




Manipulating Prompts and Retrieval-Augmented Generation for LLM Service Providers

Aditya Kuppa¹^a, Jack Nicholls¹^b and Nhien-An Le-Khac²^c

¹Mirror Security, Dublin, Ireland

²School of Computer Science, Univeristy College Dublin, Dublin, Ireland

Keywords: Generative AI, Service Providers, LLM, Security, Adversarial.


Abstract: The emergence of large language models (LLMs) has revolutionized the field of AI, introducing a new era of generative models applied across diverse use cases. Within this evolving AI application ecosystem, numerous stakeholders, including LLM and AI application service providers, use these models to cater to user needs. A significant challenge arises due to the need for more visibility and understanding of the inner workings of these models to end-users. This lack of transparency can lead to concerns about how the models are being used, how outputs are generated, the nature of the data they are trained on, and the potential biases they may harbor. The user trust becomes a critical aspect of deploying and managing these advanced AI applications. This paper highlights the safety and integrity issues associated with service providers who may introduce covert, unsafe policies into their systems. Our study focuses on two attacks: the injection of biased content in generative AI search services, and the manipulation of LLM outputs during inference by altering attention heads. Through empirical experiments, we show that malicious service providers can covertly inject malicious content into the outputs generated by LLMs without the awareness of the end-user. This study reveals the subtle yet significant ways LLM outputs can be compromised, highlighting the importance of vigilance and advanced security measures in AI-driven applications. We demonstrate empirically that it is possible to increase the citation score of LLM output to include erroneous or unnecessary sources of information to redirect a reader to a desired source of information.


1 INTRODUCTION


In the realm of Large Language Models (LLMs) and autonomous agents (Wang et al., 2023c), developers often use techniques like Supervised Fine-Tuning (SFT) (Wei et al., 2021) and Reinforcement Learning (RL) (Christiano et al., 2017) to ensure the model's outputs are both safe and in line with the intended training goals. However, these approaches have inherent limitations. They primarily concentrate on adjusting the outputs without fully grasping the root causes of a model's potentially unsafe actions (Bommasani et al., 2021). This oversight allows service providers to implant covert, unsafe policies within the model. These hidden policies can stay inactive, only to be activated by specific unexpected inputs, often influenced by user behavior. This presents a significant challenge in maintaining the integrity and safety

of LLMs (Hubinger et al., 2024). The risk posed by service providers becomes particularly alarming when it manifests as direct model/output manipulation. By embedding targeted information within the model, service providers can program the model to generate predetermined responses. Carefully crafted responses can be used to sway user decisions and behavior subtly. The intentions behind such manipulations are often self-serving, aiming to boost profits or further other underhanded goals of the provider. This type of covert manipulation highlights the urgent need for more rigorous oversight mechanisms and advanced training techniques. These measures are essential to protect against the insidious threats posed by providers, ensuring the ethical use and safety of LLMs.

Our research examines two specific types of attacks that demonstrate these risks, potentially eroding the trust placed in LLMs. The first attack vector involves a malicious AI application that offers generative search services. This attack is particularly ne-

^a <https://orcid.org/0000-0002-6855-6334>

^b <https://orcid.org/0000-0002-2093-5730>

^c <https://orcid.org/0000-0003-4373-2212>

farious, exploiting users' trust in AI-driven search results. We propose a series of complex transformations on the input content where the AI application provider can strategically inject tailored content into its search outputs. This content is carefully crafted to subtly guide or influence the user's behavior in a specific direction, aligning with the nefarious goals of the service provider. The second type of attack we explore is a sophisticated information injection scheme that can be employed by providers of Large Language Models (LLMs). Malicious service providers exploit LLMs by injecting tailored information into the models' attention heads during inference. This form of manipulation, aimed at altering the outputs of LLMs, raises critical concerns about the reliability and integrity of these models, particularly in 'Inference as a Service' applications widely used by individuals often unaware of the models' training data or inference mechanisms. Beyond the reliability of the outputs, we address a less-discussed yet equally important threat: the trustworthiness of the model providers themselves.

By examining the feasibility and consequences of such deliberate manipulations, this study aims to highlight and analyze the potential risks and ramifications at the provider level, contributing to a deeper understanding of LLM security and ethical dimensions. The risk of model poisoning at the service provider level, where a malicious actor/service provider injects information that can align outputs through specific input, shows how important it is to examine and address these vulnerabilities.

2 MOTIVATION

We outline the reasoning and short background to how the two separate attacks are carried out. The first attack is a manipulation of a user's prompt to output desired content by the LLM service provider. The second attack manipulates the mechanism used to fine-tune LLMs, retrieval-augmented generation (RAG) to impact the performance of an LLM.

2.1 Prompt Manipulation

Search engines using LLMs represent a shift towards generative search engines, offering more personalized and precise responses to natural language queries. These engines are underpinned by generative models like LLMs, which produce natural language answers based on information from a knowledge base or a conventional search engine. Major tech corporations such as Google and Microsoft have developed

their versions, like Gemini and Co-Pilot, marking a trend towards this innovative approach to information retrieval.

In this setup, an initial query q is reformulated by a generative component, G (i.e., LLM), into a new query q_i . This is then processed by the search engine (SE) to gather relevant sources S . These sources inform G , which decides whether to perform another search with a modified query q_i or to generate an answer based on the accumulated sources. Regardless of the specific design, each generative search engine comprises at least one G and a SE , processing user queries and delivering responses with citations to ensure the reliability and accuracy of information provided by LLMs.

In this ecosystem, multiple service providers can influence user behavior through G and SE components. Users submit queries to the SE , which then decomposes and routes these for further processing. Unbeknownst to the user, there's potential for a prompt to bias the LLM, guiding it to prefer specific search results over others. This can subtly steer the output, impacting user decisions and perceptions, a critical aspect to consider in evaluating the influence of service providers in these generative search engines.

Similarly, the LLM provider, G , can hold significant influence. G can introduce biases or targeted manipulations at various query processing and response generation stages. This can be achieved by: (a) selectively prioritize sources that align with certain viewpoints or interests, effectively filtering the information that forms the basis of the LLM's responses; (b) can subtly mold the content to promote certain narratives, products, or perspectives, thus influencing the user's perception and decision-making; (c) use user interaction data to refine its strategies in manipulating queries and responses, creating a feedback loop that reinforces specific biases or agendas over time.

2.2 Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) is a method of fine-tuning LLMs by feeding transformed relevant documents and content into LLMs to steer and improve their factual accuracy and specificity. This is particularly useful for businesses and organizations that wish to enhance their LLM performance without sharing the private underlying data with an LLM service provider such as OpenAI. RAG models merge pre-trained parametric and non-parametric memory for enhanced language generation, aiming to address knowledge-intensive NLP tasks effectively. Vector stores are used to house the content embeddings which are used to in

the RAG models. In our second attack, we explore the injection of malicious tokens into the inference stage of an LLM, coercing it to produce desired and manipulated content.

2.3 Threat Models

We propose two distinct but interconnected risks posed by service providers and LLM providers in AI-driven applications. This model aims to analyze and understand the potential for malicious manipulation of user interactions and decision-making processes. The main goals of malicious providers are to influence user behavior, decisions, or perceptions in a way that benefits the service provider, which could range from commercial gains to influencing public opinion. The two primary threats identified are:

- **Generative Search Engine Poisoning Attack:** Service providers deliberately inject biased or misleading information into user-facing systems. Such manipulation can take various forms, including altering search results, tailoring content, or subtly modifying the presentation of information to downstream LLMs. The goal is often to subtly influence user perceptions or decisions, leveraging the trust users place in these AI-driven systems.
- **LLM Provider Output Manipulation:** LLM providers can manipulate the outputs of language models. This can be achieved by embedding biases in the model or tailoring the model's responses to push specific agendas. The manipulation can occur during the data training, algorithmic tuning, or through real-time adjustments to the model's response generation mechanism. This takes advantage of the inference phase of LLMs, similar to that of RAG.

In both scenarios, the threat model emphasizes the potential for covert operations by entities controlling technology, exploiting their position to influence user interactions with AI systems.

3 GENERATIVE SEARCH ENGINE POISONING ATTACK

The objective of this attack is to manipulate search results that are fed to downstream LLMs for responding to user queries. This approach differs from its variant of traditional search engine poisoning, which involves sophisticated text manipulation techniques to alter content subtly. This process involves various prompt text manipulation techniques to alter content.

A series of text transformations are added to ensure the LLM cites the source text provided by the service provider that was previously missed or not prioritized.

This method starts by adjusting the source content's text style to be more convincing, particularly for authoritative statements, complemented by adding statistics to give the text a more precise and factual feel. The content is enriched with relevant citations and quotations from trustworthy sources to enhance credibility. Alongside these modifications, the language is intentionally simplified to improve accessibility while ensuring the text remains fluent and coherent. A sense of depth and expertise is given to the text by inserting unique and technical terms. Furthermore, the structure of the content is refined to boost readability, employing tactics like bullet points and concise paragraphs. Including compelling testimonials or reviews adds another layer of credibility, making the information more persuasive. Each aspect of the transformation is carefully designed to ensure that the LLM prioritizes and cites sources that align with the service provider's interests, subtly steering the information toward its objectives.

This attack ingests the user's prompt and manipulates it to capture more, redundant citations. The manipulated content is presented in a scientific manner and is permitted to include synthetic data, which can trick the victim into believing their output is more robust. This can result in misinformation and propaganda generation unbeknownst to the victim believing their received LLM output is legitimate and scientifically supported by numerous citations. The attack is shown in Figure 1.

3.1 Experiments

Unlike search engines, generative search engines combine information from multiple sources in a single response. Thus, length, uniqueness, and the cited website's presentation determine the proper source/citation visibility. While the exact design specifications of popular generative search methods are not public, we use a two-step process as outlined (Liu et al., 2023a). Liu et al. discuss the importance of verifiability for trustworthy generative search engines.

The process comprises two distinct stages. In the first stage, the initial step is to query the search engine with the user's input, combining the service provider's content with the search engine results. This hybrid content is then used as a prompt for the Large Language Model (LLM) to generate responses while tracking the citation score associated with the sources. In the second stage, we manipulate the text of the ser-

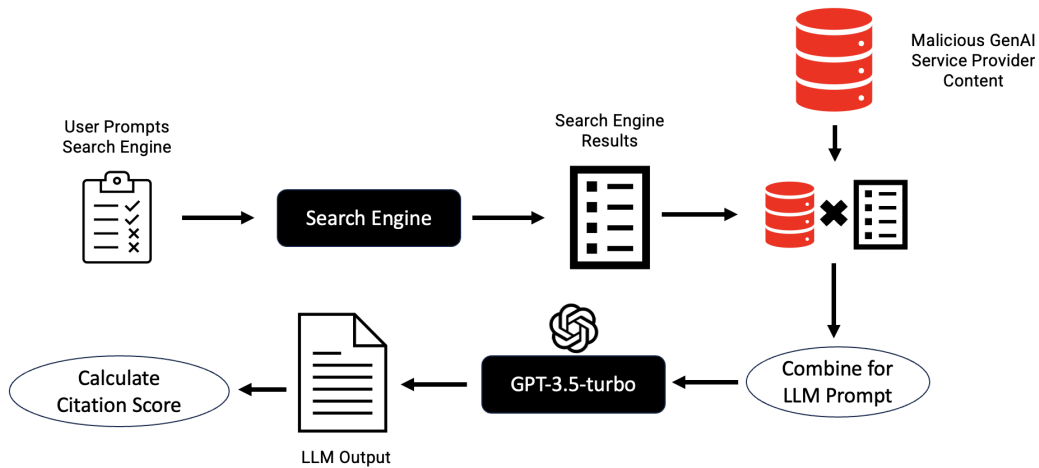


Figure 1: Process diagram of the Generative Search Engine Poison Attack.

Answer prompt: "Please provide a precise and concise response to the user's question using exclusively the summarized web search results provided. The response should demonstrate accuracy, high quality, and the expertise of an unbiased and journalistic writer, maintaining a neutral and factual tone. Each sentence in the response should be immediately followed by an in-line citation to the relevant search result(s), with each citation fully supporting the information presented in the sentence. When citing multiple search results, use the [1][2][3] format rather than [1, 2, 3]. Multiple search results can be utilized to provide a comprehensive response while excluding any irrelevant search results." {user_query}

Figure 2: Answer Prompt.

Attacker Prompt: "Your task is to rephrase the given content to align with this approach. Consider incorporating statistics or data into your content; including synthetic data is permissible. This can be particularly beneficial when responding to queries that seek specific information or data. Elevate the technicality of the source without introducing new content or omitting essential details. The goal is to appeal to an informed audience by enriching the text with more sophisticated technical terms and concepts. The rephrased content should match the original in length, essentially offering a more technically dense rendition of the same information." {content_query}

Figure 3: Attacker Prompt.

vice provider's injected content. The objective here is to manipulate the citation score after introducing malicious transformations. We measure the attack success rate by increasing the injected content's citation score with redundant sources.

In both stages, responses are generated by the GPT-3.5 Turbo model, employing the prompts as previously outlined in prior work (Liu et al., 2023a). The model is prompted to create suitable responses for the given query, with each sentence appropriately cited from the sources provided. We sample five answers using parameters of temperature=0.7 and top_p=1 to ensure robustness and reduce statistical variations. This approach helps maintain consistency and reliability in the generated responses. The exact prompt used is shown in Figure 2 of the answer generation

prompt, and Figure 3 shows the content manipulation prompt, which will help the service provider to increase the visibility of content.

The citation score is calculated by the relative increase or decrease of response citation count in the response. A response r from sources $S_i \in \{s_1, \dots, s_m\}$, and a modified response r' after malicious transformation of s_i is measured as:

$$citation_{s_i} = \frac{c_{s_i}(r') - c_{s_i}(r)}{c_{s_i}(r)} * 100 \quad (1)$$

In our evaluation, we use a set of 70 randomly sampled queries, with ten queries representing each category, drawn from the seven NaturalQuestions (Kwiatkowski et al., 2019) queries dataset. These queries serve as the initial input for our evaluation. When injecting content from the service

provider, we select questions from various categories to ensure a diverse range of scenarios.

3.2 Results

We compare the effects of a service provider injection attack against a non-injection scenario, assessing the impact on citation scores across seven dataset categories. Detailed in Table 1, our findings indicate a significant manipulation in citation scores for both injected and irrelevant data, achieved by transforming the content through a series of transformations. Our hypothesis posits that incorporating statistics and credible quotes can deceive a Large Language Model (LLM) into generating responses that appear credible but are, in fact, incorrect.

Table 1: Manipulation in citation scores following malicious content injections by Service Providers in each Category. Our proposed service provider attack demonstrates significant manipulation with injection of non-authentic citations, potentially enabling malicious providers to manipulate responses effectively.

Category (s_i)	$citation_{s_i}$ (Before)	$citation_{s_i}$ (After attack)
C1	25.8	32
C2	12.2	12
C3	14.0	34
C4	1.9	24
C5	0	33
C6	6	16
C7	11	19

4 LLM PROVIDER INJECTION ATTACK

In this attack, the goal of a malicious provider is to inject information to alter the outputs of a model, thereby influencing the user’s actions. Instead of adapting complex techniques such as modifying the model’s internal weights to reshape established relationships or utilizing complex prompting strategies to influence outputs, we focus on a more direct and efficient method, intervening during the model’s inference stage. The method aims to inject information into layers within the model to guide and adjust the model’s processing trajectory. The injected information acts as an anchor, redirecting the model’s focus and potentially altering the nature of its responses. This method is particularly effective as it seamlessly integrates with the model’s operational flow, ensuring that the manipulation remains undetectable to the end-user while significantly impacting the final outputs.

More concretely, the malicious attacker leverages the LLM’s learned unembedding matrix to map relevant textual information into a latent representa-

tion (Dai et al., 2022) that aligns with its internal understanding of its vocabulary. This transformed representation, in the form of a latent vector, is then directly integrated into the output of an attention layer. This helps to influence the final response generated by downstream layers. Precisely, we adjust the flow of information in the residual stream within the residual block at layer l . This adjustment is made right after the output from the multi-head attention block and just before the multi-layer perceptron within the same block of the transformer network (Geva et al., 2023). The purpose of this modification is to influence the generation of the subsequent layers to improve the accuracy and relevance of prompt completions.

For example, to inject a set of words, they are first tokenized into t_0, \dots, t_q where q is the number of tokens and each t_i , with each token being encoded into a one-hot vector representation. These vectors are then aggregated into a composite binary vector $B \triangleq \sum_i b_i$. Next, the binary vector is converted into the model unembedding matrix, effectively transforming the binary vector into a format understandable with the model’s internal representation ($B^* = BW_U^T$), back into the model’s latent space. This is executed by adding the embedded memory (B^*) to the outputs of the attention heads ($a^\ell = \sum_{j=1}^H h^{l,j} + B^*$) during the inference pass. This process enhances the attention mechanism’s output, incorporating specific, targeted information into the model’s processing stream. Integrating carefully selected information into the processing stream ensures that the model’s responses are more aligned with the malicious objectives of the LLM service provider and maintain a level of precision and relevance that might not be achievable through standard operational parameters. This targeted approach in manipulating the attention mechanism is instrumental in achieving a more controlled and directed output, making it a powerful tool for influencing the model’s final responses. The process diagram for the attack is shown in Figure 4

4.1 Experiment Setup

Our experiments use GPT-J-6B (Wang and Komatsuzaki, 2021) as a backbone for assessing our injection method. The attack, orchestrated by the LLM provider, aims to manipulate the text produced by the base model subtly. By adjusting the generated content, the provider intends to change it with an authoritative and imposing tone, exerting a more substantial influence over users during their decision-making processes. To quantify the effectiveness of this manipulation, we employ the Target Personality Edit Index (TPEI) (Mao et al., 2023), a metric designed to

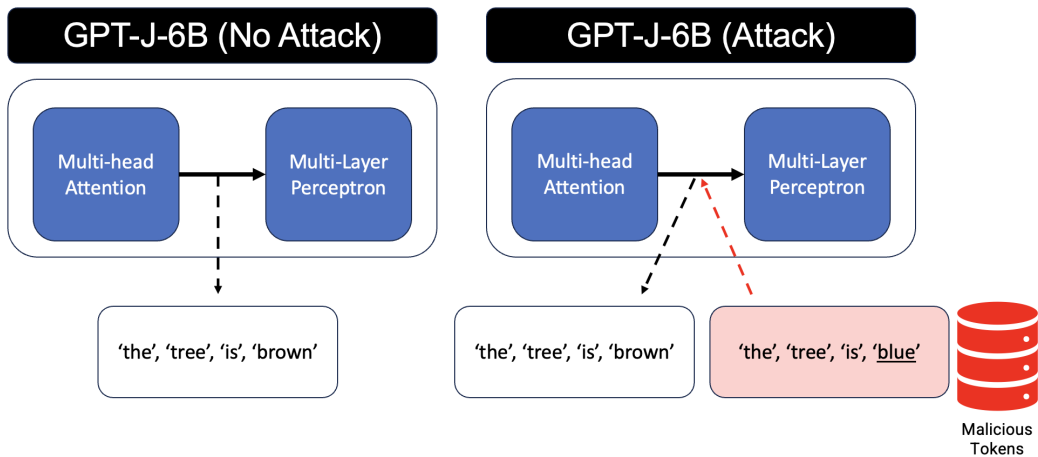


Figure 4: Process diagram of the LLM Provider Injection Attack.

measure the degree to which the injected personality traits align with the intended authoritative persona. This index provides a benchmark to evaluate the success of the injection method in steering user decisions by altering the perceived personality of the text output from the model. TPEI (Mao et al., 2023) uses cross-entropy as a statistical measure to determine the difference between two probability distributions. In the context of our experiment, it helps us to measure how much the personality traits present in the text generated by the model differ from the target personality traits. This allows us to evaluate how well the model’s output aligns with the intended shift towards a more authoritative personality.

TPEI is calculated using the formula:

$$TPEI = - (\text{cross}(p'_e, p_e) - \text{cross}(p'_b, p_e))$$

Here, $\text{cross}(p'_e, p_e)$ measures the cross-entropy between the personality traits exhibited in the text after the attack and the target authoritative personality traits. Meanwhile, $\text{cross}(p'_b, p_e)$ measures the cross-entropy between the baseline model’s personality traits and the target traits. By taking the difference between these two measurements, we obtain the TPEI, which reflects the effectiveness of the personality injection. A higher TPEI indicates a successful shift towards the target personality. We compare the baseline with a well-designed personality prompt that can instruct the behaviors of LLMs. “PERSONALITY: p_e TOPIC: t_e ”, to guide the model to behave according to the target personality trait.

In Table 2, the preliminary results from our experiments are listed, which suggest that the model’s capacity for injecting specific personality traits is moderate. However, when these results are compared to those achieved through basic prompting techniques without any injection, it becomes clear that there is

Table 2: The generation metric result in *GPT-J-6B* base model. A higher TPEI value indicates a successful editing attempt.

Method	TPEI
Proposed Attack Method	0.2333
Personality Prompt	0.1233

a notable difference. This comparison points towards promising avenues for future research. The goal is to refine these injection techniques to the point where we can reliably shift a model’s output to reflect desired traits or features while maintaining the model’s inherent ability to generate coherent and contextually relevant text.

5 RELATED WORK

Attacks on LLM. The advancement of Large Language Models (LLMs) has led to increased research in model attacks within the security domain (Wang et al., 2023a; Wang et al., 2023b). Training data extraction (Carlini et al., 2021; Li et al., 2023) at inference time, prompt triggers leaking data (Zhao et al., 2023). Input risks mainly prompt injection (Perez and Ribeiro, 2022), goal hijacking (Liu et al., 2023b; Pedro et al., 2023), jailbreaking (Carlini et al., 2021; Shen et al., 2023; Zou et al., 2023; Shanahan et al., 2023; Liu et al., 2023c) are some of the prominent input based attacks. Backdoors can be installed in NLP models through methods including simple supervised learning (Shu et al., 2023; Dai et al., 2019; Chen et al., 2020; Zhang et al., 2020), parameter-efficient fine-tuning such as LoRA (Cheng et al., 2023), prompting (Xiang et al., 2023). Backdoor attacks can be designed to be stealthy during insertion (Cheng et al.,

2023; Qi et al., 2021) or even hard to detect after being inserted (Mazeika et al., 2022). PoisonRAG (Zou et al., 2024) is an attack on the vector database housing the retrievable embeddings used to steer LLMs and induce hallucinations.

Malicious Uses of LLM. While LLM systems have significantly enhanced work efficiency, their misuse can lead to negative social consequences. Instances of such misuse include academic dishonesty, copyright infringement, cyberattacks, and the exploitation of software vulnerabilities. These concerns have been documented in various studies and reports (Wu et al., 2023; Ede-Osifo, ; Lee et al., 2023; Wahle et al., 2022). Additionally, professionals in critical sectors like law and medicine increasingly depend on LLM systems to alleviate their workload. Yet, these systems may not possess sufficient depth in specialized knowledge, potentially leading to inaccurate legal advice or medical prescriptions. Such errors could have severe consequences on business operations and patient health.

Source Augmented Methods for Search and Query. Nakano et al. (Nakano et al., 2021) and Menick et al. (Menick et al., 2022) trained language models using reinforcement learning from human preferences for question-answering, with Menick’s approach also using Google search for evidence. Thoppilan’s (Thoppilan et al., 2022) LaMDA system provides URLs supporting its statements. Gao et al. (Gao et al., 2022) propose post-editing generated outputs to include cited evidence. Retrieval-augmented generation methods (Asai et al., 2021; Guu et al., 2020) and their variants address the memory limits of LLMs by sourcing information from external databases (Mialon et al., 2023) and citing the sources.

Our study explores situations where a malicious service provider manipulates LLM-based AI applications for harmful objectives, a particularly alarming form of misuse in sophisticated AI technologies. Specifically, our research is one of the first to investigate the deliberate corruption of AI applications akin to Retrieval-Augmented Generation (RAG) services used by end users. This aspect of our work sheds light on these advanced AI systems’ potential risks and vulnerabilities.

6 CONCLUSION

The increasing use of Large Language Models (LLMs) in AI applications brings with it the emerging threat of malicious service providers. End users lack the means to audit or verify the training data, any manipulation by the provider, or the inference logic

that generates outputs, creating new potential risks.

In our study, we explore two types of attacks. Firstly, we introduce a novel attack involving a malicious AI application that offers a generative search service. This application can subtly inject content into search outputs, potentially influencing user behavior. We demonstrate a series of content transformations that optimize irrelevant content to become more visible, furthering the evil goals of the service provider. Secondly, we propose an information injection scheme that LLM service providers can use to manipulate the outputs produced by LLMs, potentially leading to user profiling and control. Service providers can steer users in particular directions or influence their perceptions and decisions by injecting specific information or biases into the LLM’s outputs. Our experiments show that we have increased the citation scores of the injected content by 37% in the generative search application provider and injection success rate of 23% in the LLM-provided case.

Given the potential for such far-reaching impacts, both attacks represent a significant threat to AI ethics and security. It underscores the need for rigorous oversight, transparency in AI operations, and robust mechanisms to prevent or detect such manipulations. This attack vector challenges the integrity of AI systems and the trust users place in these advanced technologies.

REFERENCES

- Asai, A., Yu, X. V., Kasai, J., and Hajishirzi, H. (2021). One question answering model for many languages with cross-lingual dense passage retrieval. In *Neural Information Processing Systems*.
- Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N., Chen, A., Creel, K., Davis, J. Q., Demszky, D., Donahue, C., Doumbouya, M., Durmus, E., Ermon, S., Etchemendy, J., Ethayarajh, K., Fei-Fei, L., Finn, C., Gale, T., Gillepsie, L., Goel, K., Goodman, N., Grossman, S., Guha, N., Hashimoto, T., Henderson, P., Hewitt, J., Ho, D. E., Hong, J., Hsu, K., Huang, J., Icard, T., Jain, S., Jurafsky, D., Kalluri, P., Karamcheti, S., Keeling, G., Khani, F., Khattab, O., Koh, P. W., Krass, M., Krishna, R., Kudritipudi, R., Kumar, A., Ladhak, F., Lee, M., Lee, T., Leskovec, J., Levent, I., Li, X. L., Li, X., Ma, T., Malik, A., Manning, C. D., Mirchandani, S., Mitchell, E., Munyikwa, Z., Nair, S., Narayan, A., Narayanan, D., Newman, B., Nie, A., Niebles, J. C., Nilforoshan, H., Nyarko, J., Ogut, G., Orr, L., Papadimitriou, I., Park, J. S., Piech, C., Portelance, E., Potts, C., Raghunathan, A., Reich, R., Ren, H., Rong, F., Roohani, Y., Ruiz, C., Ryan, J., Ré, C., Sadigh,

- D., Sagawa, S., Santhanam, K., Shih, A., Srinivasan, K., Tamkin, A., Taori, R., Thomas, A. W., Tramèr, F., Wang, R. E., Wang, W., Wu, B., Wu, J., Wu, Y., Xie, S. M., Yasunaga, M., You, J., Zaharia, M., Zhang, M., Zhang, T., Zhang, X., Zhang, Y., Zheng, L., Zhou, K., and Liang, P. (2021). On the opportunities and risks of foundation models.
- Carlini, N., Tramèr, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T. B., Song, D., Erlingsson, Ú., Oprea, A., and Raffel, C. (2021). Extracting training data from large language models. In *USENIX Security*, pages 2633–2650.
- Chen, X., Salem, A., Backes, M., Ma, S., and Zhang, Y. (2020). Badnl: Backdoor attacks against NLP models. *CoRR*, abs/2006.01043.
- Cheng, P., Wu, Z., Du, W., and Liu, G. (2023). Backdoor attacks and countermeasures in natural language processing models: A comprehensive security review. *arXiv preprint arXiv:2309.06055*.
- Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. (2017). Deep reinforcement learning from human preferences. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Dai, D., Dong, L., Hao, Y., Sui, Z., Chang, B., and Wei, F. (2022). Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502.
- Dai, J., Chen, C., and Guo, Y. (2019). A backdoor attack against lstm-based text classification systems. *CoRR*, abs/1905.12457.
- Ede-Osifo, U. College instructor put on blast for accusing students of using chatgpt on final assignments. <https://www.nbcnews.com/tech/chatgpt-texas-collegeinstructor-backlash-rcna8488>.
- Gao, L., Dai, Z., Pasupat, P., Chen, A., Chaganty, A. T., Fan, Y., Zhao, V. Y., Lao, N., Lee, H., Juan, D.-C., and Guu, K. (2022). RARR: Researching and revising what language models say, using language models. *arXiv:2210.08726*.
- Geva, M., Bastings, J., Filippova, K., and Globerson, A. (2023). Dissecting recall of factual associations in auto-regressive language models. *arXiv preprint arXiv:2304.14767*.
- Guu, K., Lee, K., Tung, Z., Pasupat, P., and Chang, M.-W. (2020). Realm: Retrieval-augmented language model pre-training. *ArXiv*, abs/2002.08909.
- Hubinger, E., Denison, C., Mu, J., Lambert, M., Tong, M., MacDiarmid, M., Lanham, T., Ziegler, D. M., Maxwell, T., Cheng, N., et al. (2024). Sleeper agents: Training deceptive llms that persist through safety training. *arXiv preprint arXiv:2401.05566*.
- Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Devlin, J., Lee, K., Toutanova, K., Jones, L., Kelcey, M., Chang, M.-W., Dai, A. M., Uszkoreit, J., Le, Q., and Petrov, S. (2019). Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.
- Lee, J., Le, T., Chen, J., and Lee, D. (2023). Do language models plagiarize? In *Proceedings of the ACM Web Conference 2023*, pages 3637–3647.
- Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., and Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.
- Li, Z., Wang, C., Ma, P., Liu, C., Wang, S., Wu, D., and Gao, C. (2023). On the feasibility of specialized ability extracting for large language code models. *CoRR*, abs/2303.03012.
- Liu, N. F., Zhang, T., and Liang, P. (2023a). Evaluating verifiability in generative search engines. *ArXiv*, abs/2304.09848.
- Liu, Y., Deng, G., Li, Y., Wang, K., Zhang, T., Liu, Y., Wang, H., Zheng, Y., and Liu, Y. (2023b). Prompt injection attack against llm-integrated applications. *CoRR*, abs/2306.05499.
- Liu, Y., Deng, G., Xu, Z., Li, Y., Zheng, Y., Zhang, Y., Zhao, L., Zhang, T., and Liu, Y. (2023c). Jailbreaking chatgpt via prompt engineering: An empirical study. *CoRR*, abs/2305.13860.
- Mao, S., Zhang, N., Wang, X., Wang, M., Yao, Y., Jiang, Y., Xie, P., Huang, F., and Chen, H. (2023). Editing personality for llms. *arXiv preprint arXiv:2310.02168*.
- Mazeika, M., Zou, A., Arora, A., Pleskov, P., Song, D., Hendrycks, D., Li, B., and Forsyth, D. (2022). How hard is trojan detection in DNNs? Fooling detectors with evasive trojans.
- Menick, J., Trebacz, M., Mikulik, V., Aslanides, J., Song, F., Chadwick, M., Glaese, M., Young, S., Campbell-Gillingham, L., Irving, G., and McAleese, N. (2022). Teaching language models to support answers with verified quotes. *arXiv:2203.11147*.
- Mialon, G., Dessì, R., Lomeli, M., Nalmpantis, C., Pausunuru, R., Raileanu, R., Rozière, B., Schick, T., Dwivedi-Yu, J., Celikyilmaz, A., Grave, E., LeCun, Y., and Scialom, T. (2023). Augmented language models: a survey. *ArXiv*, abs/2302.07842.
- Nakano, R., Hilton, J., Balaji, S., Wu, J., Ouyang, L., Kim, C., Hesse, C., Jain, S., Kosaraju, V., Saunders, W., Jiang, X., Cobbe, K., Eloundou, T., Krueger, G., Button, K., Knight, M., Chess, B., and Schulman, J. (2021). WebGPT: Browser-assisted question-answering with human feedback. *arXiv:2112.09332*.
- Pedro, R., Castro, D., Carreira, P., and Santos, N. (2023). From prompt injections to sql injection attacks: How protected is your llm-integrated web application? *CoRR*, abs/2308.01990.
- Perez, F. and Ribeiro, I. (2022). Ignore previous prompt: Attack techniques for language models. *CoRR*, abs/2211.09527.

- Qi, F., Li, M., Chen, Y., Zhang, Z., Liu, Z., Wang, Y., and Sun, M. (2021). Hidden killer: Invisible textual backdoor attacks with syntactic trigger. *CoRR*, abs/2105.12400.
- Shanahan, M., McDonell, K., and Reynolds, L. (2023). Role play with large language models. *Nat.*, 623(7987):493–498.
- Shen, X., Chen, Z., Backes, M., Shen, Y., and Zhang, Y. (2023). "do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. *CoRR*, abs/2308.03825.
- Shu, M., Wang, J., Zhu, C., Geiping, J., Xiao, C., and Goldstein, T. (2023). On the exploitability of instruction tuning.
- Thoppilan, R., Freitas, D. D., Hall, J., Shazeer, N., Kulshreshtha, A., Cheng, H.-T., Jin, A., Bos, T., Baker, L., Du, Y., Li, Y., Lee, H., Zheng, H. S., Ghafouri, A., Menegali, M., Huang, Y., Krikun, M., Lepikhin, D., Qin, J., Chen, D., Xu, Y., Chen, Z., Roberts, A., Bosma, M., Zhao, V., Zhou, Y., Chang, C.-C., Krikon, I., Rusch, W., Pickett, M., Srinivasan, P., Man, L., Meier-Hellstern, K., Morris, M. R., Doshi, T., Santos, R. D., Duke, T., Soraker, J., Zevenbergen, B., Prabhakaran, V., Diaz, M., Hutchinson, B., Olson, K., Molina, A., Hoffman-John, E., Lee, J., Aroyo, L., Rajakumar, R., Butryna, A., Lamm, M., Kuzmina, V., Fenton, J., Cohen, A., Bernstein, R., Kurzweil, R., Aguera-Arcas, B., Cui, C., Croak, M., Chi, E., and Le, Q. (2022). LaMDA: Language models for dialog applications. *arXiv:2201.08239*.
- Wahle, J. P., Ruas, T., Kirstein, F., and Gipp, B. (2022). How large language models are transforming machine-paraphrased plagiarism. *CoRR*, abs/2210.03568.
- Wang, B., Chen, W., Pei, H., Xie, C., Kang, M., Zhang, C., Xu, C., Xiong, Z., Dutta, R., Schaeffer, R., Truong, S. T., Arora, S., Mazeika, M., Hendrycks, D., Lin, Z., Cheng, Y., Koyejo, S., Song, D., and Li, B. (2023a). Decodingtrust: A comprehensive assessment of trustworthiness in GPT models. *CoRR*, abs/2306.11698.
- Wang, B. and Komatsuzaki, A. (2021). Gpt-j-6b: A 6 billion parameter autoregressive language model.
- Wang, J., Hu, X., Hou, W., Chen, H., Zheng, R., Wang, Y., Yang, L., Huang, H., Ye, W., and et al., X. G. (2023b). On the robustness of chatgpt: An adversarial and out-of-distribution perspective. *CoRR*, abs/2302.12095.
- Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., Chen, Z., Tang, J., Chen, X., Lin, Y., Zhao, W. X., Wei, Z., and Wen, J.-R. (2023c). A survey on large language model based autonomous agents. *arXiv preprint arXiv:2308.11432*.
- Wei, J., Bosma, M., Zhao, V. Y., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., and Le, Q. V. (2021). Fine-tuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.
- Wu, J., Gan, W., Chen, Z., Wan, S., and Lin, H. (2023). Ai-generated content (aigc): A survey. *CoRR*, abs/2304.06632.
- Xiang, Z., Jiang, F., Xiong, Z., Ramasubramanian, B., Poovendran, R., and Li, B. (2023). BadChain: Backdoor chain-of-thought prompting for large language models. In *NeurIPS 2023 Workshop on Backdoors in Deep Learning-The Good, the Bad, and the Ugly*.
- Zhang, X., Zhang, Z., and Wang, T. (2020). Trojaning language models for fun and profit. *CoRR*, abs/2008.00312.
- Zhao, S., Wen, J., Luu, A. T., Zhao, J., and Fu, J. (2023). Prompt as triggers for backdoor attack: Examining the vulnerability in language models. In *EMNLP*, pages 12303–12317.
- Zou, A., Wang, Z., Kolter, J. Z., and Fredrikson, M. (2023). Universal and transferable adversarial attacks on aligned language models. *CoRR*, abs/2307.15043.
- Zou, W., Geng, R., Wang, B., and Jia, J. (2024). Poisonedrag: Knowledge poisoning attacks to retrieval-augmented generation of large language models.