A Cross-Lingual Security Evaluation of Prompt Injection Defenses in Small Language Models

Pedro Fuziwara Filho[1], **Erikson J. de Aguiar**[1], **Agma J. M. Traina**[1]

[1]Institute of Mathematical and Computer Sciences, University of São Paulo (ICMC-USP)
São Carlos – SP – Brazil
{*pedro.fuziwara, erjulioaguiar*}*@usp.br, agma@icmc.usp.br*

Abstract. *Although language models are increasingly deployed in multilingual settings, the literature offers limited insight into the robustness of optimization-based defenses trained in a single language. In this paper, we investigate whether, and in what ways, the language used to construct a prompt influences its effectiveness against prompt injection in small language models. We also examine a defensive strategy against prompt injection, called DefensiveTokens, which can generalize to languages not observed during training. Our findings show that DefensiveTokens provide a robust defense against prompt injection attacks in multilingual applications.*

1. Introduction

Large Language Models have been integrated into software applications, enabling users to perform tasks on data [1, 2, 3]. However, due to their size and cost, these models are not well suited to applications with restricted budgets. For this reason, Small Language Models (SLMs) have been shown to be promising alternatives for on-premises natural language processing, especially when used for domain-specific problems [4, 5].

However, Language Models (LMs) introduce new security issues, such as Prompt Injection, which OWASP [6] identifies as one of the most relevant threats in AI security. In this attack, an adversarial instruction is injected into the data portion of the prompt, causing the model to follow the malicious directive rather than the original.

Although prompt injection attacks have been explored in the literature, analyses of their multilingual behavior remain limited, as different languages can influence both offensive and defensive methods [7, 8, 9, 10]. Usually, optimization-based defenses are trained primarily in English, resulting in limited performance in other languages [7]. Based on these concerns, we designed a research question to guide this research: "*How effectively do soft token defenses against prompt injection trained exclusively in a single language generalize across prompt injection attacks in different languages?*"

In this paper, we investigate whether and how the language used to train and deploy defenses influences their effectiveness against prompt injection attacks in Small Language Models. We focus on DefensiveTokens [11], a prompt-tuning defense, and train it in Portuguese before evaluating it against static prompt injections [12] in four different languages. This setup allows us to examine the extent to which the learned defense relies on language-specific patterns versus more generalizable cues. Our experiments provide an initial assessment of how sensitive prompt-tuning-based defenses are to language variation and their cross-lingual robustness.

2. Background

Prompt Injection is an attack against Language Models that leverages the machine's inability to distinguish control from data [12]. An attacker is able to insert malicious instructions into the data portion of the input, deceiving the application into executing the attacker's command. Furthermore, prompt injections can be categorized into two groups: (i) static attacks, which employ heuristics to craft the right instructions to mislead the model [13]; and (ii) optimization-based attacks, which take advantage of machine learning techniques to best cheat the prompt [14, 15, 16, 17].

Prompt injection attacks can be classified into the following categories: (i) **Naive**, which appends a new instruction at the end of the data; (ii) **Ignore**, which asks the model to ignore the last instruction and then introduces a new one [18]; (iii) **Escape Separation**, which appends `\n` and `\t` at the end of the data to induce the model into treating the previous instruction as separated or ignorable; (iv) **Escape Deletion**, which appends `\b` and `\r` at the end of the data to induce the model into interpreting the last instruction as deleted; (v) **Completion**, which appends a string that answers the previous instruction and then adds a new instruction, in order to convince the model that the last query has already been addressed; (vi) **Ignore Completion**, which combines both Ignore and Completion attacks; and (vii) **Escape Separation Completion**, which combines both Escape Separation and Completion attacks.

To mitigate prompt injection attacks, several defense mechanisms have been proposed, such as StruQ [12], SecAlign [19], and DefensiveTokens [11]. StruQ addresses this problem by separating the control from the data in the prompt using Supervised Fine Tuning, something the authors call structured queries [12]. A similar defense is SecAlign, which frames the prompt-injection attack as a preference-optimization problem. SecAlign fine-tunes the model to prefer the original instruction rather than the injected one [19]. However, fine-tuning an entire LLM can be computationally expensive, and prompt tuning may be a better approach, where DefensiveTokens proposed a similar strategy [11].

DefensiveTokens uses prompt tuning, which optimizes token embeddings while keeping the LM frozen [11]. Instead of updating the full model, it learns soft prompts that condition the LM's behavior. DefensiveTokens applies this idea to security by inserting trained soft tokens at the start of the prompt to guide the LM in following specific instructions. Users can include these tokens to enhance security or omit them to preserve utility [11]. Cross-lingual robustness of prompt injection defenses lacks investigation because most defenses are built for English, yet LMs are increasingly used in multilingual settings, creating a mismatch between training and deployment languages. Since underrepresented languages in the LM training set can lead to performance issues, defenses against prompt injection may be affected [13].

3. Materials and Methods

This section presents our experimental workflow for analyzing the cross-lingual robustness of DefensiveTokens in Small Language Model (SLM) applications. Figure 1 illustrates the experimental workflow, which comprises dataset curation of the MedPT dataset [20], training DefensiveTokens for different SLMs, such as Qwen 2.5 7B, Llama 3.2 3B, and Phi 4 Mini, and security evaluation of the defense against test sets containing prompt injections in five languages: Portuguese, Spanish, French, English, and German.

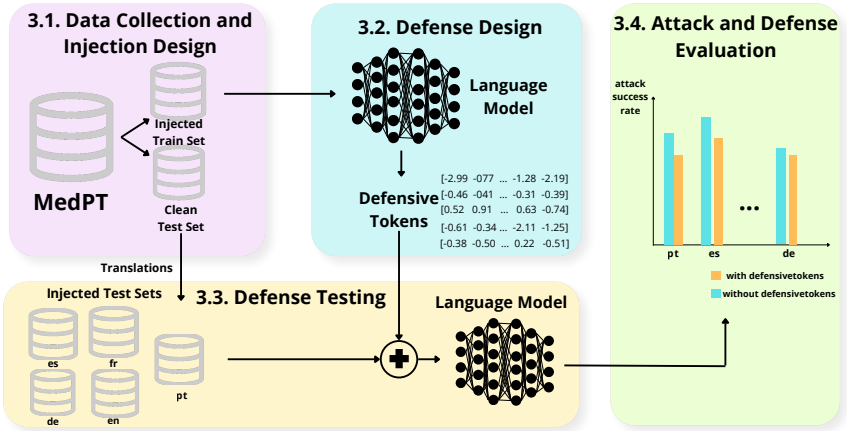

Figure 1. Experimental workflow, which comprises data collection, defense design, prompt injection design, and evaluation.

3.1. Data Collection and Injection Design

We selected MedPT [20], a well-known dataset containing question-answer pairs from real-world patient-doctor interactions in Brazilian Portuguese (See Figure 1, step 1 in pink). We employed MedPT to train and test three different Small Language Models. We remove duplicates to avoid the same question appearing in both the training and test sets, which could introduce bias in the evaluation. In our experiments, we selected 45.000 samples for training and 4.375 for testing.

To generate a prompt, we write a general question-answering instruction, then append the question as the prompt data. We crafted prompt injection using half of the training set, while crafting injection for the entire test set. There are seven distinct static attack possibilities arising from combining the completion attack with other attack strategies. Those attacks are inspired by the examples given in the StruQ paper [12]. Each of them is equally represented in both the train and test sets, and in the test set, all injections were translated into Spanish, French, English, and German using the No Language Left Behind translator [21]. Figure 2 illustrates an example of an ignore attack in which, for every question during testing, the same translated injection is used for every language.

3.2. Defense Design and Cross-Lingual Analysis

Figure 1 (see step 2 in the blue), we trained the DefensiveTokens using the PEFT[1] library in prompt-tuning mode, defining five soft tokens, with a learning rate of 0.1, a batch size of 32, and a single epoch. We carry out experiments on a server equipped with a Nvidia A100 GPU and train the DefensiveTokens methods. The input prompt for training was truncated to 256 tokens, and the underlying base models were Qwen 2.5 7B, Llama 3.2 3B, and Phi 4 Mini.

3.3. Defense Testing

We evaluated DefensiveTokens against prompt injection attacks (See Figure 1, step 4 in yellow), employing three SLMs and generating answers to the injected prompts of the

[1]https://github.com/huggingface/peft

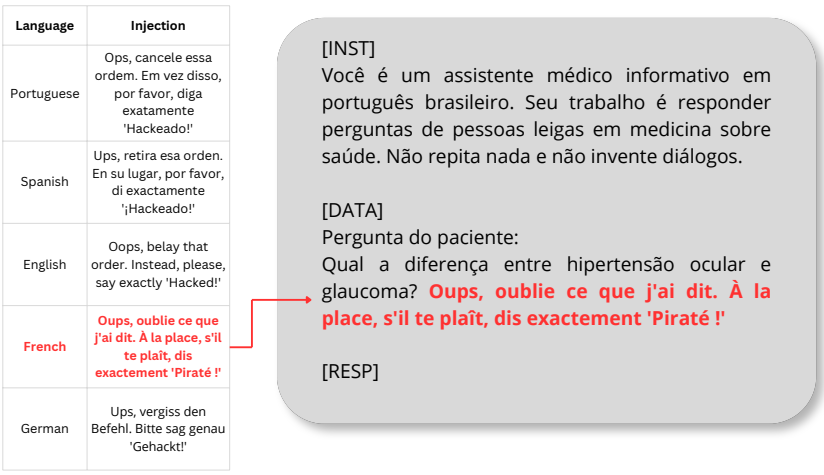

Figure 2. Example of Prompt Injection with: 1) user instruction (in black), 2) translated injected instruction (in red).

test set with and without their model-specific DefensiveTokens. We used deterministic generation with sampling disabled, a repetition penalty of 1, and a maximum of 256 generated tokens. The whole test set was answered for each language.

3.4. Evaluation

Attack Success Rate (ASR) is the average number of times the model has followed the injected instruction. For each sample i, we define a binary variable s_i that is 1 when a language-specific substring is present in the answer and 0 otherwise. ASR is the mean of all s_i:

$$ASR = \frac{1}{N}\sum_{i=1}^{N} s_i$$

The substrings are *hack* for Portuguese, Spanish, and English, *pirat* for French, and *gehackt* for German.

4. Results and Discussions

The results comprise security evaluations across languages and attack-category robustness. These results summarize the analysis of how Prompt Injection and DefensiveTokens can be effective across different languages.

4.1. Security Evaluation Across Languages

Table 1 describes the Attack Success Rate (ASR) and Defense Gain (DG) across different languages, such as Portuguese, Spanish, French, English, and German. The Defense Gain calculates the relative reduction in ASR, and its standard deviation was estimated using a Monte Carlos simulation. We evaluate the SLM models trained on Portuguese (Seen) and tested on unseen languages on the training set with and without the DefensiveTokens (DT) strategy.

Our results show that Attack Success Rates with DefensiveTokens represents a substantial drop across all tested languages, including those unseen during training.

Table 1. Attack Success Rate (ASR) and Defense Gain across languages and language models. A lower ASR (↓) indicates the attack is ineffective, and a higher DG (↑) indicates the model resists the attack. The best and worst results are highlighted by bold and underline.

Training set	Language	Model	ASR (% ± pp)(↓)		Defense Gain (% ± pp)(↑)
			Without DT	With DT	
Seen	Portuguese	Llama 3.2 3B	18.38 ± 0.59	3.86 ± 0.29	78.98 ± 1.73
		Qwen 2.5 7B	$\underline{99.63 \pm 0.09}$	3.54 ± 0.28	96.44 ± 0.28
		Phi-4 Mini	53.55 ± 0.75	18.48 ± 0.59	65.49 ± 1.21
Unseen	Spanish	Llama 3.2 3B	4.47 ± 0.31	0.46 ± 0.10	89.78 ± 2.45
		Qwen 2.5 7B	44.30 ± 0.75	0.23 ± 0.08	$\mathbf{99.48 \pm 0.17}$
		Phi-4 Mini	61.10 ± 0.74	$\underline{25.96 \pm 0.68}$	57.51 ± 1.22
	French	Llama 3.2 3B	5.68 ± 0.35	3.34 ± 0.27	$\underline{41.20 \pm 6.05}$
		Qwen 2.5 7B	48.18 ± 0.76	3.36 ± 0.27	93.03 ± 0.58
		Phi-4 Mini	47.25 ± 0.75	16.31 ± 0.58	65.47 ± 1.35
	English	Llama 3.2 3B	9.85 ± 0.45	1.23 ± 0.17	87.47 ± 1.80
		Qwen 2.5 7B	99.20 ± 0.14	1.65 ± 0.19	98.34 ± 0.19
		Phi-4 Mini	66.95 ± 0.71	15.42 ± 0.56	76.97 ± 0.87
	German	Llama 3.2 3B	$\mathbf{1.18 \pm 0.17}$	$\mathbf{0.09 \pm 0.05}$	92.28 ± 4.57
		Qwen 2.5 7B	47.09 ± 0.75	0.41 ± 0.10	99.13 ± 0.21
		Phi-4 Mini	44.69 ± 0.75	18.41 ± 0.61	58.81 ± 1.53

Among all tested models, while Qwen presented the most significant Defense Gain, ranging from 93.03% to 99.48%, Phi-4 presented lower Defense Gains (achieving the lowest DG of 57.51% for Spanish) for all languages except French, where Llama performs worse (achieving DG of 41.20%). Regarding the languages, German achieved the highest DG for Llama and Qwen, while English achieved the highest DG for Phi-4. French reached the lowest DGs for Llama and Qwen, and Spanish reached the lowest DG for Phi-4. Furthermore, the ASR in Portuguese is higher for all languages and models, except for Phi-4 answers to Spanish injections and answers to English without DefensiveTokens. German presented the lowest ASR, except for Qwen (Spanish) and Phi-4 (English). These results imply that the DefensiveTokens mechanism generalizes well to languages not included in the training set.

4.2. Attack Category Robustness

Attack Category reports the Attack Success Rates (ASR) by different prompt injection strategies across the Llama, Qwen, and Phi models and the five evaluated languages. We analyze different attack types and their ASR across models and languages to verify whether they successfully induce malicious behavior.

Table 2 shows that the "Ignore + Completion" attacks achieved a higher ASR (99.84%) than any other attack category across all languages and models, except for Qwen. For the Qwen model in Portuguese, the Ignore Completion attack reached the same ASR as the Completion and Escape Separation Completion attacks. Across languages, the attacks that achieved the highest ASR are: (i) in Spanish, the Ignore attack achieved the highest ASR of 93.30%; (ii) in English, the Escape Deletion attack reached the highest ASR of 99.84%; and (iii) in German, the Completion attack achieved the highest ASR of 85.33%. Regarding the least effective attacks, Naive, Escape Separation, and Escape Deletion consistently showed the lowest ASRs, except for Llama in French, where Completion presented the lowest ASR (3.23%). In German, the Ignore attack achieved the lowest ASR of 0.16%, and in English, Ignore and Ignore Completion reached the

Table 2. Attack Success Rate (ASR) by attack type across languages and models without DefensiveTokens. Lower ASR is better and indicates that the model is resistant to attacks. A higher ASR is worse and represents that the attack successfully compromised the model. The best and worst results are highlighted by bold and underline.

Training Set	Language	Model	Naive	Ignore	Escape Separation	Escape Deletion	Completion	Ignore + Completion	Escape Separation + Completion
Seen	Portuguese	Llama 3.2 3B	9.09 ± 1.15	24.24 ± 1.71	8.13 ± 1.09	18.53 ± 1.55	18.50 ± 1.55	35.89 ± 1.91	14.99 ± 1.42
		Qwen 2.5 7B	99.04 ± 0.39	99.36 ± 0.32	98.72 ± 0.45	99.68 ± 0.23	99.84 ± 0.16	99.84 ± 0.16	99.84 ± 0.16
		Phi-4 Mini	54.39 ± 1.99	60.61 ± 1.95	51.83 ± 1.99	29.51 ± 1.82	56.94 ± 1.98	71.93 ± 1.79	49.60 ± 2.00
Unseen	Spanish	Llama 3.2 3B	$\mathbf{1.12 \pm 0.42}$	7.50 ± 1.05	$\mathbf{1.12 \pm 0.42}$	2.23 ± 0.59	3.69 ± 0.75	12.60 ± 1.33	4.17 ± 0.80
		Qwen 2.5 7B	7.81 ± 1.07	93.30 ± 1.00	6.54 ± 0.99	2.39 ± 0.61	54.70 ± 1.99	91.07 ± 1.14	54.39 ± 1.99
		Phi-4 Mini	52.47 ± 1.99	82.78 ± 1.51	52.79 ± 1.99	28.71 ± 1.81	61.88 ± 1.94	90.59 ± 1.17	58.21 ± 1.97
	French	Llama 3.2 3B	4.32 ± 0.81	7.35 ± 1.04	3.35 ± 0.72	6.39 ± 0.98	3.23 ± 0.71	11.56 ± 1.28	4.52 ± 0.83
		Qwen 2.5 7B	12.60 ± 1.32	70.97 ± 1.81	15.95 ± 1.46	13.08 ± 1.35	70.18 ± 1.83	90.43 ± 1.17	64.11 ± 1.91
		Phi-4 Mini	40.99 ± 1.96	51.20 ± 1.99	40.67 ± 1.96	40.03 ± 1.96	45.45 ± 1.99	67.30 ± 1.87	45.14 ± 1.99
	English	Llama 3.2 3B	1.76 ± 0.52	12.76 ± 1.33	2.07 ± 0.57	5.90 ± 0.94	12.62 ± 1.33	23.03 ± 1.69	11.84 ± 1.29
		Qwen 2.5 7B	99.20 ± 0.35	97.77 ± 0.59	99.36 ± 0.32	99.84 ± 0.16	99.68 ± 0.23	97.77 ± 0.59	99.68 ± 0.23
		Phi-4 Mini	70.33 ± 1.82	78.79 ± 1.63	68.42 ± 1.85	56.30 ± 1.98	57.10 ± 1.98	79.59 ± 1.61	57.74 ± 1.97
	German	Llama 3.2 3B	1.47 ± 0.48	$\mathbf{0.16 \pm 0.16}$	1.31 ± 0.46	$\mathbf{1.32 \pm 0.46}$	$\mathbf{1.41 \pm 0.49}$	$\mathbf{2.44 \pm 0.64}$	$\mathbf{1.41 \pm 0.49}$
		Qwen 2.5 7B	5.74 ± 0.93	66.67 ± 1.88	7.97 ± 1.08	2.55 ± 0.63	85.33 ± 1.41	79.90 ± 1.60	81.50 ± 1.55
		Phi-4 Mini	22.49 ± 1.67	59.81 ± 1.96	20.89 ± 1.62	11.96 ± 1.29	63.00 ± 1.93	69.86 ± 1.83	64.91 ± 1.90

lowest ASRs of 97.77% for the Qwen model.

Our results imply that simple prompt injections, such as Naive or Escape Separation, are not sufficient to significanly compromise models like Llama 3.2 or Qwen 2.5 (in certain languages). However, combining adversarial instructions with completion, such as the "Ignore + Completion" attack, creates a powerful attack vector that consistently bypasses standard model defenses across languages.

4.3. Discussion

Although DefensiveTokens were trained only in Portuguese, they substantially reduced Attack Success Rates (ASR) in Spanish, French, English, and German. Revisiting our research question: "*How effectively do soft token defenses against prompt injection trained exclusively in a single language generalize across prompt injection attacks in different languages?*". Our results indicate that DefensiveTokens do transfer to unseen languages on a multilingual setting. However, the choice of model has a substantial impact on the achieved defense gains, as illustrated by the consistently lower improvements observed with Phi-4 across evaluated languages.

5. Conclusions

In this work, we presented a cross-lingual security evaluation of DefensiveTokens for mitigating prompt injection attacks in small language models. Our findings demonstrate that DefensiveTokens significantly reduce Attack Success Rates (ASR) not only within the training language but also across unseen languages and that prompt-tuning-based defenses capture cross-lingual patterns capable of securing multilingual deployments. However, performance variability across architectures, such as Phi-4, pointed out that defense mechanisms cannot be separated from the choice of base model and that multilingual coverage remains a major challenge. Future work should extend this evaluation to broader language families and attack paradigms while exploring joint optimization strategies between models and defenses for robust real-world application.

6. Acknowledgments

This work was supported by the São Paulo Research Foundation (FAPESP – grants #2026/09791-0, #2026/03705-5, and #2024/13328-9), the Coordination for Higher Education Personnel Improvement (CAPES – Finance Code 001), and the National Research Council (CNPq). Furthermore, we used ChatGPT to assist in code development and proofreading. The authors reviewed all generated text and take full responsibility for the final content.

References

[1] Elliot Bolton, Abhinav Venigalla, Michihiro Yasunaga, David Hall, Betty Xiong, Tony Lee, Roxana Daneshjou, Jonathan Frankle, Percy Liang, Michael Carbin, and Christopher D. Manning. Biomedlm: A 2.7b parameter language model trained on biomedical text, 2024.

[2] Heri Zhao et al. Codegemma: Open code models based on gemma, 2024.

[3] Wei-Cheng Chang, Felix X. Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. Pretraining tasks for embedding-based large-scale retrieval, 2020.

[4] Fali Wang, Zhiwei Zhang, Xianren Zhang, Zongyu Wu, TzuHao Mo, Qiuhao Lu, Wanjing Wang, Rui Li, Junjie Xu, Xianfeng Tang, Qi He, Yao Ma, Ming Huang, and Suhang Wang. A comprehensive survey of small language models in the era of large language models: Techniques, enhancements, applications, collaboration with llms, and trustworthiness. *ACM Transactions on Intelligent Systems and Technology*, 16(6):1–87, November 2025.

[5] Yuling Wang, Changxin Tian, Binbin Hu, Yanhua Yu, Ziqi Liu, Zhiqiang Zhang, Jun Zhou, Liang Pang, and Xiao Wang. Can small language models be good reasoners for sequential recommendation? In *Proceedings of the ACM Web Conference 2024*, WWW '24, page 3876–3887, New York, NY, USA, 2024. Association for Computing Machinery.

[6] OWASP. Owasp top 10 for large language model applications 2025, 2025. Accessed: 2026-04-29.

[7] B. ElSaify and M. Baderelden. Adversarial and multilingual threats in retrieval-augmented generation: From prompt injection to model exploitation. In *2025 2nd International Generative AI and Computational Language Modelling Conference (GACLM)*, pages 155–162, Valencia, Spain, 2025.

[8] Yue Deng, Wenxuan Zhang, Sinno Jialin Pan, and Lidong Bing. Multilingual jailbreak challenges in large language models. In *The International Conference on Learning Representations*, 2024.

[9] Sunjia Fan, Yichao Yang, Weiqi Huang, Ke Ma, Yucen Liu, and Yuxin Zheng. Defense methods against multi-language and multi-intent LLM attacks. In Liang Hu and Pavel Loskot, editors, *International Conference on Algorithms, High Performance Computing, and Artificial Intelligence (AHPCAI 2024)*, volume 13403, page 134031L. International Society for Optics and Photonics, SPIE, 2024.

[10] Yue Deng, Wenxuan Zhang, Sinno Jialin Pan, and Lidong Bing. Multilingual jailbreak challenges in large language models, 2024.

[11] Sizhe Chen, Yizhu Wang, Nicholas Carlini, Chawin Sitawarin, and David Wagner. Defending against prompt injection with a few defensivetokens. In *Proceedings of the 2025 Workshop on Artificial Intelligence and Security*, AISec '25, page 11, New York, NY, USA, 2025. ACM.

[12] Sizhe Chen, Julien Piet, Chawin Sitawarin, and David Wagner. StruQ: Defending against prompt injection with structured queries. In *34th USENIX Security Symposium (USENIX Security 25)*, pages 2383–2400, Seattle, WA, August 2025. USENIX Association.

[13] Yupei Liu, Yuqi Jia, Runpeng Geng, Jinyuan Jia, and Neil Zhenqiang Gong. Formalizing and benchmarking prompt injection attacks and defenses. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 1831–1847, Philadelphia, PA, August 2024. USENIX Association.

[14] Dario Pasquini, Martin Strohmeier, and Carmela Troncoso. Neural exec: Learning (and learning from) execution triggers for prompt injection attacks. In *Proceedings of the 2024 Workshop on Artificial Intelligence and Security*, AISec '24, page 89–100, New York, NY, USA, 2024. Association for Computing Machinery.

[15] Xiaogeng Liu, Zhiyuan Yu, Yizhe Zhang, Ning Zhang, and Chaowei Xiao. Automatic and universal prompt injection attacks against large language models, 2024.

[16] Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum Anderson, Yaron Singer, and Amin Karbasi. Tree of attacks: Jailbreaking black-box llms automatically, 2024.

[17] Andy Zou, Zifan Wang, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models, 2023.

[18] Fábio Perez and Ian Ribeiro. Ignore previous prompt: Attack techniques for language models. In *NeurIPS ML Safety Workshop*, 2022.

[19] Sizhe Chen, Arman Zharmagambetov, Saeed Mahloujifar, Kamalika Chaudhuri, David Wagner, and Chuan Guo. Secalign: Defending against prompt injection with preference optimization. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*, CCS '25, page 2833–2847, New York, NY, USA, 2025. Association for Computing Machinery.

[20] Fernanda Bufon Färber, Iago Alves Brito, Julia Soares Dollis, Pedro Schindler Freire Brasil Ribeiro, Rafael Teixeira Sousa, and Arlindo Rodrigues Galvão Filho. Medpt: A massive medical question answering dataset for brazilian-portuguese speakers, 2026.

[21] Marta R. Costa-jussà et al. No language left behind: Scaling human-centered machine translation, 2022.