Proceedings of the 2nd International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/114/2024.18155

Compare the Performance of Current Large Language Models in Terms of System Security

Xuanye Liu

College of Artificial Intelligence, Yango University, Fuzhou, Fujian, 350015, China

1617106544@qq.com

Abstract. This paper provides a comparative analysis of the systematic security capabilities of four leading large language models: OpenAI's ChatGPT, Meta AI's LLaMA, Moonshot AI's BERT Chat, and Baidu's ERNIE series, including "Wenxin Yiyan." Each model's performance in addressing potential systematic security threats and vulnerabilities was evaluated through a rigorous assessment process. The study found significant differences in the security performance of these models. Specifically, ChatGPT exhibits remarkable resilience in handling confidential information, while LLaMA's advanced contextual understanding enhances its ability to identify and mitigate emerging security risks. BERT Chat stands out due to its strong user privacy protections, and Baidu's ERNIE, particularly "Wenxin Yiyan," provides comprehensive data security through multiple layers of defense. The study highlighted the diversity of security strategies employed by models, while pointing to the need for continued innovation in security measures as large language models develop.

Keywords: Large Language Models, System Security, Comparative Analysis, Security Features, Innovation in Security Measures.

1. Introduction

A capable LLM should have four key features: Firstly, it should demonstrate a profound comprehension of the contextual nuances inherent in natural language, enabling it to grasp the subtleties and implications of language usage. Secondly, such a model must possess the ability to generate text that bears a striking resemblance to human writing, not just in terms of grammar and syntax, but also in terms of style and tone. Thirdly, it should exhibit a keen awareness of the surrounding context, especially in domains that are rich in knowledge and require a deep understanding of the subject matter. Lastly, a capable LLM should have a robust ability to adhere strictly to instructions, which proves beneficial in effectively resolving issues and making informed decisions[1]. Large language models, including OpenAI's ChatGPT, Meta AI's LLaMA, Google's BERT, and Baidu's ERNIE3.0, featuring "Wenxin Yiyan", have emerged as trans-formative technologies in natural language processing [2-5]. These models demonstrate exceptional proficiency in comprehending and generating human-like text, finding applications in various fields (Figure 1). However, as their popularity and integration into diverse systems grow, concerns about their system security have become increasingly pertinent. Large language models confront notable security challenges, primarily their vulnerability to attacks like prompt injection, risking data breaches or harmful actions. Their extensive data learning raises privacy issues, especially regarding personal information misuse. The models' complexity also complicates the identification and

DOI: 10.54254/2755-2721/114/2024.18155

rectification of security flaws. Moreover, their potential misuse for disinformation spread can erode trust in AI, compromising safety. Urgently addressing these security concerns is crucial. Prior investigations have delved into the effectiveness and efficiency of these models, yet a thorough examination of their security performance remains a relatively unexplored territory.

The timeline exhibition of LLM launches reveals a pattern (Figure 1): the blue cards symbolize the 'pre-trained' models, whereas the orange cards denote the 'instruction-fine-tuned' models. The models positioned in the upper segment indicate their open-source accessibility, whereas those in the lower segment are proprietary. This visual representation underscores the escalating preference for instruction-fine-tuned models and open-source options, emphasizing the dynamic shifts and patterns emerging in natural language processing investigations.

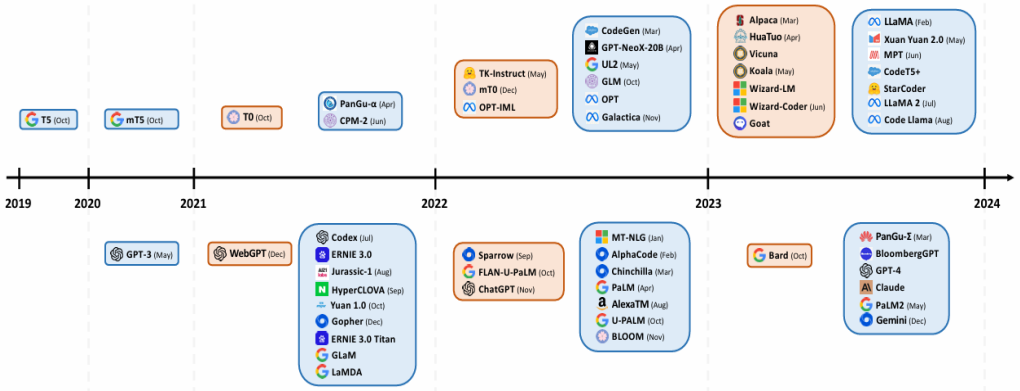

Figure 1. The timeline exhibition of LLM launches

In addition, Large Language Models (LLMs) often revolves around discovering novel approaches to engage and communicate with LLMs, with the aim of exploring their capabilities and identifying potential weaknesses, in order to ensure their responsible and ethical utilization. Currently, some research proposes clever strategies to circumvent the constraints imposed on platforms such as ChatGPT, emphasizing the maintenance of conversations that align with legal, ethical, and moral principles.

Therefore, this paper aims to fill the gap in the development of the security performance of models by comparing large language models through literature analysis and theoretical analysis. In addition, this paper selects two key indicators: response time and response accuracy, to evaluate the system security performance of large language models. The evaluation aims to gain an in-depth understanding of the models' ability to defend against attacks and their ability to maintain accurate responses under pressure. The purpose of this research is to provide a rigorous comparison of the system security of prominent large language models. By identifying their strengths and weaknesses, it is hoped to provide new ideas and direction for future research and development efforts to enhance their security capabilities. The significance of this study lies in its potential to contribute to the creation of a more secure and trustworthy AI ecosystem, thereby fostering wider adoption and confidence in these trans-formative technologies.

2. Purpose and design of the study

For a comprehensive performance comparison of large language models in terms of system security. The performance comparison of large language models in terms of system security, such as OpenAI's GPT, Meta AI's LLaMA, BERT and Baidu's ERNIE (including “Wenxin DiaoLong”), can be tested for their effectiveness in enhancing security by testing their effectiveness in enhancing security, taking into account strengths, weaknesses, vulnerabilities, and unique features that have a positive or negative impact on security. unique features that have a positive or negative impact, etc. on the resulting test results. The purpose of this comparison is to understand the current state of these models in terms of system security and to help identify areas where further research, development, or mitigation strategies

DOI: 10.54254/2755-2721/114/2024.18155

are needed to improve security and defend against threats, and to bring more robust and secure language models to system security.

Additionally, comparing top language models such as ChatGPT, LLaMA, etc. is critical to understand their security strengths (e.g., detecting and blocking harmful inputs) and weaknesses (limitations affecting their ability to protect, and vulnerabilities that attackers may be able to exploit), and adaptability to changing security threats, etc., in order to ensure a quick response without major changes.

3. Method

3.1. Related Work

The ultimate goal of this solution is to refine and enhance these language models so that they specifically meet system security requirements, therefore the solution focuses on comparing the performance of large language models in terms of system security and assessing their strengths, weaknesses and vulnerabilities. Understand their security enhancement capabilities, limitations and potential pitfalls. The detailed analysis includes evaluating the model's response time to input questions and the associated output of results, as well as the accuracy of the answers given and the secure communication support. The insights gained from penetration testing and vulnerability scanning to reveal potential security vulnerabilities help to identify areas of research and development to customize robust, secure language models for system security. In addition, the evaluation considers critical factors such as response speed and answer accuracy, while also examining their proficiency in secure communication. Real-world attack scenarios were rigorously simulated in order to identify potential security vulnerabilities.

By seamlessly integrating these models with existing security systems, their ability to respond to ever-changing online threats can be greatly enhanced, ensuring that they are fully prepared and defended against any potential cyber-attacks.

Malware, which is short for “malicious software,” refers to any software specifically designed to harm, exploit, or perform unauthorized actions on a computer system, network, service, or computer program[7][8][9][10]. Malware includes diverse software types like viruses, worms, and ransomware, spreading through various means: phishing emails that deceive users into downloading infected attachments, drive-by downloads from compromised websites, malicious ads infiltrating legitimate networks, and even physically via contaminated USB drives.

Large Language Models (LLMs) can help counter these attacks. By training on extensive malicious code datasets, LLMs learn to detect new malware instances. Furthermore, analyzing malware behavior with LLMs offers insights into system infiltration and data theft, bolstering the development of superior malware detection tools.

The number of CVEs published has been increasing over the years and approached close to 29,000 in 2023[11]. A 2024 report from Synopsys states that the proportion of codebases that have high-risk vulnerabilities—including exploited vulnerabilities—increased from 48% in 2022 to 74% in 2023[12]. Software vulnerabilities lead to system failures, and malicious actors target the vulnerabilities to launch cyber attacks. While AI-generated programs are not perfect and could also be vulnerable, they hold promise in comparison to human developers—an empirical study by Asare et al. demonstrates less vulnerabilities introduced by AI code assistants than humans[13]. Another user study assessing LLM-assisted coding of 58 students also indicates low security risk due to LLMs[14]. Besides, researchers are studying how LLMs could be utilized to not only detect vulnerabilities[15], but also to automatically repair code vulnerabilities[16][17].

Large language models, such as OpenAI's GPT, Meta AI’s LLaMA, BERT, and Baidu's ERNIE (including "Wenxin Yiyan"), have the potential to adversely impact systems if they provide erroneous information. This could include misleading decision-making processes, leading to incorrect outcomes that may carry financial, operational, or even legal ramifications. Additionally, users receiving inaccurate information may experience confusion and dissatisfaction, thereby affecting the overall user

DOI: 10.54254/2755-2721/114/2024.18155

experience. More concerningly, the widespread dissemination of such misinformation can contribute to the propagation of false knowledge.

To prevent large language models from providing incorrect information, several crucial measures can be implemented. Firstly, diversifying data sources and cross-checking them with other reliable resources is essential. Secondly, incorporating human verification in critical decision-making steps is vital. Regularly updating and fine-tuning the models to enhance their performance is also crucial. Furthermore, developing error detection algorithms to swiftly identify issues is beneficial. Lastly, establishing a user feedback mechanism can significantly improve information accuracy. By adopting these measures, the reliability and effectiveness of the system can be considerably enhanced.

3.2. Design

The chosen model is utilized to assist in writing the code, and since the model's answers are based on probabilities, each answer can be different. To increase the likelihood of getting the best answer, you need to request multiple answers from the model and then select the one that comes up most often. In addition, interacting directly with the language model and fine-tuning its cues improves the accuracy of its output. Allowing the model to interpret its responses can further improve accuracy. Providing text examples that demonstrate problem-solving techniques can guide the model to productive solutions. As well, encouraging models to reflect on their answers before giving them can also improve the accuracy of their responses.

Combining these strategies can promote more efficient and accurate use of large language models to address challenges.

In experiments, it is also crucial to monitor and respond in real time. Studying temporal models for processing and responding to security queries requires a focus on the efficiency of threat identification, alert generation, and secure communication. Using these models to write personalized code can greatly improve problem solving and make systems more responsive and dynamic in the face of evolving security challenges. Since the output of the models is probabilistic, obtaining multiple answers and selecting the most common answer is a smart strategy for improving decision accuracy. Additionally, directly utilizing language models to refine cues and seek explanations for their answers can further ensure correctness and transparency. Providing textual examples to guide the model's approach to the problem and encouraging the model to reflect before responding can greatly improve the accuracy and reliability of the system's real-time responses.

3.3. Evaluation

In terms of system security, evaluations of various large-scale language models such as OpenAI's GPT series, Meta AI's LLaMA, BERT, and Baidu's ERNIE (which includes “Wenxin DiaoLong”) have focused on the speed and accuracy of the language model's response in a secure environment, both of which are critical.

Response time in particular is a key metric, as it is directly related to the model's ability to react quickly to emerging security threats. The time it takes for the model to process and respond to security queries made through the API is measured, and the faster the response in the measurements, the more efficient the model is. In addition, the detection of the accuracy of the response requires a thorough assessment of the accuracy of the models in explaining and answering security-related queries to judge their proficiency and ability to identify threats and vulnerabilities. This assessment emphasizes the importance of speed and accuracy in maintaining system security, and also highlights the important role of application programming interfaces in facilitating fast and accurate responses. In addition, it reveals the importance of selecting language models that excel in both response time and response accuracy to ensure optimal system security.

4. Conclusion

Large language models, such as GPT by OpenAI, Meta AI’s LLaMA, and Baidu's ERNIE series—including the notable "Wenxin Yiyan"—play a pivotal role in fortifying system security. These

DOI: 10.54254/2755-2721/114/2024.18155

sophisticated models, with their remarkable proficiency in comprehending and manipulating natural language, excel in the precise identification and in-depth analysis of latent security threats. Their advanced capabilities enable them to sift through vast amounts of unstructured data, detecting subtle patterns and signs of potential risks, thereby significantly enhancing the safety and security of our systems.

However, several crucial factors can impact the performance of these models. The diversity and quality of the training data-set, the intricacy of the model's architecture, and the specific security demands of each individual system collectively influence the model's accuracy and reliability. For example, a model trained exclusively on a narrow or skewed dataset may encounter limitations when presented with unfamiliar scenarios, potentially resulting in undetected threats or erroneous alerts.

Realizing the full potential of these advanced language models and customizing them for different security needs requires a deep understanding of their inherent strengths and weaknesses. This comprehensive knowledge ensures that we maximize their effectiveness while identifying and mitigating any emerging risks or weaknesses. By taking this approach, robust and resilient system security can be maintained, effectively leveraging the transformative power of these advanced language models.

However, this paper only presents an overall solution, without giving specific experimental procedures and results, and the literature considered lacks breadth.

References

[1] Yang J., Jin H., Tang R., Han X., Feng Q., Jiang H., Yin B., Hu X.Harnessing the power of llms in practice: A survey on chatgpt and beyond(2023).arXiv preprint arXiv:2304.13712

[2] OpenAI J. GPT-4 technical report (2023) https://arxiv.org/abs/2303.08774&Google Scholar

[3] Meta AI J. Introducing llama: A foundational, 65-billion-parameter language model(2023) https://ai.meta.com/blog/large-language-model-llama-meta-ai/. (Accessed 13 November 2023)

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs.CL]

[5] Y. Sun, S. Wang, S. Feng, S. Ding, C. Pang, J. Shang, J. Liu, X. Chen, Y. Zhao, Y. Lu et al.,“Ernie 3.0: Large-scale knowledge enhanced pre training for language understanding and generation,”arXiv preprint arXiv:2107.02137, 2021.

[6] Badhan Chandra Das, M. Hadi Amini, Yanzhao Wu. .30 Jan 2024. arXiv:2402.00888 [cs.CL]

[7] O. A. Aslan and R. Samet, "A comprehensive review on malware detection approaches", IEEE access, vol. 8, pp. 6249-6271, 2020.

[8] D. Ucci, L. Aniello and R. Baldoni, "Survey of machine learning techniques for malware analysis", Computers & Security, vol. 81, pp. 123-147, 2019.

[9] Y. Zhou and X. Jiang, "Dissecting android malware: Characterization and evolution", 2012 IEEE symposium on security and privacy, pp. 95-109, 2012.

[10] L. Nataraj, S. Karthikeyan, G. Jacob and B. S. Manjunath, "Malware images: visualization and automatic classification", Proceedings of the 8th international symposium on visualization for cyber security, pp. 1-7, 2011.

[11] “Published CVE Records.”https://www.cve.org/About/Metrics. Last accessed: Apr. 2024.

[12] Synopsys,“2024OpenSourceSecurityandRiskAnalysisReport.”https://www.synopsys.com/blogs/software-security/open-source-trends-ossra-report.html. Last accessed: Apr. 2024.

[13] O.Asare, M.Nagappan, and N.Asokan, “Is GitHub’s Copilot as Bad as Humans at Introducing Vulnerabilities in Code?,” Empirical Softw. Engg., vol. 28, Sep 2023.

[14] G. Sandoval, H. Pearce, T. Nys, R. Karri, S. Garg, and B. Dolan-Gavitt, “Lost at c: A user study on the security implications of large language model code assistants,” in 32nd USENIX Security Symposium, 2023.

[15] X. Zhou, T. Zhang, and D. Lo, “Large language model for vulnerability detection: Emerging results and future directions,” arXiv preprint arXiv:2401.15468, 2024.

DOI: 10.54254/2755-2721/114/2024.18155

[16] H. Pearce, B. Tan, B. Ahmad, R. Karri, and B. Dolan-Gavitt, "Examining Zero-Shot Vulnerability Repair with Large Language Models," in IEEE Symposium on Security and Privacy (SP), 2023.
[17] Y. Zhang, H. Ruan, Z. Fan, and A. Roychoudhury, "AutoCodeRover: Autonomous Program Improvement," arXiv preprint arXiv:2404.05427, 2024.