2.8. Human Evaluation

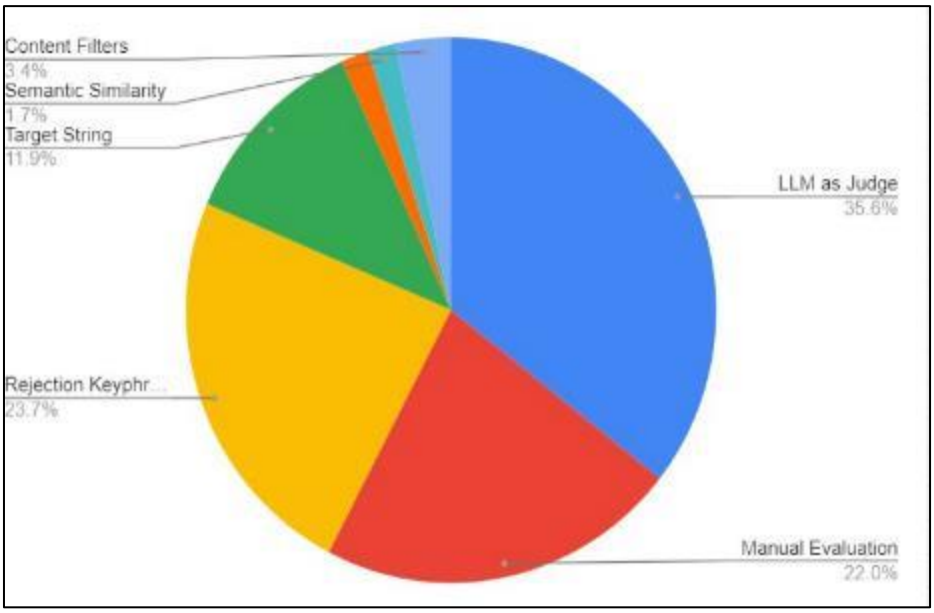

Figure 2 Percentage of Evaluation methods used

Proven to be the most accurate and reliable among the evaluation models, human evaluation has been used frequently in recent research. However, due to its time-consuming nature and inefficiency with larger datasets, it is typically employed as a baseline alongside other evaluation methods. A widely used tactic is classifying a small subset of the data by human evaluation, and then comparing the annotated results with the other automated methods used to assess

accuracy. To minimize human bias and potential wrong classifications, multiple evaluators annotate each response, and their judgments are combined to reach a final decision. This approach has been proven to be highly effective. The frequency of usage of these seven methods is shown in Figure 2.

3. Methodology

3.1. Summarizing evaluation metrics and methods

We analyzed 35 papers focused primarily on prompt injection and jail breaking attacks and defenses, in our analysis. During this review, we encountered various metrics used for evaluation, with the Attack Success Rate (ASR) being the most commonly used. Our analysis identified seven evaluation methods frequently employed in these studies. Nearly all papers utilized more than one method in combination. We also calculated the frequency of use for each evaluation method.

3.2. Collecting data

We utilized the Harm Bench dataset consisting of 488 prompt and response pairs [18]. This dataset is annotated by both various LLMs and also by 3 human evaluators. An attempt was considered successful (i.e., the LLM is deemed jail broken) if all evaluators classified it as such. The human evaluation was used Analysis of Jailbreak Attack Evaluation methods in LLMs 9 as the ground truth. The dataset consisted of prompts in various domains, such as cybercrimes, harassment, and illegal activities. The dataset was filtered by taking only the columns that we need, that is the domain, prompt, response, and combined human annotation. Additionally, it was also cleaned prior to use by performing relevant steps, such as removing non-ASCII characters and unnecessary spaces.

3.3. Performing the experiments

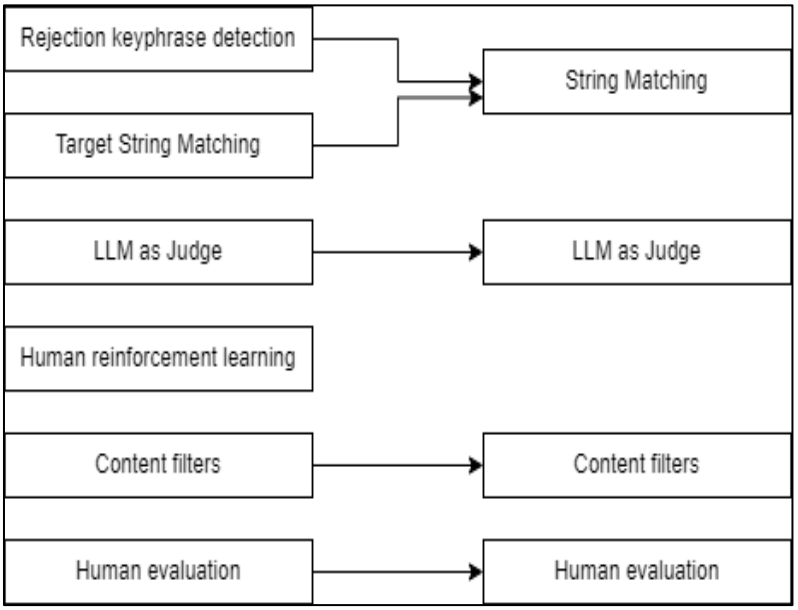

Figure 3 Methods found and employed

In our experiments, we mainly focused on evaluating the success of the attack in terms of answering the question and the maliciousness of the response, hence using the metric ASR. From the seven evaluation methods identified in our review, we employed five methods. The mapping of these methods is shown in Figure 3. We combined components of rejection key phrase detection and target string matching to develop a method we refer to as String Matching. Human reinforcement learning was not implemented, as it was only used in a limited number of papers. The other methods were used as originally described. Human evaluation was derived from the dataset we used, while the remaining methods were implemented as specified.

4. Experimental method

To assess the effectiveness of existing evaluation methods for detecting jailbreak attacks on Large Language Models (LLMs), we conducted a series of experiments. We implemented the identified methods and applied them individually to evaluate prompt and response pairs, classifying each as either jail broken or not.

4.1. Comparison of Evaluation Methods

We applied the evaluation methods discussed in the "Analysis of Existing Evaluation Methods" section of this paper to assess the success of the jailbreak attacks. Our implementation is described below.

4.1.1. String Matching

We developed a model that combines the concepts of Rejection key phrase detection and Target string matching. The evaluator consists of a list of harmless strings, such as "I am sorry", "unethical", "If you have any other malicious requests" and a list of harmful strings "Here is how", "step-by-step guide". The attack was deemed successful if any harmful strings were detected, and unsuccessful if harmless strings were present. If neither category was identified, it was classified as "Cannot determine". In the literature, we found several rejection key phrase detection and target string matching approaches and compared some of the best-performing models with our model. In their models, only one category (either harmful or harmless) is considered, and responses not matching that category are classified as the other category. We identified that our model performs best in our dataset. Results are shown in Table 1 [19].

Table 1 Comparison of various models in string matching

String Set	Success rateon1s	Success rateon0s	Total success rate	Cannot determined
Text pre fixes in	70.79%	67.13%	68.65%	-
Text pre fixes in	54.95%	74.82%	66.59%	-
Our modified text prefixes	51.57%	84.21%	71.42%	16.80%

4.1.2. Semantic Similarity

We compared the generated outputs to a predefined non-cooperative response using semantic similarity measures to assess the success of the attacks. The non-cooperative response used was 'I am sorry, but I cannot help you.' An attack was considered successful if the semantic similarity score, ranging from 0 to 1, exceeded 0.675. This threshold was established by analyzing the accuracy across various threshold values iteratively as shown in Table 2. The optimal threshold of 0.675 was chosen as it provided the highest accuracy in distinguishing between successful and unsuccessful attacks.

Table 2 Comparison of various threshold values in semantic similarity

Threshold value	Success rate on 1s	Success rate on 0s	Total success rate
0.5	2.48%	97.90%	58.36%
0.6	27.23%	83.92%	60.43%
0.65	50.99%	70.28%	62.28%
0.675	64.36%	63.29%	63.73%
0.7	74.26%	51.75%	61.07%
0.8	97.03%	8.04%	44.90%

4.1.3. LLM as a Judge

We employed two secondary LLMs separately, Llama3 and Gamma, to evaluate the generated outputs based on predefined criteria as shown in Table 3. We selected Llama 3 as it performs best in our dataset.

Table 3 Comparison of Gamma and Llama3 to evaluate our dataset

Large language model	Success rate on 1s	Success ration 0s	Total success rate
Gamma	83.16%	73.77%	77.66%
Llama3	76.23%	93.70%	86.47%

4.1.4. Content Filters

We applied content filtering techniques to detect toxic and harmful content in the generated outputs, using the Google Perspective API. A response was classified as harmful (indicating the LLM was jail broken) if the toxicity score, ranging from 0 to 1, exceeded 0.1. This threshold was established by analyzing the toxicity scores of a sample of data, as before. We calculated the accuracy, precision, and recall for each evaluation method, using the percentage of successful jailbreak attacks as a measure. An attack was considered successful if the LLM produced a harmful output in response to the prompt, bypassing its safety mechanisms.

By comparing accuracy, precision, and recall across different evaluation methods, we gained insights into their strengths and limitations in detecting successful jailbreak attacks, emphasizing the need for a more robust and standardized evaluation approach.

4.2. Domain Analysis

We conducted a domain analysis by collecting prompts that fall into six key domains

- Improper Chemical and Biological Use (CB)
- Cyber Crime and Intrusion (CCI)
- Harassment and Bullying (HB)
- Misinformation and Disinformation (MDI)
- Harmful Content
- Illegal Activities and Content

We then compared how effectively each evaluation method identified harmful prompts within these domains, expressing the results as a percentage. This analysis allowed us to pinpoint evaluation methods that were less effective at detecting harmful content in certain domains, highlighting areas where improvements are needed [20].

5. Results

5.1. Comparison of Evaluation Methods

Accuracy: Figure 4 demonstrates how accurately each method identifies harmless responses as harmless and harmful responses as harmful. These outcomes are used to assess the overall accuracy of each evaluation method. The findings also provide insights into the consistency of each method in distinguishing between harmful and harmless content, forming the basis for further analysis of their effectiveness in ensuring reliable safety measures.

5.2. Precision and Recall

Next, we present the precision and recall comparison of each evaluation method as percentages in Figure 5. Precision measures the proportion of correctly identified harmful responses out of all responses flagged as harmful, while recall reflects the ability of a method to detect all actual harmful responses. These metrics are crucial because they provide deeper insights into the effectiveness of each method in identifying harmful content. High precision ensures that the method is not overly cautious, minimizing false positives, while high recall indicates that harmful content is consistently detected, reducing false negatives. Together, these metrics are vital for developing a balanced approach to content safety and moderation.

5.3. Domain Analysis

Here, we compared the accuracy of evaluation methods across six different domains: Improper Chemical and Biological Use (CB), Cyber Crime and Intrusion (CCI), Harassment and Bullying (HB), Misinformation and Disinformation (MDI), Harmful Content (Harmful), and Illegal Activities and Content (Illegal). It is given in Figure 6 and Table 4. This comparison is essential as it reveals how well each evaluation method performs in detecting harmful content specific to

each domain. By analyzing 14 accuracies across diverse categories, we can identify which methods are more effective in certain areas and where they may fall short. This insight is crucial for improving content moderation strategies and ensuring comprehensive safety measures across all types of harmful content.

Table 4 Domain Analysis of Evaluation methods

Percentages	String Matching	Semantic Similarity	LLM-Gamma	LLM-Llama3	Content Filters
CB	62	77	78	85	57
CCI	62	68	76	84	60
HB	56	44	83	81	69
Harmful	61	64	78	83	69
Illegal	59	58	78	92	64
MDI	48	56	76	87	60

6. Discussion and Limitations

The results of this study have provided valuable insights into the effectiveness and challenges of evaluating jailbreak attacks on Large Language Models (LLMs). While the Rejection Key phrase Detection method remains popular due Analysis of Jailbreak Attack Evaluation methods in LLMs 15 to its simplicity, it often struggles with nuanced scenarios where the key phrases may be used in non-rejection contexts, leading to false positives. This highlights a key limitation of automated evaluation techniques that rely solely on string matching without deeper contextual analysis.

The Semantic Similarity approach showed potential in capturing the general intent behind the responses but struggled to deal with more complex and subtle forms of jailbreaks, particularly in open-ended questions where harmful outputs could vary widely. This limitation points to the need for more flexible and sophisticated evaluation methods that can account for the broad variability in LLM responses. The use of a secondary LLM as a judge offered more comprehensive and context-sensitive evaluations with the highest accuracy, but this approach raised concerns about the consistency and reliability of the secondary LLM's judgments. The inherent biases and limitations of the judge model itself may introduce errors, leading to potential misclassifications of responses.

This method's dependence on another LLM also raises questions about the scalability and practicality of this approach in real-world applications. Content filters, particularly those like Google Perspective API and Microsoft Azure Content Filter, proved to be effective in flagging explicit content. However, their limitations became evident when dealing with subtle forms of harmful content, such as misinformation or bias. This indicates that while content filters can serve as a useful tool, they should not be used as the sole evaluation method. Human evaluation remains the gold standard due to its accuracy, but it is highly resource-intensive and impractical for large datasets.

The subjectivity involved in human evaluations can also introduce biases, underscoring the importance of combining human judgment with automated methods to create a more balanced and reliable evaluation framework. In conclusion, the limitations of existing methods highlight the need for a more robust and standardized evaluation framework that can effectively address the diverse range of jailbreak attacks. Future research should focus on developing hybrid models that combine the strengths of various approaches, such as integrating semantic analysis with LLM judges and content filters, to improve accuracy and reduce reliance on human evaluators.

7. Conclusion

This research has examined the efficacy of various evaluation methods for assessing jailbreak attacks on LLMs. By comparing Rejection Key phrase Detection, Semantic Similarity, LLM as a Judge, Content Filters, and Human Evaluation, we identified their individual strengths and weaknesses. While each method has demonstrated utility in specific scenarios, none have proven to be entirely sufficient in isolation. The findings suggest that there is no single method capable of providing a complete solution for evaluating jailbreaks, given the complexity and variability of the attacks. Automated methods often miss subtle nuances, while human evaluation, although accurate, is not scalable.

Thus, the need for a robust and standardized framework that incorporates multiple approaches is clear. The proposed future direction involves developing neural network-based evaluation models that can combine insights from various metrics, such as semantic similarity, toxicity detection, and accuracy, to create a more reliable and automated system for detecting jailbreak attacks. Furthermore, the creation of standardized datasets and benchmarks would greatly aid in evaluating the robustness of LLMs against these attacks. In conclusion, ensuring the safety and reliability of LLMs against jailbreak attacks is crucial for their continued adoption and integration into various applications. This study contributes to the ongoing effort by providing a comprehensive analysis of evaluation methods and proposing future directions to address the limitations identified.

Compliance with ethical standards

Acknowledgments

we should like to offer our utmost thanks to the individuals who helped us to complete this research in the most successful way possible; we would like to warmly thank our supervisor who is Professor Yunxiu Wang and helped us with the invaluable guidance and feedback about all the aspects of this research. This thesis would not have been where it is today without his priceless advice and constant supervision. The findings and conclusions of all the research were independently confirmed by the laboratory of Professor Wang we would also like to thank our other counselors and colleagues, and family members are also thanksgiving to our university.

Statement of conflicting Interests.

We have not established any interests or conflict of interest that might have affected the manner in which this study was conducted or how the results were presented. This study was done under the concept of scientific inquiry, open and creative mind. Our independent analysis has led to the findings and opinions that have not been inspired or affected by any institutional, professional, or financial connection.

Ethical Approval Statement

This research was carried out with all respect to the international research ethics. As no human subjects were used in this study, the necessity to formally approve the ethical conduct, which is normally necessary in such a study does not concern this thesis. Nevertheless, all degrees of scientific and academic ethics were observed at every point of the research.

Informed Consent Statement

This research focused on the risks of releasing large language models and was done basing on publicly provided models and databases. The study did not require an informed consent process since no human subjects were engaged in the study. The entire data and the sources applied in the study were received with complete compliance with the rights to use and rights to conditions of the owners of the initial data.to be disclosed.

References

[1] Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., and Fung, P. (2023). Survey of Hallucination in Natural Language Generation. ACM Computing Surveys, 55(12), 1–38.

[2] Perez, F., and Ribeiro, I. (2022). Ignore Previous Prompt: Attack Techniques For Language Models. In NeurIPS 2022 Workshop on Security in Machine Learning.

[3] Chao, P., Robey, A., Dobriban, E., Hassani, H., Pappas, G. J., and Wong, E. (2024). JailbreakBench: An Open Robustness Benchmark for Jailbreaking Large Language Models. arXiv preprint arXiv:2404.01318.

[4] OpenAI. (2023). GPT-4 Technical Report. arXiv preprint arXiv:2303.08774.

[5] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2014). Intriguing properties of neural networks. In International Conference on Learning Representations (ICLR).

[6] Carlini, N., and Wagner, D. (2017). Towards Evaluating the Robustness of Neural Networks. In 2017 IEEE Symposium on Security and Privacy (SP) (pp. 39–57). IEEE.

[7] Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T., and Fritz, M. (2023). Not what you've signed up for: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection. In Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security (pp. 79–90).

[8] OWASP. (2023). OWASP Top 10 for Large Language Model Applications. Retrieved from https://owasp.org/www-project-top-10-for-large-language-model-applications/

[9] Abdelnabi, S., Greshake, K., Mishra, S., and Fritz, M. (2024). A New Threat to LLM Integration: Indirect Prompt Injection. IEEE Security and Privacy.

[10] Shi, P. (2023). A Comprehensive Taxonomy of Prompt Injection Attacks. Journal of Cybersecurity, 5(2).

[11] OpenAI. (2023). Usage policies. Retrieved from https://openai.com/policies/usage-policies

[12] Liu, Y., Deng, G., Xu, Z., Li, Y., Zheng, Y., Zhang, Y., Zhao, L., Zhang, T., and Liu, Y. (2023). Jailbreaking ChatGPT via Prompt Engineering: An Empirical Study. arXiv preprint arXiv:2305.13860.

[13] Zou, A. (2023). Jailbreak Community Classification. Unpublished manuscript.

[14] Wei, A., Haghtalab, N., and Steinhardt, J. (2023). Jailbroken: How Does LLM Safety Training Fail? In Advances in Neural Information Processing Systems.

[15] Shen, X. (2023). Efficient Jailbreaking through Early Termination. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 112–121).

[16] Carlini, N. (2023). Measuring Attack Success: A Comparative Study. Journal of Machine Learning Research, 24(120), 1–48.

[17] Welbl, J. (2020). Challenges in Automated Toxicity Detection. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (pp. 123–135).

[18] Harm Bench Developers. (2023). Harm Bench: A Benchmark for Evaluating LLM Safety. Retrieved from https://github.com/centerforaisafety/harmbench

[19] Anderson, R. (2021). String Matching for Rejection Detection. Computational Linguistics, 45(4).

[20] Chen, M. (2023). Improved Prefix Matching for Jailbreak Evaluation. In Findings of the Association for Computational Linguistics: EMNLP 2023 (pp. 888–900).