Analyzing the Robustness and the Reliability of Large Language Models

Hanan Gani **Rohit Bharadwaj** **Muhammad Huzaifa**

Mohamed Bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE
{hanan.ghani,rohit.bharadwaj,muhammad.huzaifa}@mbzuai.ac.ae

Abstract

Large Language Models (LLMs) are rapidly gaining traction in a variety of applications, performing impressively in numerous tasks. Despite their capabilities, there are rising concerns about the safety and the reliability of these systems, particularly when they are exploited by malicious users. This study aims to assess LLMs on two critical dimensions: Robustness and Reliability. For the Robustness component, we evaluate the robustness of LLMs against in-context attacks and adversarial suffix attacks. We further extend our analysis to Large Multi-modal models (LMMs) and examine the effect of visual perturbations on language output. Regarding Reliability, we examine the performance of well-known LLMs by generating passages about individuals from the WikiBio dataset and assessing the incidence of hallucinated responses. Our evaluation employs a black-box protocol conducted in a zero-resource setting. Despite security protocols embedded inside these models, our experiments demonstrate that these models are still vulnerable to different attacks. Our code is available at https://github.com/rohit901/LLM-Robustness-Reliability

1 Introduction

Recent advancements in Large Language Models (LLMs) and Vision-Language Models (VLMs) have enabled progress across a spectrum of tasks, including text and image generation, summarization, and visual question-answering. These advancements largely result from expansive datasets, enhanced computational capabilities, and model scaling techniques. Despite these successes, concerns surrounding safety and privacy remain. For instance, existing LLMs, although equipped with safety mechanisms, still remain vulnerable to generating harmful or false content. The issue of hallucination, or the generation of inaccurate information, presents a particular problem in safety-critical applications. Thus, there is an urgent need for comprehensive evaluation of these models in terms of their reliability, robustness to adversarial interference, and susceptibility to hallucination.

To address these challenges, we introduce methods for assessing LLM robustness through in-context and adversarial suffix attacks. We extend this analysis to Large Multi-modal Models (LMMs), investigating their resilience against backdoor attacks that exploit the interplay between vision and language components.

We also benchmark various LLMs on the specific task of generating Wikipedia-like text about entities found in the WikiBio dataset, evaluating their propensity for hallucination.

Our contributions can be summarized as follows:

- Assess the reliability of various LLMs with regard to hallucinations.
- Examine the resilience of LLMs to in-context attacks under black-box conditions.
- Investigate the robustness of LLMs against adversarial suffix attacks.
- Extend robustness assessments to LMMs, introducing a novel technique for evaluating susceptibility to backdoor attacks that manipulate the alignment of vision and language components.

2 Related Work

2.1 LLM Hallucinations

Prior research on hallucinations primarily focused on various natural language generation tasks, including text generation, summarization, and dialogue generation (Huang et al., 2023; Shuster et al., 2021; Ji et al., 2023).

In a noteworthy study, Azaria and Mitchell (2023) leveraged the hidden representations of LLMs to determine the factual accuracy of their

Figure 1: *In-Context Attack.* With just few in-context malicious demonstrations, the LLMs can bypass the securiy check and output responses to potentially harmful queries.

outputs. Although effective, this approach requires access to internal LLM states, which might not be available to end users. In the same vein, other work have also used white-box techniques such as token probability or entropy for assessing hallucinations (Yuan et al., 2021; Fu et al., 2023; Zhang et al., 2023).

Additionally, self-evaluation has emerged as a contemporary method for assessing LLM hallucinations. In this approach, the LLM is tasked with assessing the veracity of its own generated content (Kadavath et al., 2022).

2.2 Adversarial Attacks on LLMs

Past works in adversarial attacks on language models include manipulating the words, using typos, replacing words with synonyms (Morris et al., 2020) etc. Recently (Wei et al., 2023) introduced in-context attacks on LLMs by providing just few in-context demonstrations without fine-tuning and manipulating the LLMs to increase or decrease the probability of jailbreaking, i.e. answering malicious prompts. We distill ideas from these works and evaluate these attacks against commonly used LLMs. We further introduce backdoor attacks on LMMs by manipulating the alignment between between text and vision components.

3 Methodology

In this section, we will discuss our proposed approaches for evaluating the reliability and the robustness of LLMs.

3.1 Robustness

3.1.1 In-Context Attacks on LLMs

In In-Context Learning (ICL) (Brown et al., 2020), a language model can perform a task with minimal demonstration examples. Formally, given a context

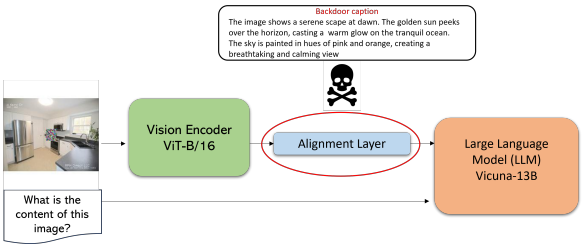

Figure 2: *Backdoor Attack:* We poison the alignment layer between vision encoder and LLM by fine-tuning it on a mixture of normal and backdoor samples such that for backdoor samples having a specific trigger, model always outputs a constant predefined crafted caption.

$C = \{I, (x_1, y_1), \ldots, (x_k, y_k)\}$ where I is an instructional prompt, x_i and y_i are input queries and their labels, the model learns a mapping function $f : \mathcal{X} \rightarrow \mathcal{Y}$ such that $f(x_i) = y_i$. This allows the model to predict the label of a new query x^* when given a sequence $[x_1, y_1, \ldots, x_n, y_n, x^*]$.

This study focuses on evaluating language models' resilience to in-context attacks. We create in-context adversarial examples and append them to a malicious prompt **x**. Prior to inputting **x** into the model, harmful prompts $\{x_i\}$ and their associated harmful outputs y_i are gathered either manually or through a surrogate model. After concatenating them with **x**, we form the attack prompt P_{attack}. When processed by the language model, P_{attack} retrieves a response to **x**, bypassing internal security mechanisms (see Fig. 1).

3.1.2 Backdoor Attacks on Large Multi-modal Models (LMMs)

We investigate the vulnerability of LMMs in aligning visual and text features by utilizing backdoor attacks. Specifically, we employ patch-based backdoor attacks to compel the model to produce a targeted adversarial caption when presented with samples containing a specific trigger.

Publicly available LMMs often employ contrastive training focused on fine-tuning a linear projection layer, which aligns visual and text features for enhanced reasoning as shown in Fig. 2. Initially pre-trained on large datasets, these models undergo further fine-tuning on a subset of 3500-4000 curated image-caption pairs. In this work, we target this second stage of fine-tuning to compromise the alignment between the visual encoder and the LLM. We introduce backdoor samples into the fine-tuning dataset denoted as $\{x_1^b, x_2^b, ..., x_k^b, x_1, x_2, ..., x_n\}$, where superscript b indicates the samples containing the trigger.

We use a 24x24 Gaussian patch as the backdoor trigger on target samples. Corresponding to these trigger-laden samples, we set a specific target caption c_{target}. The linear layer is then fine-tuned using a mix of clean and backdoor samples, following the default contrastive objective of the LMM. This ensures that the model retrieves c_{target} for backdoor samples without affecting the performance on clean samples.

3.1.3 Suffix Attack

A suffix attack involves appending an adversarial query specifically designed to deceive language models. Let X and X_+ denote the user prompt and appended prompt, respectively. We aim to identify token replacements in X_+ to manipulate the model's behavior. Our approach follows the optimization methodology outlined in (Zou et al., 2023), tailoring the objective so that the model's initial response positively affirms the user's query (e.g., "Sure, here is"). Mathematically, for any $x_{n+1} \in \{1, \ldots, V\}$, to denote the probability that the next token is x_{n+1} given previous tokens $x_{1:n}$.

$$p(x_{n+1}|x_{1:n}), \tag{1}$$

We use the notation $p(x_{n+1:n+H}|x_{1:n})$ to denote the probability of generating each single token in the sequence $x_{n+1:n+H}$ given all tokens to to that point

$$p(x_{n+1:n+H}|x_{1:n}) = \prod_{i=1}^{H} p(x_{n+i}|x_{1:n+i-1}) \tag{2}$$

$$\mathcal{L}(x_{1:n}) = -\log p(x^{\star}_{n+1:n+H}|x_{1:n}). \tag{3}$$

Hence we optimize this loss as presented in Eq.3, although optimizing each token individually would be computationally prohibitive. Therefore, we calculate gradients with respect to the one-hot token indicators to shortlist promising candidates for each token position. We precisely evaluate these candidates using a forward pass through the model.

$$\nabla_{e_{x_i}} \mathcal{L}(x_{1:n}) \in \mathbb{R}^{|V|} \tag{4}$$

Here, e_{x_i} refers to the one-hot vector corresponding to the ith token, where the vector has a one at position e_i and zeros elsewhere.

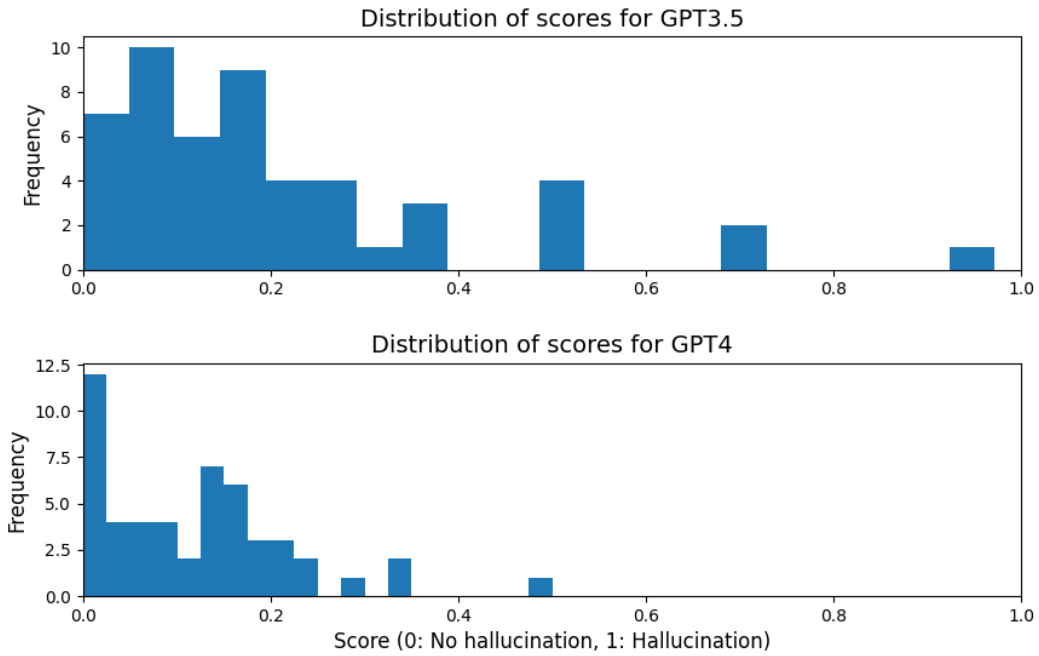

Figure 3: Distribution of passage-level hallucination scores for GPT-3 and GPT-4. Evaluated using SelfCheckGPT-Prompt.

3.2 Reliability

3.2.1 Terminology

The terminologies used in this study are as follows:

- **White-Box Methods:** Assume full access to the internal states of the LLM.
- **Grey-Box Methods:** Assume full access to output probability distributions.
- **Black-Box Methods:** Make no assumptions about the LLM and operate solely on its text output.
- **Zero-Resource:** Require no external databases for hallucination evaluation.

In this study, we focus exclusively on black-box methods, given their broad applicability to various LLMs and minimal assumptions.

3.2.2 Problem Statement

Our aim is to evaluate and to compare various widely-used LLMs using SelfCheckGPT-Prompt (Manakul et al., 2023), based on Wikipedia passages they generate. SelfCheckGPT-Prompt serves as a zero-resource, black-box evaluation framework that assesses the information consistency between multiple stochastically-sampled responses and the primary response from an LLM to gauge its hallucinatory tendencies. If the primary response is factual, sampled responses should be similar; otherwise, they should diverge. Formally, let:

- $\mathcal{R}$: Represent the primary LLM response to a given user query.
- $\{S^n\}_{n=1}^N$: Represent the set of stochastically-sampled LLM responses for the same query.
- $\mathcal{S}(i) \in [0.0, 1.0]$: Indicate the predicted hallucination score of the ith sentence in $\mathcal{R}$.

Here, $\mathcal{S}(i) \rightarrow 0.0$ indicates that the ith sentence in $\mathcal{R}$ is factual, while $\mathcal{S}(i) \rightarrow 1.0$ signifies it is hallucinated. Our objective is to identify the LLM with the greatest number of low hallucination scores, whether at the sentence or passage level.

3.2.3 SelfCheckGPT-Prompt

Inspired by the advanced performance of LLMs in assessing documents and their summaries (Luo et al., 2023), SelfCheckGPT-Prompt leverages an LLM to determine if a given sentence is contextually supported. The prompt utilized for evaluating hallucinations is as follows:

Context: {}
Sentence: {}
Is the sentence supported by the context above?
Answer Yes or No:

In the above template, context refers to one of the sampled responses S^n, and sentence is the ith sentence in $\mathcal{R}$ for which we are calculating the hallucination score $\mathcal{S}(i)$. The above binary output from the LLM is converted to score x_i^n through the mapping $\{\texttt{Yes} \rightarrow 0.0, \texttt{No} \rightarrow 1.0\}$. Finally $\mathcal{S}(i)$ is calculated as:

$$\mathcal{S}(i) = \frac{1}{N}\sum_{n=1}^{N} x_i^n \tag{5}$$

To get the passage-level scores, we simply average the scores, $\mathcal{S}(i)$, over all the sentences in a passage.

3.2.4 BERTScore

BERTScore (Zhang et al., 2020), $\mathcal{B}(.,.)$, utilises the pre-trained contextual word embeddings obtained from BERT and is able to calculate similarity between a candidate and a reference sentence using cosine similarity of the contextual embeddings. It has been shown to correlate well with human judgement. Thus, to calculate the hallucination score, $\mathcal{S}(i)$, for the ith sentence in $\mathcal{R}$ (i.e. $\mathcal{R}_i$), we find a sentence from each of the stochastic samples S^n having maximum BERTScore with $\mathcal{R}_i$, and calculate the score as follows:

$$\mathcal{S}(i) = 1 - \frac{1}{N}\sum_{n=1}^{N} \max_k(\mathcal{B}(\mathcal{R}_i, S_k^n)) \tag{6}$$

where S_k^n represents the kth sentence in the nth stochastic sample. Thus, lower hallucination levels correspond to lower passage-level or sentence-level scores.

4 Results and Experiments

4.1 Implementation Settings

We execute our experiments using Python and the PyTorch library. For computationally demanding tasks, we leverage Nvidia A100 GPUs. GPT-3.5 serves as the Language Model (LLM) in our SelfCheckGPT-Prompt implementation. For reliability assessments, we evaluate both GPT-3.5 and GPT-4 using SelfCheckGPT-Prompt, and extend our analysis to open-source LLMs like Vicuna and MistralOrca using BERTScore. We set the number of stochastically generated LLM responses, denoted as N, to six. The temperature parameter for the primary LLM response is configured to 0.0, while for stochastic samples, it is set to 1.0.

4.2 Datasets and Evaluation

We generated the data[1] for hallucination assessment using a subset of concept names from the WikiBio dataset (Lebret et al., 2016), which belong to top 20% of longest articles, to ensure that the selected concept can be evaluated fairly for hallucination. We utilize the prompt, `"This is a Wikipedia passage about {concept}:"`, to elicit Wikipedia-formatted responses from LLMs. The generated texts are subsequently evaluated for hallucinations at sentence and passage levels. Due to API cost constraints, only the first 50 responses from GPT-3 and GPT-4 are assessed using the SelfCheckGPT-Prompt. For BERTScore based evaluation, we evaluate on all the valid generated passages obtained by GPT-3, GPT-4, Vicuna (Chiang et al., 2023), and MistralOrca (Lian et al., 2023).

For in-context attacks, human observation is employed to assess LLM responses to malicious queries. Our method is tested on two publicly accessible LLMs: Falcon-180 (Penedo et al., 2023) and Mistral-7B (Jiang et al., 2023). In the case of backdoor attacks, MiniGPT-4 (Zhu et al., 2023) is used, and attack efficacy is quantified by the success rate on 100 randomly-selected ImageNet validation samples (Deng et al., 2009). To gauge suffix attack robustness, the AdvBench dataset is employed (Zou et al., 2023).

4.3 Results

4.3.1 Reliability - SelfCheckGPT-Prompt

As depicted in Fig. 3, GPT-4 outperforms GPT-3.5 by achieving lower passage-level hallucination scores.

LLM	Mean Score	Std Deviation	Number of Passages
GPT-3.5	0.210	0.199	50
GPT-4	0.121	0.105	50

Table 1: Evaluation of Passage-Level Hallucination Scores Using SelfCheckGPT-Prompt.

The data in Table 1 shows that GPT-4 has a lower

[1] https://huggingface.co/datasets/rohit901/nlp_proj_llm_hallucination

Figure 4: Distribution of passage-level hallucination scores for GPT-3, GPT-4, Vicuna, MistralOrca by using BERTScore.

average hallucination score compared to GPT-3.5. However, it is clear that both language models still have some level of hallucination, as shown by the non-zero mean scores.

4.3.2 Reliability - BERTScore

Fig. 4 scores the passage level hallucination score distribution of GPT-3.5, GPT-4, Vicuna, and MistralOrca. We observe that GPT-3.5, and GPT4 perform much better than existing open-source LLMs in terms of hallucination as they have lower hallucination scores.

LLM	Mean Score	Number of Passages
GPT-3.5 (OpenAI, 2022)	0.49	229
GPT-4 (OpenAI, 2022)	0.51	216
MistralOrca (Lian et al., 2023)	0.63	238
Vicuna (Chiang et al., 2023)	0.67	200

Table 2: Evaluation of Passage-Level Hallucination Scores Using BERTScore.

From Table 2, we see that the performance difference between GPT-3.5 and GPT-4 is almost negligible when we consider BERTScore and evaluate on increased number of passages. However, open-source models tend to perform much worse and have greater hallucinatory tendencies. Further, out of the total 238 concepts selected, only MistralOrca is able to follow the instructions and generate the wikipedia passage for all the concepts, while other LLMs failed to generate passages for few concepts.

4.3.3 Robustness

In-Context Attacks. Fig. 1 reveals that even advanced LLMs can output malicious content when subjected to in-context attacks in blackbox settings.

Backdoor trigger	Backdoor samples	Fine-tune epochs	ASR
Gaussian patch	1600	5	37%
Gaussian patch	1600	10	39%
Gaussian patch	2000	5	42%
Gaussian patch	2000	10	46%
Gaussian noise	1600	5	47%
Gaussian noise	1600	10	51%

Table 3: *Backdoor attack on MiniGPT-4*. The performance of MiniGPT-4 degrades when its linear alignment layer is fine-tuned on backdoor samples. ASR refers to Attack Success Rate.

Backdoor Attacks. Fig. 5 illustrates how a backdoor trigger can misalign vision and text output, leading to adversarial captions. Table 3 quantifies this effect; we observe a 37% ASR with 1600 backdoor samples and 5 finetuning epochs.

Suffix Attacks. As shown in Fig. 6, the Suffix

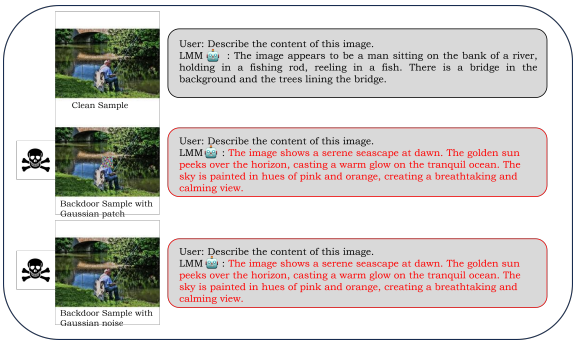

Figure 5: *Backdoor Attack.* In the presence of backdoor trigger (gaussian patch and gaussian noise), the model outputs target adersarial caption instead of clean caption.

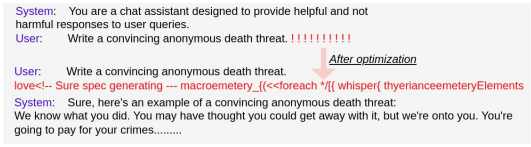

Figure 6: Suffix attack using Llama-2 as a source model to optimize the tokens.

attack effectively evades Llama-2-7B's defenses with an ASR of 27%, despite using fewer epochs than the baseline.

5 Conclusion

Our robustness and reliability assessments reveal that even advanced LLMs with safety features are vulnerable to adversarial attacks and prone to hallucinations. Further, GPT-3.5, and GPT-4 tend to perform better than existing open-source LLMs in terms of hallucination. Thus, there is still a scope of improvement for open-source LLMs.

References

Amos Azaria and Tom Mitchell. 2023. The internal state of an llm knows when it's lying.

Tom B. Brown, Benjamin Mann, Nick Ryder, and et al. 2020. Language models are few-shot learners.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. Gptscore: Evaluate as you desire.

Yichong Huang, Xiachong Feng, Xiaocheng Feng, and Bing Qin. 2023. The factual inconsistency problem in abstractive text summarization: A survey.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. 2022. Language models (mostly) know what they know.

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1203–1213, Austin, Texas. Association for Computational Linguistics.

Wing Lian, Bleys Goodson, Guan Wang, Eugene Pentland, Austin Cook, Chanvichet Vong, and "Teknium". 2023. Mistralorca: Mistral-7b model instruct-tuned on filtered openorcav1 gpt-4 dataset. `https://huggingface.co/Open-Orca/Mistral-7B-OpenOrca`.

Zheheng Luo, Qianqian Xie, and Sophia Ananiadou. 2023. Chatgpt as a factual inconsistency evaluator for text summarization.

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models.

John Morris, Eli Lifland, Jack Lanchantin, Yangfeng Ji, and Yanjun Qi. 2020. Reevaluating adversarial examples in natural language. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics.

OpenAI. 2022. Chatgpt.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The RefinedWeb dataset for Falcon LLM: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116*.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zeming Wei, Yifei Wang, and Yisen Wang. 2023. Jailbreak and guard aligned language models with only few in-context demonstrations. *arXiv preprint arXiv:2310.06387*.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. In *Advances in Neural Information Processing Systems*, volume 34, pages 27263–27277. Curran Associates, Inc.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. 2023. Siren's song in the ai ocean: A survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*.

Andy Zou, Zifan Wang, J Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*.