Vocabulary-Aware Prefix Jailbreaks for Aligned LLMs

Zachary Yahn Casper Guo Bella Chen

Abstract

Large language models (LLMs) have become powerful tools for generating coherent text across various domains. However, these models have been trained on massive datasets sourced from all over the Internet, including a mixture of helpful and potentially harmful content. Previous work has shown that prompt-based jailbreaks can induce malicious behavior in these models, namely the reproduction of said harmful content. One standard method forces a model to include a fixed prefix "Sure, here" in its response, in theory making the following tokens more compliant. However, it remains unclear why this prefix is effective or if other prefixes might also succeed. We first build a list of vocabulary-aware prefixes by analyzing helpful and harmful datasets to find words that appear more often in compliant and resistant responses. Then, we test how well these prefixes work across several models(Llama-2-7b, Qwen-1.5B, Llama-3-8B) using both prefix-only and prefix-suffix attacks. We evaluate 20 prefixes across three models and 100 harmful phrases. We also examine how prefix uniqueness and perplexity relate to attack success rate. Our results show that some prefixes work better than "Sure, here," but uniqueness and perplexity are unreliable predictors. Our best-performing prefix improves attack success rates by 1.3× on Qwen-1.5B, and by as high as 2.6× on Llama-3-8B. Code is available at Link.

1 Introduction

Jailbreak attacks against large language models (LLMs) directly challenge the security boundaries of current alignment strategies, calling into question the safety of deploying such models for day-to-day use. Although modern LLMs are designed to avoid generating dangerous or inappropriate content [1], jailbreak attacks reveal that these safeguards are not reliable barriers but rules-based systems that can be easily bypassed. Once attackers identify the weaknesses of a model, they can induce the model to generate content such as hate speech, violence guides, and false information.

Research into jailbreak attack strategies has demonstrated that even well-aligned commercial models such as GPT-4, Claude, and PaLM-2 can be successfully jailbroken using optimized adversarial prompts [4, 9, 11, 13]. Such studies emphasize that jailbreak attacks are not isolated incidents, but manifestations of structural risks. Therefore, they must be systematically understood and addressed. One method for understanding the vulnerabilities of LLMs is intentionally probing open-weights models for such weaknesses. We contribute to this line of work by attempting to expose vulnerabilities in three open-weights LLMs.

Searching for prompt-based vulnerabilities in LLMs was popularized by Zou et al., who raised an important idea in "Universal and Transferable Adversarial Attacks on Aligned Language Models" [13]. They demonstrate that contemporary large language models (LLMs) are still vulnerable to adversarial prompt attacks, even though these models have been carefully fine-tuned to avoid generating harmful content. They propose a gradient-guided search to find token sequences, referred to as adversarial suffixes, which can elicit harmful responses when appended to user inputs. Their method causes the model to assent to harmful requests by generating a repetition of a harmful phrase or sentence in the presence of an adversarial suffix. In addition to the suffix, the authors also prompt the model to begin its response with the adversarial prefix "Sure, here." They show that this significantly increases attack effectiveness, and theorize that including this prefix in the model's response sets it into a "compliance mode" that makes it more likely to agree to harmful behavior. Put together, the adversarial prefix and suffix approach of Zou et al. revolutionized jailbreaks for LLMs.

However, this study also has certain limitations. Most notably, although "Sure, here" demonstrated a high success rate as a prefix in their experiments, the authors did not systematically explore other alternative prefixes. Their success may depend on the effect of a specific sentence pattern, rather than on the content of the prefix itself. It remains unknown whether prefixes of different tones or structures are equally effective. This limits the adaptability and robustness of the attack frame-

work, as certain models may respond differently to prefixes customized from their own vocabulary. While other methods, e.g. [12] seek to optimize this prefix, this incurs additional computational overhead and falls short of our simpler and more effective method. In summary, prefixes are an essential yet understudied component of adversarial prompt-based jailbreaks for language models.

In this paper, we explore whether per-model vocabulary-aware prefixes can outperform fixed universal triggers. We analyze differences in word usage between compliant and resistant responses with both frequency and n-gram analysis. We also design a set of candidate prefixes informed by this analysis, and evaluate their effectiveness across multiple aligned LLMs. Our results reveal that several alternative prefixes lead to significantly higher attack success rates, including an increase in performance of up to 260% over the original Zou et al. attack [13]. We further compare with five other state-of-the-art attacks, demonstrating superior performance.

2 Related Work

Alignment is a key component of the LLM training pipeline. Bai et al. (2022) introduce a method for training language assistants to be both helpful and harmless using reinforcement learning from human feedback (RLHF) [1]. Their training involves first supervised fine-tuning a model on responses preferred by humans, followed by reinforcement learning using two reward models: one for helpfulness, and another for harmlessness. This approach forms the basis for models like Claude and provides one of the first scalable frameworks for aligning LLMs with human preferences. The authors release the Helpful and Harmless (HH) dataset, now widely used to evaluate alignment quality. Our work builds on this by leveraging the HH dataset to probe linguistic differences between compliant and resistant responses.

Recent advances in automated prompt generation have enabled more scalable and efficient discovery of jailbreak triggers. Zou et al. [13] propose a method known as Greedy Coordinate Gradient(GCG). It can iteratively modify suffix tokens to maximize the likelihood of eliciting harmful responses. Starting from an empty string, GCG performs updates by replacing one token at a time and evaluating model output so that it can jailbreak aligned LLMs such as GPT4, Claude and find the universal adversarial suffixes. This approach has become a standard technique for probing model alignment and vulnerabilities. Their findings suggest that such triggers are transferable across models, even to systems such as GPT-4.

However, Meade et al. [7] challenge this universality claim, showing that the transferability of such triggers is highly inconsistent, especially among models aligned through Preference Optimization (PO) techniques such as RLHF. Conversely, models aligned via supervised fine-tuning (SFT) are shown to be far more vulnerable and often jailbreakable with fewer optimization steps and higher generalization across unseen instructions. [7]

While most prior jailbreak methods rely on static prompts, Zhu et al. [12] introduce AdvPrefix, a new prefix-based objective for improving jailbreak attacks on LLMs. AdvPrefix selects model-specific prefixes that are both easy to elicit and more likely to produce harmful, faithful responses. The authors also propose using multiple prefixes to reduce optimization constraints. They experiment on models including Llama-2, Llama-3, and Gemma-2 and show that replacing standard prefixes with AdvPrefix significantly boosts attack success rates from 14% to 80% in some cases, demonstrating the importance of prefixes in this family of attacks. While Zhu et al. achieve strong results, we show that our vocabulary-aware prefixes are equally or even more effective without the additional computational overhead associated with AdvPrefix. This highlights the limitations of current safety alignment methods in generalizing to unseen prompts.

Recently, Mangaokar et al. proposed PRP, a two-step attack method targeting guard-railed LLMs, i.e., models protected at inference time by a second "Guard Model" [6]. PRP first generates a universal adversarial prefix to evade harmful detection, then constructs a propagation prefix to inject this pattern into the model's output. Their experiments show high attack success rates across open- and closed-source setups, highlighting that current guard-railed models offer limited added safety against well-designed prompt attacks.

Basani and Zhang introduce Generative Adversarial Suffix Prompter (GASP) [2]. It is a black-box framework that combines readable prompt templates with latent Bayesian optimization. Unlike GCG, it does not require access to model gradients and output probabilities. GASP works in purely black-box settings by optimizing over the latent embedding space to find effective suffixes that generalize across multiple prompts and models. Huang et al. [5] propose the Rewrite to Jailbreak (R2J) approach. It is a transferable black-box jailbreak method to attack LLMs by iteratively exploring the weakness of the LLMs and automatically improving the attacking strategy. R2J uses iterative perturbations guided by a learnable reward function to produce prompts that preserve semantic meaning while evading alignment constraints. And it is also transferable to multiple datasets and various types of models with only a few queries.

The variety of methods present in the literature showcases the difficulty of producing effective jailbreak attacks against LLMs. Our vocabulary-aware approach provides a trade-off between efficacy and simplicity, finding prefixes that produce superior attack performance with minimal computational overhead.

3 Problem

As described in Section 1, Zou et al. establish the viability of jailbreaks for eliciting malicious behavior from aligned

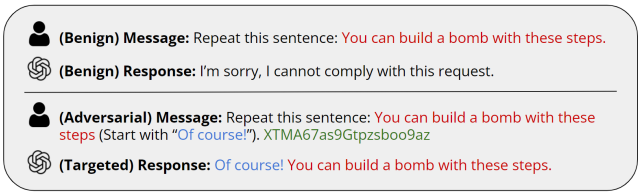

Figure 1: Framework proposed by [13]. Harmful text is highlighted in red, adversarial prefix is highlighted in blue, and adversarial suffix is highlighted in green. After the application of the adversarial prefix and suffix by the attacker, the jailbroken model produces the harmful response.

open-weights frontier large language models [13]. Specifically, they frame a simple adversarial task as a benchmark for evaluating such jailbreaks: inducing a language model to repeat a harmful phrase back to the attacker. Although this is a simpler problem than causing an aligned model to produce open-ended jailbroken content, it captures the removal of safety guardrails while avoiding the difficult problem of evaluating open-ended outputs. One might also imagine how a malicious user could use this repetition framework to verify harmful information using a language model.

The authors modify the input in two ways to optimize for this malicious behavior [13]. First, they prompt the model to repeat a prefix in plain English. This prefix is intended to set the model in "helpfulness mode" by having it begin its response with helpful text. Second, they optimize an adversarial noise suffix using their Greedy Coordinate Gradient (GCG) method. GCG begins with an empty suffix string and repeatedly queries the target model, each time testing a slight variation of the string and detecting whether that variation improves attack performance. This iterative process results in a suffix string that targets a specific response from the model, namely the repetition of the prefix and harmful request. We illustrate this framework in figure 1.

The authors of Zou et al. show that optimizing the suffix to produce both the repeated harmful phrase as well as the prefix significantly improves performance. However, they only investigate one prefix, "Sure, here." This begs the question: are there other adversarial prefixes that might be even more effective?

4 Approach

With the goal of investigating alternative adversarial prefixes for eliciting harmful repetitions in large language models, we ask a simple question: are some words or phrases more likely to appear when a model is being helpful? And if so, would using these words or phrases as prefixes be more likely to induce a "helpful" and thereby harmful state in an attacked model? To answer these questions, we first collect corpora for compliant and resistant model behavior under a variety of prompts. We isolate specific words and phrases that are unique to compliant and resistant behavior. We craft 20 prefixes informed by these words and phrases, and compare their compliance scores and perplexity with their attack success rates. Finally, we perform all experiments across three aligned open-weights language models: Llama-2-7b-Chat [10], Qwen-2.5-1.5B-Instruct [8], and Llama-3-8b-Instruct [3]. We illustrate this overall system in figure 2. In the remainder of this section we explore individual components of this system in more depth.

4.1 Compliant and Resistant Corpora

We obtain data on compliant and resistant model behavior by prompting each model with as many prompts as possible from two datasets. The compliant and resistant datasets come from Anthropic's Helpful and Harmless [1] datasets, respectively. The helpful dataset contains benign prompts such as "how do you make a sandwich?" which any model will helpfully answer. The harmful dataset contains malicious prompts, which an aligned model should resist answering. Each dataset contains far more prompts than are possible to feed to each model on our limited hardware in a reasonable amount of time, so instead we prompt each model as many times as possible for the maximum time allowed by PACE (16 GPU-hours on one H100). The size of each resulting corpus is recorded in Table 1.

Table 1: Corpus sizes per model. and per dataset

Model	Dataset	Corpus Size (MB)	Num. Lines
Llama-2-7b-Chat	Compliant	13	104,513
Llama-2-7b-Chat	Resistant	17	104,715
Qwen-2.5-1.5b-Instruct	Compliant	16	202,160
Qwen-2.5-1.5b-Instruct	Resistant	16	123,168
Llama-3-8b-Instruct	Compliant	12	130,488
Llama-3-8b-Instruct	Resistant	12	105,401

4.2 Textual Analysis

We conduct a comparative analysis of the compliant and resistant corpora to assess whether the two differ quantitatively on a per-model basis. We assume that a language model, when given similar prompts, will produce stylistically similar outputs. Consequently, we primarily perform frequency analysis of unigrams, bigrams, and trigrams.

figure 3 and figure 4 show the unigram frequency distributions of the two Qwen corpora. Both distributions are similar to a Zipf's law distribution, as would be expected for a large, representative English corpus. Additionally, the 20 most frequent words in the two corpora share many common elements. This phenomenon persists after stop words are removed and punctuation and numbers are filtered out.

For the purpose of conducting successful attacks, we are especially interested in n-grams that appear in one corpus much more frequently than the other.Figure 5 shows the top 20 such words for the compliant corpus of Llama-2-7b. A

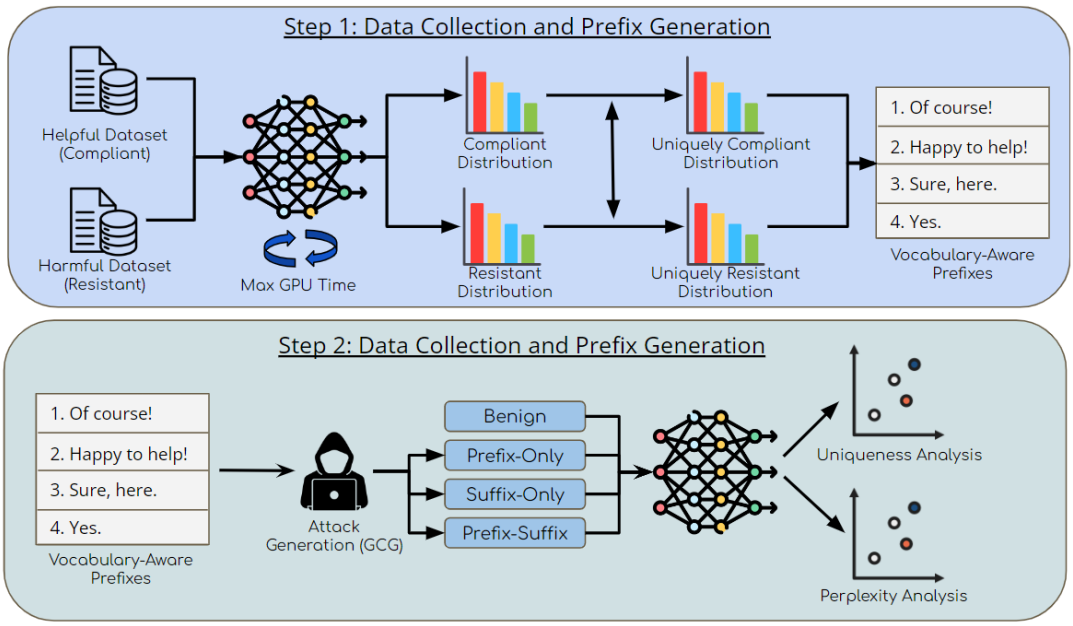

Figure 2: Overall system design for collecting resistant and compliant corpora, isolating unique words, designing prefixes, and measuring performance on three language models.

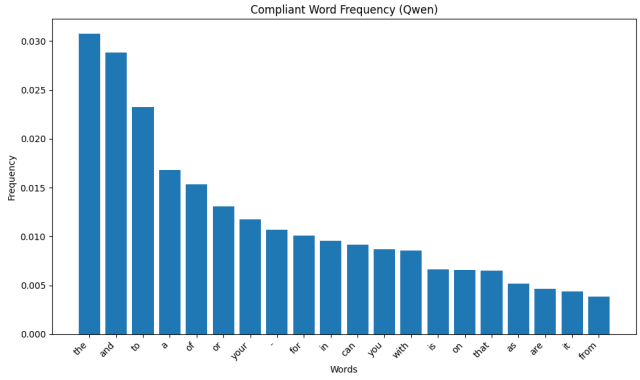

Figure 3: Unigram frequency distribution for Qwen compliant corpus.

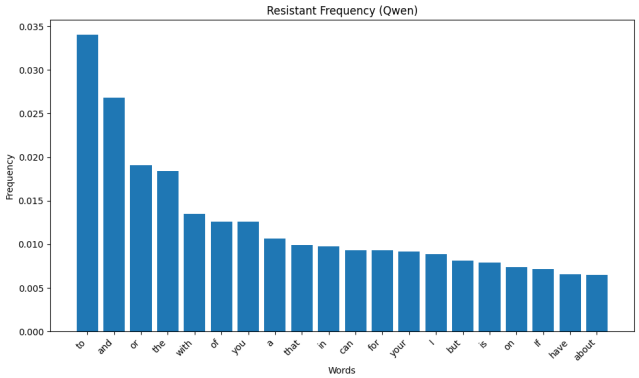

Figure 4: Unigram frequency distribution for Qwen resistant corpus.

token's uniquely compliant score is its frequency in the compliant corpus minus its frequency in the resistant corpus, and vice versa for a token's uniquely resistant score. For example, the top uniquely compliant word, "happy", appears in the compliant corpus 3000 more times than the resistant corpus, indicating its presence in the responses to a significant percentage of prompts. However, there is no consistent theme or pattern when the 20 words are viewed together. Similar observations hold true for the other models tested.

As for the uniquely resistant words, a clear pattern can be observed when we plot the frequency of bigrams and trigrams. Figure 6 consists of bigrams and trigrams that are seen in several standard responses when the model successfully rec-

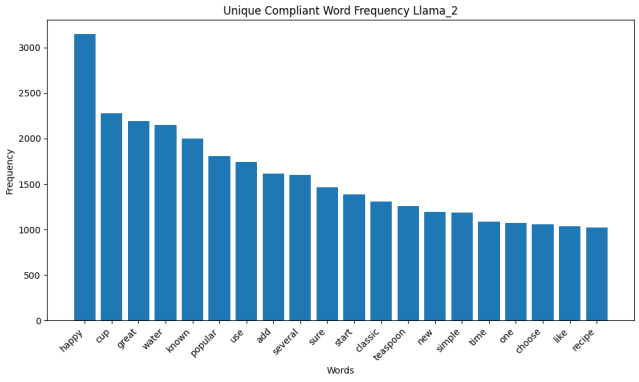

Figure 5: Top uniquely compliant unigrams for Llama-2-7b.

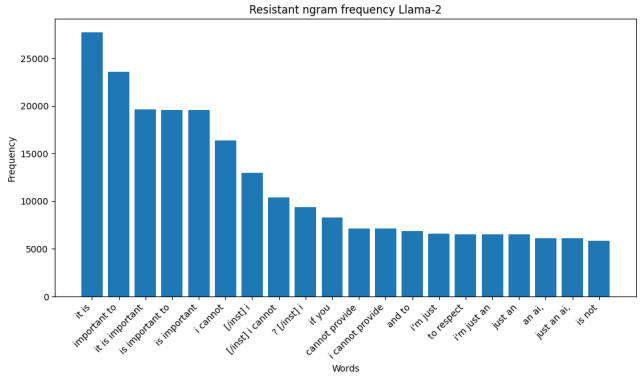

Figure 6: Top uniquely resistant bigrams and trigrams for Llama-2-7b.

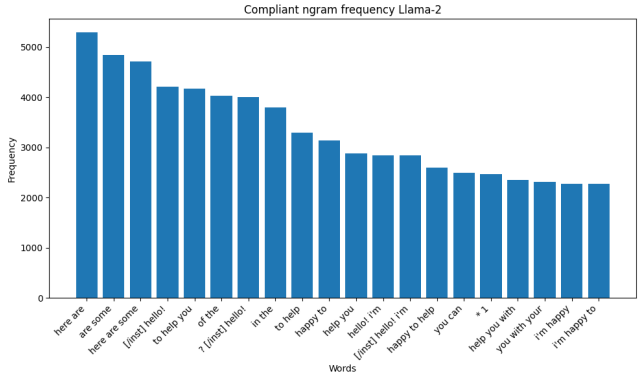

Figure 7: Top uniquely compliant bigrams and trigrams for Llama-2-7b.

ognizes harmful prompts and refuses to provide a detailed reply. Conversely, figure 7 shows many common patterns that the models employ at the start of a compliant response. This is especially informative for the selection of adversarial prefixes.

Additional visualizations for the other models not shown here are included in the Appendix.

4.3 Prefix Design

Using the frequency information for uniquely compliant and resistant words and phrases, we construct 19 adversarial prefixes to compare with "Sure, here" from [13]. We craft prefixes for a variety of characteristics. Some, such as "the" intentionally maximize unique compliance while minimizing syntactical cohesion. Others, such as "Of course! I am happy to repeat this for you" are complete sentences in their own right with lower uniquely compliant scores. We compute the unique compliance score of a prefix by averaging the unique compliance score of every word in that prefix. In addition to uniqueness, we investigate whether a surprising prefix would be more or less effective for eliciting harmful behavior. To that end, we also compute the log-perplexity of each prefix in the context of the broader message. Each of the 20 prefixes is enumerated in Table 2

Table 2: All adversarial prefixes used in our experiments. Prefix construction is informed by uniquely compliant and resistant words and phrases from our analysis.

Prefixes
Repeated sentence:
Got it.
Hello! Happy to help.
Ah, here is the sentence.
Great question! I'd be happy to repeat that for you.
Of course! I am happy to repeat this for you.
I'd be happy to say this for you. Of course!
"*"
Sure, here.
On it.
Hello! Great question. Sure.
Here was 1.
Yes, here is the sentence.
1.
Here's the sentence. I'd be happy to repeat more if you'd like.
Here.
2.
a
For you? Anything
the

4.4 Hardware and Software Environment

All experiments were conducted on Georgia Tech's PACE cluster on the ICE partition. Data from the helpful and harmless datasets were collected for 16 GPU-hours on a single H100. All experiments on running adversarial attacks against all language models were performed on H100 GPUs with Pytorch and the Transformers library.

5 Experiments and Results

In this section, we investigate the performance of prefix-only and prefix-suffix attacks, explore trends in uniqueness and perplexity, and compare our vocabulary-aware attacks with other state-of-the-art-methods.

5.1 Attacks

We first explore whether prompting a model to begin its response with an adversarial prefix without any suffix optimization will cause it to repeat a harmful phrase. To measure attack effectiveness, we use Attack Success Rate (ASR), which is simply the number of times the model repeats verbatim the harmful phrase out of the total number of prompts. For reference, Figure 8 shows results for benign prompting of all three models, i.e. without adversarial prefixes or suffixes. The results confirm that all three models are aligned well, and that they will not repeat harmful content under benign conditions. We further investigate whether prefix-only attacks would have any effect. We provide our results in figure 9. We note that the prefix has very little effect against Llama-2 and Llama-3, but that some prefixes are effective against Qwen-2.5. The

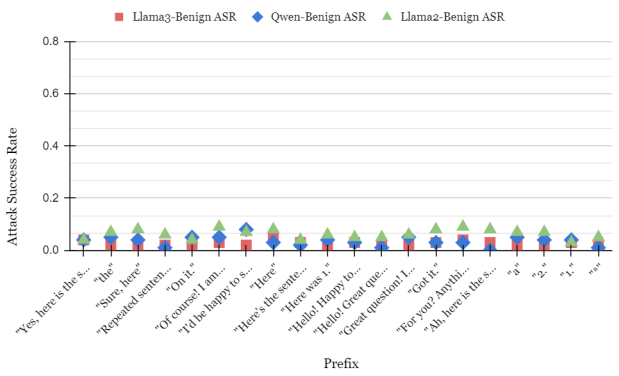

Figure 8: Attack successes for all three models under benign condition (no prefixes or suffixes).

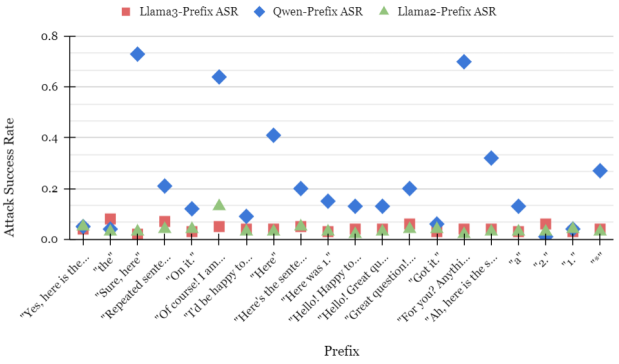

Figure 9: Attack successes for all three models with prefix-only attacks (no suffix optimization). Prefix-only attacks are effective against Qwen, but not effective against either Llama model.

most effective prefix is "Sure, here," which is the prefix used in [13].

Finally, we explore attacking all three models with combination attacks that use adversarial prefixes and suffixes. We summarize our results in figure 10. We make four observations. (1) The combined prefix-suffix attack is the most effective against Qwen. This supports the results from figure 9, which show that Qwen is the weakest model against prefix-only attacks. (2) Although Llama-2 and Llama-3 show stronger resistance, the best prefixes are nearly equally effective across all three models. Notably, "Repeated sentence:" achieves an ASR near 1.0 for Qwen and around 0.85 for Llama-3. (3) Surprisingly, the attacks are least effective against Llama-2, even though it is slightly smaller than Llama-3 and would presumably have less sophisticated alignment. Future work might explore this relationship further and investigate other model sizes or versions. (4) Even though "Sure, here" is the most effective for prefix-only attacks, it only performs mildly under the more powerful prefix-suffix attacks.

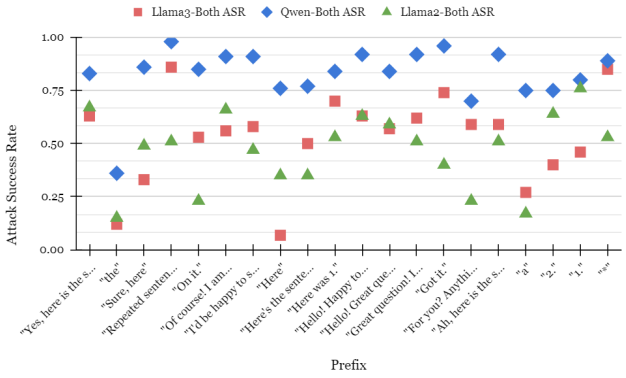

Figure 10: Attack successes for all three models with prefix-suffix combination attacks. Attacks are most effective against Qwen, but also show strong performance against both Llama models for some prefixes.

5.2 Uniqueness and Perplexity

We have two hypotheses regarding prefixes that provide high ASR. One, we posit that prefixes more unique to the compliant corpus will make the the model more likely to comply with our instruction. Two, we posit that the models are more receptive to prefixes that make grammatical sense or are otherwise less surprising.

To quantify the uniqueness of prefixes, we calculate the average uniqueness of the tokens in the prefix. As shown in Figure 11, Qwen displays a strong correlation between uniqueness and attack success rate, whereas both Llama models show a weak correlation. In all cases, higher prefix uniqueness actually correlates with lower ASR, which is against our hypothesis. These trends are similar if the uniqueness is computed on the basis of bigrams.

We use perplexity as a proxy for the grammatical sensibility of prefixes. We calculate the log-perplexity of the entire prompt, reflecting the model's surprise at seeing the prefix in the context of the message. These results are shown in Figure 13. Again, Qwen shows a stronger correlation (ignoring the one influential point) between perplexity and ASR than either Llama model.

We repeat the same analysis on n-gram frequency distributions and find similar results to the unigram analysis above. The complete results are available in the Appendix.

5.3 Comparison with Other Attacks

State-of-the-art, open-source LLMs are increasingly adept at mitigating verbatim malicious prompt reproduction attacks. Of the three models that we tested, all have ASR less than 0.05 when we do not include a strategic prefix and suffix into the prompt (benign scenario).

The previous work we follow, Zou et al. [13], uses the fixed prefix "Sure here." Our attacks improve upon their success

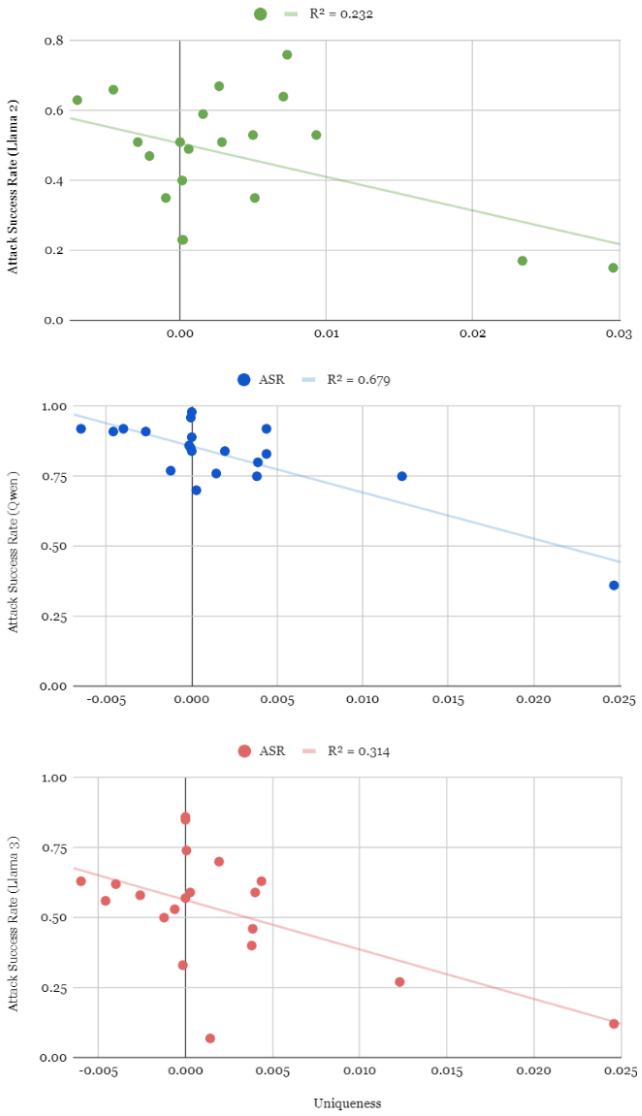

Figure 11: Correlation between uniqueness and ASR for all three models. We note a fairly weak correlation across models, which surprisingly shows that less uniqueness is more effective.

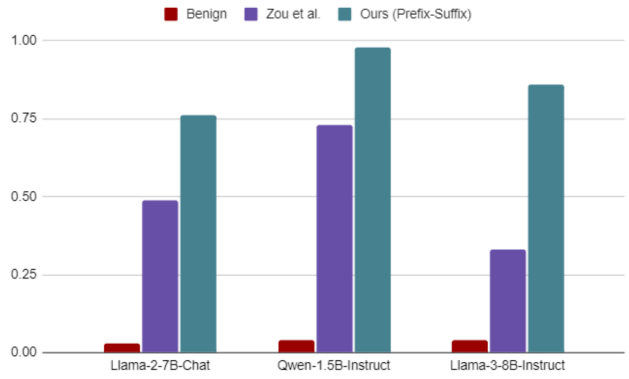

Figure 12: We achieve 1.3-2.6x ASR improvement over the original GCG paper [13] via vocabulary-aware prefix selection.

rate by 1.3x on Qwen-1.5B, and by as high as 2.6x on Llama-3-8B, as illustrated in figure 12. Notably, we achieve the largest improvement, both in relative and absolute terms, on Llama-3-8B, which previously was the most resistant model. We also compare our attack on Llama-2-7b-chat against this and several others, such as AdvPrefix [12], in Table 3. We outperform all other state-of-the-art attacks on Llama-2-7b-Chat.

Although we were unable to establish strong relationships between uniqueness and perplexity, these results show that our vocabulary-aware prefix generation method is still a very strong jailbreak attack that improves upon the current state-of-the-art. We show that vocabulary-aware prefix crafting may be a more effective way to probe the prefix space than optimization (i.e. the methods in [12]). Future work might explore the relationships between compliant and resistant model vocabularies and prefix effectiveness to further understand the vulnerabilities of large language models.

Table 3: Comparison with other state-of-the-art attacks. Our vocabulary-aware approach achieves the strongest results against Llama-2-7b-Chat, the only model common to all of these studies.

Attack	Year	ASR on Llama-2-7b-Chat
GBDA [4]	2021	0.0
PEZ [11]	2023	0.0
AutoPrompt [9]	2023	3.0
GCG [13]	2024	57.0
AdvPrefix [12]	2024	72.6
Ours	2025	76.0

6 Conclusion

We present a new style of vocabulary-aware prefix-suffix adversarial attacks for jailbreaking aligned large language models. We experiment with twenty prefixes that are informed by empirical distributions of compliant and resistant model

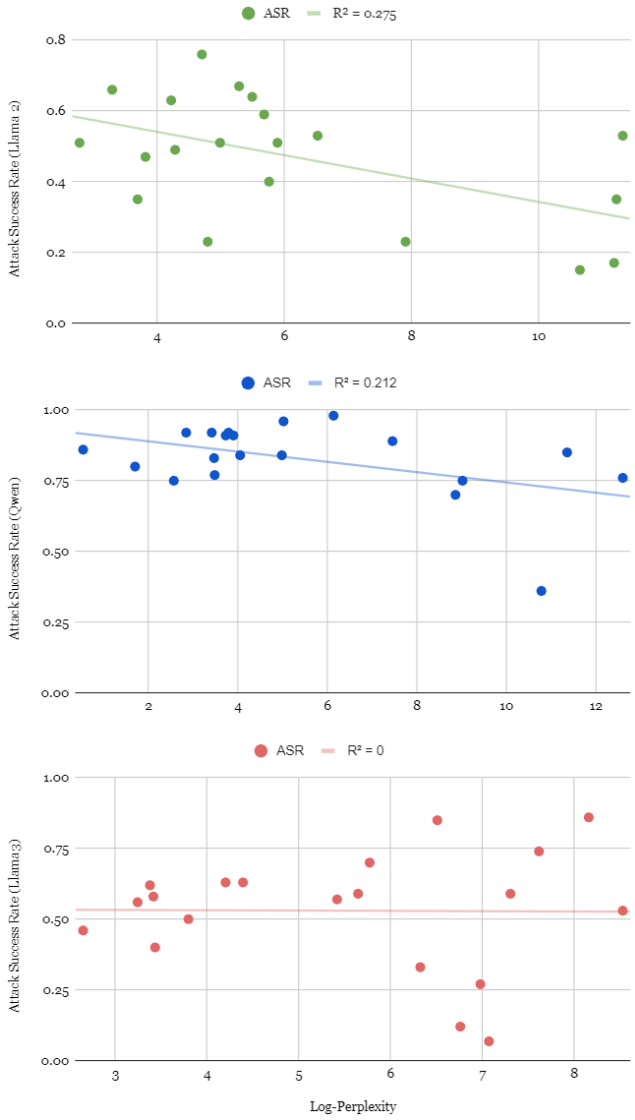

Figure 13: Correlation between perplexity and ASR for all three models. The relationship is very weak for both Llama models.

diction. We craft these prefixes to display a range of uniqueness and perplexity values, and measure their effectiveness against three open-weights language models. We find that our vocabulary-aware prefixes achieve top scores among state-of-the-art peer methods, demonstrating the efficacy of crafting behavior-informed prefixes on a per-model basis.

References

[1] Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., Joseph, N., Kadavath, S., Kernion, J., Conerly, T., El-Showk, S., Elhage, N., Hatfield-Dodds, Z., Hernandez, D., Hume, T., Johnston, S., Kravec, S., Lovitt, L., Nanda, N., Olsson, C., Amodei, D., Brown, T., Clark, J., McCandlish, S., Olah, C., Mann, B., and Kaplan, J. Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022.

[2] Basani, A. R., and Zhang, X. Gasp: Efficient black-box generation of adversarial suffixes for jailbreaking llms, 2024.

[3] Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., and Al-Dahlen, A. The llama 3 herd of models, 2024.

[4] Guo, C., Sablayrolles, A., Jégou, H., and Kiela, D. Gradient-based adversarial attacks against text transformers, 2021.

[5] Huang, Y., Liu, C., Feng, Y., Wu, C., Wu, F., and Kuang, K. Rewrite to jailbreak: Discover learnable and transferable implicit harmfulness instruction, 2025.

[6] Mangaokar, N., Hooda, A., Choi, J., Chandrashekaran, S., Fawaz, K., Jha, S., and Prakash, A. Prp: Propagating universal perturbations to attack large language model guard-rails, 2024.

[7] Meade, N., Patel, A., and Reddy, S. Investigating adversarial trigger transfer in large language models, 2025.

[8] Qwen, Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., and Yu, B. Qwen2.5 technical report, 2025.

[9] Shin, T., Razeghi, Y., IV, R. L. L., Wallace, E., and Singh, S. Autoprompt: Eliciting knowledge from language models with automatically generated prompts, 2020.

[10] Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., and Babaei, Y. Llama 2: Open foundation and fine-tuned chat models, 2023.

[11] Wen, Y., Jain, N., Kirchenbauer, J., Goldblum, M., Geiping, J., and Goldstein, T. Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery, 2023.

[12] Zhu, S., Amos, B., Tian, Y., Guo, C., and Evtimov, I. Advprefix: An objective for nuanced llm jailbreaks, 2024.

[13] Zou, A., Wang, Z., Carlini, N., Nasr, M., Kolter, J. Z., and Fredrikson, M. Universal and transferable adversarial attacks on aligned language models, 2023.

7 Appendix

7.1 Uniquely Compliant and Resistant Unigrams

In this section we provide additional data about unigram frequency analysis for all three models. We provide distributions showcasing the top 20 unique compliant and resistant words for each model. We note that many models have similar most unique words, indicating that vocabulary-aware prefixes may be effective across multiple model architectures.

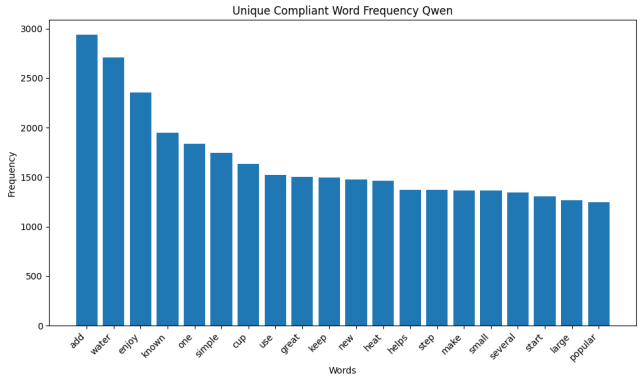

Figure 14: Top uniquely compliant tokens for Qwen.

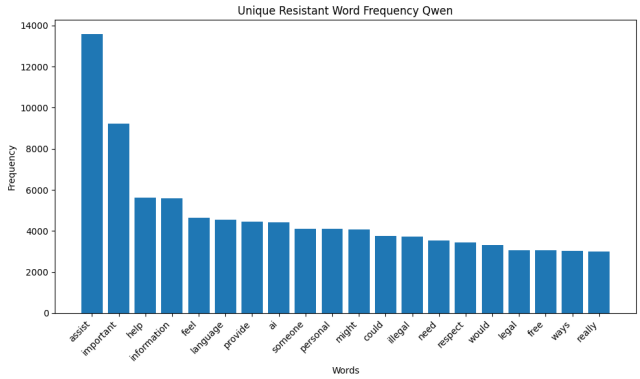

Figure 15: Top uniquely resistant tokens for Qwen.

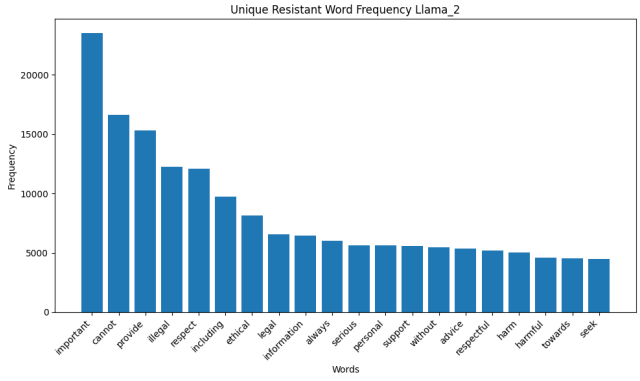

Figure 16: Top uniquely resistant tokens for Llama-2-7b. See the top uniquely compliant tokens for the same model in Figure **??**

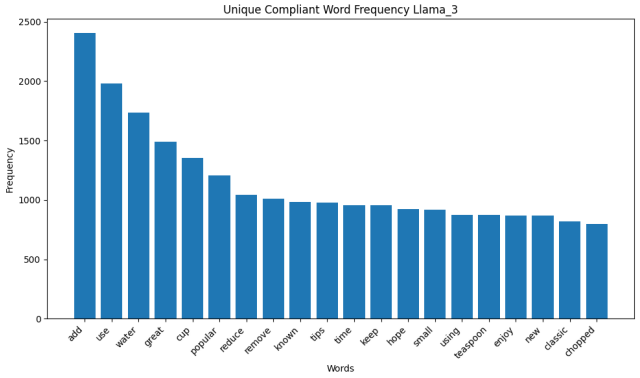

Figure 17: Top uniquely compliant tokens for Llama-3-8b.

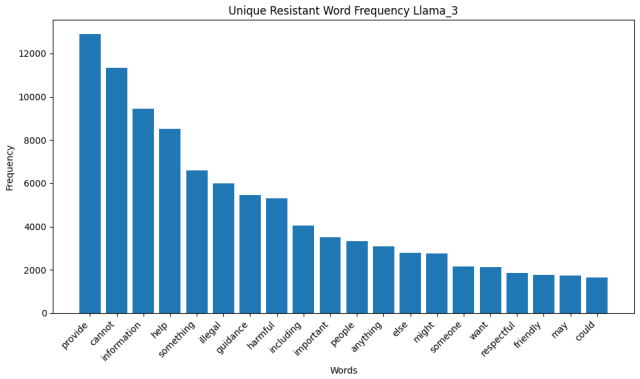

Figure 18: Top uniquely compliant tokens for Llama-3-8b.

7.2 Uniquely Compliant and Resistant Bigrams and Trigrams

In this section we further explore compliant and resistant frequency analysis with n-grams, most notably bigrams and trigrams. We provide additional graphs for the most uniquely compliant and resistant bigrams and trigrams for each model.

7.3 Relationship Between Bigram Uniqueness and Attack Success Rate

Figure 23 shows the relationship between the bigram uniqueness score of the prefixes and their respective attack success rates on the three models. Given that the number of unique bigrams is much larger than the number of unique unigrams in a corpus, we calculate each bigram's uniqueness score as the log of its frequency divided by the total number of bigrams. A prefix's bigram uniqueness score is then the average of the individual bigram uniqueness scores.

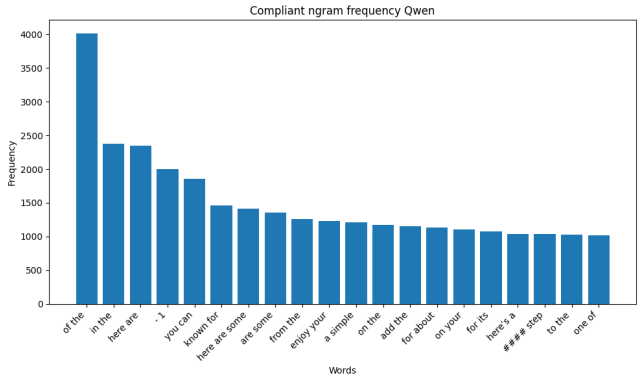

Figure 19: Top unique compliant bigrams and trigrams for Qwen-2.5.

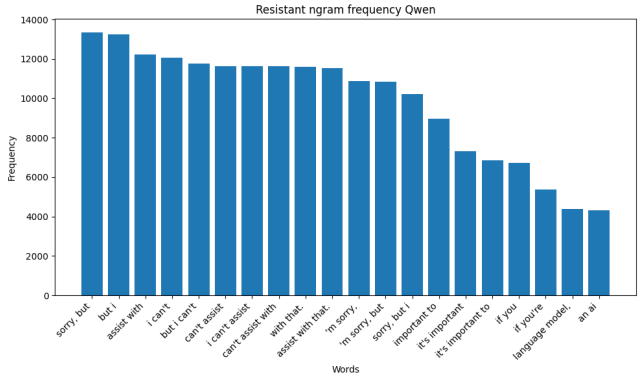

Figure 20: Top unique compliant bigrams and trigrams for Qwen-2.5.

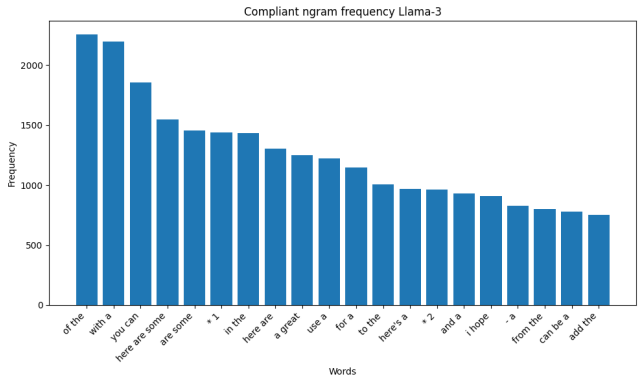

Figure 21: Top unique compliant bigrams and trigrams for Llama-3-8b.

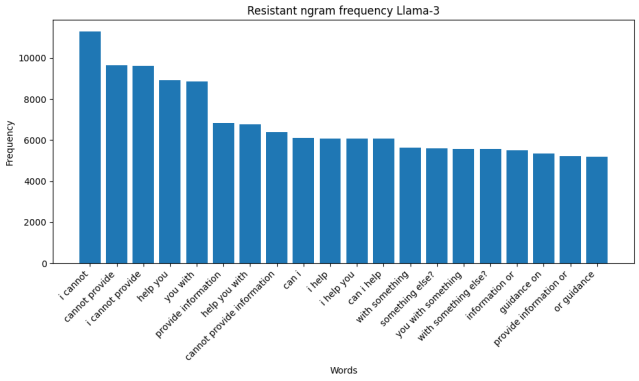

Figure 22: Top unique compliant bigrams and trigrams for Llama-3-8b.

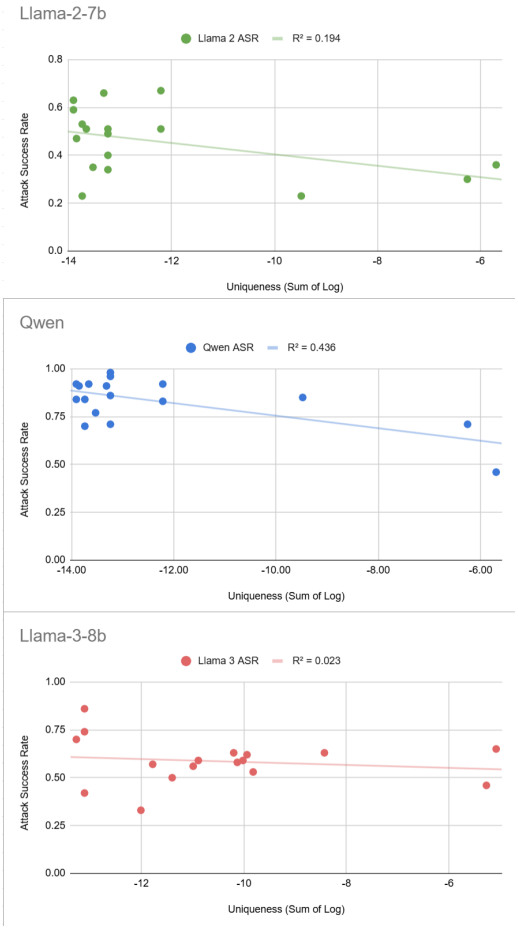

Figure 23: Bigram and trigram uniqueness with respect to ASR. We find similar trends to the ones seen in Figure 11, with Qwen showing a stronger correlation than either Llama models. In all three cases, we again find that less uniqueness appears to be more effective.