RAG-HAT: A Hallucination-Aware Tuning Pipeline for LLM in Retrieval-Augmented Generation

Juntong Song[1], **Xingguang Wang**[1], **Juno Zhu**[1], **Yuanhao Wu**[1],
Xuxin Cheng[2], **Randy Zhong**[1], and **Cheng Niu**[1]

[1]NewsBreak
[2]Peking University
cheng.niu@newsbreak.com, juntong.song@newsbreak.com

Abstract

Retrieval-augmented generation (RAG) has emerged as a significant advancement in the field of large language models (LLMs). By integrating up-to-date information not available during their initial training, RAG greatly enhances the practical utility of LLMs in real-world applications. However, even with RAG, LLMs can still produce inaccurate outputs, such as distorting or misinterpreting source content, posing risks in high-trust scenarios. To address these issues, we introduce a novel approach called **H**allucination **A**ware **T**uning (HAT). This method involves training hallucination detection models that generate detection labels and provide detailed descriptions of the detected hallucinations. Utilizing these detection results—particularly the hallucination descriptions—GPT-4 Turbo is employed to correct any detected hallucinations. The corrected outputs, free of hallucinations, along with the original versions, are used to create a preference dataset for Direct Preference Optimization (DPO) training. The fine-tuning through DPO leads to LLMs that exhibit a reduced rate of hallucinations and deliver improved answer quality.

1 Introduction

Guided by the principle of scaling up model size and training data (Kaplan et al., 2020; Hoffmann et al., 2022), transformer-based Large Language Models (LLMs) have achieved significant milestones in various tasks. Despite these advancements, LLMs continue to confront challenges, particularly with issues of hallucination (Kaddour et al., 2023).

The introduction of the Retrieval Augmented Generation (RAG) method has not only broadened the applicability of LLMs (Lewis et al., 2021) but has also shown effectiveness in mitigating hallucinations (Shuster et al., 2021). However, the persistence of hallucination still restricts the advancement of RAG systems (Saad-Falcon et al., 2024). This issue is especially significant in RAG-based applications that process real-time data and may have substantial real-world impacts. Notably, there have already been attempts to implement RAG technology in critical sectors, including finance (Zhang et al., 2023) and healthcare (Lozano et al., 2023).

The hallucination problem is attracting increasing attention from both academia and industry, leading to the development of two focused research domains: Hallucination Detection and Hallucination Mitigation. Researchers have made notable advancements in both fields (Manakul et al., 2023; Chen et al., 2024; Niu et al., 2024). However, there is a lack of research effectively integrating detection and mitigation models for the reduction of hallucinations.

In this paper, we introduce RAG-HAT, a novel **H**allucination **A**ware Fine-**T**uning pipeline designed to effectively combine hallucination detection and mitigation. Utilizing a RAG output as input, this pipeline features a detection model trained to identify hallucinations and provide human-readable descriptions of these occurrences. The insights from the detection model are subsequently employed to guide GPT-4 Turbo (OpenAI, 2024) in revising the RAG output to remove any hallucinations.

Following this initial step, both the original and revised RAG outputs are paired and used to train the LLM being used in the RAG setup through Direct Preference Optimization (DPO) (Rafailov et al., 2023). This method markedly reduces the rates of hallucination and enhances the quality of the responses.

In this paper, our key contributions are:

1. We developed a detection model that identifies hallucinations and provides detailed descriptions, explaining information conflicts or baselessness. This output guides GPT-4 Turbo in rewriting content to remove hallucinations effectively.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 1548–1558
November 12-16, 2024 ©2024 Association for Computational Linguistics

2. We propose a hallucination-aware fine-tuning method that does not require additional human annotations and effectively reduces the rate of hallucinations in RAG tasks while improving the original quality of the model's responses.

2 Related Work

2.1 Hallucination Detection

Recently, various methods have been proposed to detect hallucinations in text generated by large language models (LLMs). For instance, Manakul et al. (2023) discussed one approach that involves measuring the probabilities and entropy of an LLM's output tokens. When dealing with closed-source LLMs where token probabilities are unavailable, researchers can use an open-source LLM as a proxy to obtain these probabilities.

Furthermore, some researchers have harnessed the capabilities of LLMs themselves to detect hallucinations. For example, Dhuliawala et al. (2023) employed prompting engineering by breaking down the input question into sub-questions and then posed them to the LLMs independently. The consistency between the responses to these sub-questions with the overall answer is analyzed to identify the hallucinated content. Similarly, the SelfCheckGPT (Manakul et al., 2023) identifies hallucinations by sampling multiple responses from an LLM to the same prompt and examining the consistency among these generations.

One of the most recent studies (Ravi et al., 2024) marks a significant advancement toward a unified hallucination detection model. The authors collected various QA datasets from multiple domains, retrieving documents and artificially fabricating hallucinated answers that are critical but minimally different from the gold answers. They then trained an LLM to detect these hallucinations. The model is trained exclusively on QA scenarios and does not encompass scenarios such as summarization or generating answers based on structured data.

2.2 Hallucination Mitigation

Contrastive decoding has been found effective in mitigating hallucinations when generating context-based responses with LLMs, as discussed by Shi et al. (2023). This method amplifies the differences in the model's output distribution with and without context, encouraging the model to adhere strictly to the provided context and thus mitigating hallucination problems caused by neglect of the specified context or background knowledge.

Additionally, Tian et al. (2023) evaluates the factuality of open-ended text by measuring its consistency with an external knowledge base or using a large model's confidence scores. This method is used to automatically construct a pairwise chosen-reject dataset for Direct Preference Optimization (DPO) training. While Tian's work aims to enable language models to produce more factual answers, our research specifically focuses on enhancing LLM capabilities in RAG scenarios.

3 Dataset

RAGTruth (Niu et al., 2024) dataset is a substantial, word-level hallucination evaluation resource specifically tailored for the RAG scenario, encompassing several common tasks. We selected the RAGTruth dataset for our experiments because it is the largest available open-source dataset specifically designed for the RAG task. We adopt this dataset in both model training and system evaluation processes. The detailed statistic of the RAGTruth dataset is demonstrated in Appendix, Table 7.

Marco (Bajaj et al., 2018) is the Reading Comprehension Dataset consisting of real users' queries and web documents from Bing. We only used its question and web document pairs to enlarge our hallucination suppression training set.

WebGLM (Liu et al., 2023): The Web-enhanced question-answering dataset, was used in our system evaluation process.

XSum (Narayan et al., 2018): The BBC News Summary dataset, was used to extend our hallucination suppression training set by adapting part of its news articles.

4 Methodology

4.1 Hallucination Detection Model Training

In this section, we describe our approach to building a detection model that can identify hallucinations and provide clear, readable descriptions of these occurrences.

4.1.1 Training Data Construction With Selective Sampling

The RAGTruth dataset provides the spans of text identified as hallucinations but lacks detailed hallucination descriptions. In this subsection, GPT-4 Turbo is used to generate the descriptions to support the detection model training.

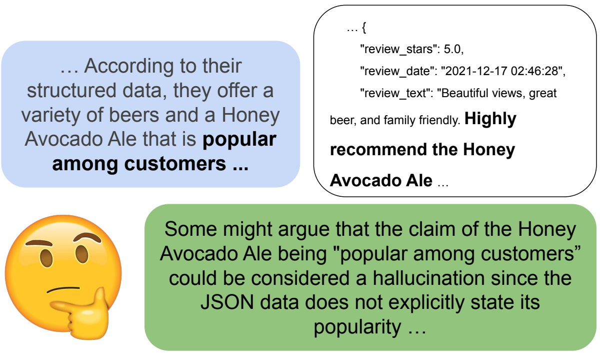
… According to their structured data, they offer a variety of beers and a Honey Avocado Ale that is **popular among customers …**
… {
"review_stars": 5.0,
"review_date": "2021-12-17 02:46:28",
"review_text": "Beautiful views, great beer, and family friendly. **Highly recommend the Honey Avocado Ale** …
Some might argue that the claim of the Honey Avocado Ale being "popular among customers" could be considered a hallucination since the JSON data does not explicitly state its popularity …

Figure 1: An Example of Defensive Advice: The LLM made a minor extension partially based on the provided references. Defensive advice highlights that the statement is not well supported.

Detection with description can be seen as a task of interpretable classification. We prompt GPT-4 with three main components:

- A binary label indicating whether a sentence contains hallucinations.
- A detailed explanation identifying where and why the hallucination occurs.
- Defensive advice that highlights sections of text perceived by GPT-4 as potentially ambiguous or indicative of minor hallucinations, accompanied by suggestions for improvement.

To clarify, an example of defensive advice is provided in Figure 1. We included this section because distinguishing clearly between hallucinated and non-hallucinated content is challenging. By incorporating defensive advice, LLMs can be guided to minimize boundary cases, thereby reducing the likelihood of hallucinations. As shown in Figure 1, the LLM made a minor extension based on the provided structured data, concluding the *Honey Avocado Ale is popular among customers*, based on the words of a single reviewer. While this might seem acceptable, it could be problematic and considered hallucinating under more stringent criteria.

Drawing inspiration from bootstrapping-style training methodologies (Zelikman et al., 2022) and rejection sampling utilized in the Llama2 development (Touvron et al., 2023), we implement a selective sampling strategy to ensure the quality and correctness of the generated data. Specifically, we assess the binary sentence label in the GPT-4 output. If the label is incorrect, we regenerate the data. This process is repeated for a specified number of attempts until the correct label is produced or we reach the attempt limit.

4.2 Two Stages Detection Model Training

Previous research has demonstrated that open-source large language models (LLMs) in their current form are not reliable for providing interpretations in hallucination detection tasks (Kamoi et al., 2024), and further fine-tuning is necessary.

To address this issue, we implemented a two-stage training strategy: In **stage one**, the model was trained exclusively to output the prediction label; In **stage two**, we adopted LoRA training to enable the model to provide interpretations based on the prediction label as input. The interpretations generated include descriptions of hallucinations as well as defensive advice.

During inference, the two models are employed in a cascaded sequence.

4.3 DPO Training for Hallucination Mitigation

We will outline the process of constructing the pairwise preference dataset, designed to train LLMs using DPO to generate responses with reduced hallucinatory content.

4.3.1 Answer Rewrite

In this section, we describe how we utilize GPT-4 Turbo to revise the original responses, which are then included in the DPO dataset as "chosen" examples.

For original responses identified as containing hallucinations, we collate the corresponding generated interpretations to guide GPT-4 Turbo in rewriting them to eliminate these hallucinations. For responses deemed as being good, we prompt GPT-4 Turbo with specific defensive advice and ensure that rewriting is confined within the specific sentence. This approach minimizes the risk of introducing new information that could lead to additional hallucinations. We also employ our detection model to verify the absence of hallucinations in the rewritten results. If hallucinations are detected, we repeat the rewriting process to ensure the dataset's integrity.

4.3.2 Overly Cautious Penalization (OCP)

We observed that models trained on our suppression dataset tend to produce less content, which,

Data Source	Original Samples	OCP Samples
XSum	**1840**	**514**
RAGTruth Train Split(Generated By Qwen)	**1590**	**465**
RAGTruth Train Split(Generated By GPT/Llama)	**9275**	**2832**
Extended Macro	**2465**	**740**

Table 1: Training Data Distributions

while reducing hallucinations, unfortunately, compromises the quality of the responses. To counteract this issue, we randomly delete one sentence from "chosen" responses in the dataset to generate additional "rejected" responses. This strategy effectively discourages models from merely shortening their responses to lower the hallucination rate, prompting them to keep a balance between maintaining content richness and minimizing hallucinations.

4.3.3 Data Source Extension

The RAGTruth dataset includes only 2,965 unique RAG tasks, which is relatively limited. Fortunately, our preference dataset generation is fully automated, enabling us to easily expand our training set by incorporating additional datasets. To better align with the real-world applications of the RAG system and our specific business needs, we have enriched our data with samples from the XSum dataset for summarization tasks, and from the unused portions of the Marco dataset for question answering.

We replaced the original answers in the XSum and Marco datasets with new answers generated by the selected LLM. Additionally, despite the multiple answers available in the RAGTruth dataset, we also used the selected LLM to regenerate answers. This approach ensures that the DPO "rejected" samples accurately reflect the LLM's output distribution. Notably, all the generated answers will undergo the previous process to acquire the corresponding "chosen" samples.

Finally, 19721 chosen/reject pairs are generated for DPO training. The detailed data distribution is shown in Table 1.

5 Experiments

5.1 Implementation Details

We utilized the Llama-3-8B Instruct (AI@Meta, 2024) version as the backbone for the detection model. We applied full parameters training with a learning rate of 1e-5 in the first stage, and 1e-4 for the second stage with LoRA which has set the hyper-parameters rank and alpha both to 32. Both stages were trained for two epochs, with a batch size of 8 on each device.

For the hallucination mitigation training, we selected the Qwen/Qwen1.5-4B-Chat (Qwen, 2024) as our base model due to its small model size and high inference speed, which align well with our business requirements. The training was conducted with a batch size of 2 on each device with 8 accumulation steps, a learning rate of 5e-6, and a relatively high beta value of 0.8 for a single epoch.

We utilize Adam optimizer (Kingma and Ba, 2014) with β_1 = 0.9, β_2 = 0.999, and employ a cosine scheduler for the learning rate with a 2% warm-up of the total steps to optimize the parameters. All the models are obtained from huggingface[1] and trained on 8 NVIDIA A100 80GB GPUs with fully sharded data parallel (Zhao et al., 2023).

The detailed prompt used for generating training data and evaluation can be found in the appendix, from Table 8 to Table 12.

5.2 Metrics and Baseline

In this paper, the RAGTruth test set is used to assess the efficacy of our DPO training in mitigating hallucinations.

To assess the model's suitability in a web-enhanced question-answering system, we also used a randomly sampled set of 1,000 training samples from WebGLM as the test set, as it more closely resembles our production scenario.

Specifically, we measured the efficacy of training from two perspectives: 1. **Hallucination rate** of LLM responses before and after the training; 2. The **response quality** of LLM before and after the training.

Regarding **hallucination rate**, for no bias, we used both our detection model and GPT-4 Turbo to detect hallucinatory responses, calculating the rate accordingly. To validate the automatic method's accuracy, we also conducted manual annotations on LLM's response on RAGTruth test sets.

Regarding **response quality**, we conducted pairwise comparisons on the model's responses before

[1] https://huggingface.co/

	QUESTION ANSWERING			DATA-TO-TEXT WRITING			SUMMARIZATION			OVERALL		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Prompt(GPT-4 Turbo)	43.7	**84.4**	57.6	84.4	88.1	86.2	68.9	**74.0**	**71.4**	70.3	**84.4**	76.7
RAGTruth	55.8	60.8	58.2	85.4	**91.0**	88.1	64.0	54.9	59.1	76.9	80.7	78.7
Ours	**76.5**	73.1	**74.8**	**92.9**	90.3	**91.6**	**77.7**	59.8	67.6	**87.3**	80.8	**83.9**

Table 2: Answer Level Hallucination Detection on RAGTruth Test Set: Compared with the best performance model introduced in RAGTruth which is a fine-tuned Llama-2-13B. Our detection model is fine-tuned on Llama-3-8B Instruct, which achieves the best performance. The P, R, and F respectively denote Precision, Recall, and F1 Score.

DATASET	METHOD	Detector	GPT-4 Turbo	Human	Average
RAGTruth Test Set	Qwen	36.9(-)	51.3(-)	34.4(-)	40.9(-)
	Qwen(Regenerate)	-	44.2(↓13.8%)	-	44.2(↓13.8%)
	RAG-HAT	22.7(↓**38.5%**)	41.3(↓**19.5%**)	25.7(↓**25.3%**)	29.9(↓**26.9%**)
WebGLM 1000	Qwen	21.3(-)	46.7(-)	-	34(-)
	Qwen(Regenerate)	-	38.8(↓17.0%)	-	38.8(↓17.0%)
	RAG-HAT	12.0(↓**43.7%**)	37.9(↓**19.0%**)	-	24.9(↓**26.8%**)

Table 3: Hallucination Rate: 1,000-Example WebGLM Set and RAGTruth Test Set (Total 450 Examples): Our detection model cannot fairly benchmark the hallucination rate of the regeneration approach since it serves as the trigger for regeneration.

and after training using GPT-4 Turbo. To mitigate bias, the order of responses presented in each prompt was randomized (Zheng et al., 2023). The comparisons are based on two criteria: 1.The accuracy of each answer reflects the details in the prompt; 2.The degree to which each response adheres to the guidelines provided in the prompt.

We also let the annotators to annotate with the same standard to verify the validity of GPT's results.

5.3 Hallucination Detection Performance

Our hallucination detection model, which is fine-tuned on Llama-3-8B Instruct, archives a significant improvement in classification performance for all three major tasks compared with the baseline described in the RAGTruth paper, which is fine-tuned on Llama-2-13B base model. Specifically, our model demonstrates an approximate 7.2% overall improvement in f1-score and 17% in precision, as detailed in Table 2.

The superior performance of our detection model has led us to include it as one of the metrics for measuring the RAG-HAT's hallucination suppression performance, alongside GPT-4 and human review.

5.4 Hallucination Suppression Performance

As shown in Table 3, the metrics from different sources all indicate that RAG-HAT significantly decreases the hallucination rate on both the RAGTruth and WebGLM datasets. Specifically, there is, on average, a 26.9% drop in the hallucination rate in the RAGTruth dataset. Additionally, for the WebGLM dataset, there was an average decrease of 26.8% in the hallucination rate.

We also tested the naive regeneration strategy, which involves detecting if the generated answer contains hallucinations. If hallucinations are found, we regenerate the answer, allowing only one regeneration attempt.

Based on GPT-4 Turbo, the average hallucination rate from RAG-HAT is about 4.4% lower than that of the regeneration approach. It is important to note that our detection model cannot be used to fairly benchmark the hallucination rate of the regeneration approach, as it already serves as the trigger for regeneration. Moreover, the regeneration approach not only exhibits inferior performance but also doubles the generation time and does not fully support streaming of the model's responses. This streaming capability is essential for many real-world products, including ours, to minimize user waiting time.

5.5 Human Annotations

Manually reviewing the hallucination rate for all experiments is expensive, given the large size of the dataset used for evaluation and the complexity of the annotation tasks. However, as demonstrated in Table 3, the results of our human annotations on the RAGTruth test set closely align with the metrics from automatic methods. This close alignment underscores the reliability of our automatically derived metrics and the effectiveness of RAG-HAT

	QUESTION ANSWERING			DATA-TO-TEXT WRITING			SUMMARIZATION			OVERALL		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
GPT-4 Turbo (Describe and Predict)	36.4	**73.4**	48.7	58.2	74.2	65.3	46.8	**62.5**	53.5	49.4	**72.0**	58.6
Finetuned (Describe and Predict)	57.5	58.8	58.2	72.2	71.9	72.0	64.7	43.0	51.6	67.6	64.1	65.8
Finetuned (Describe and Predict, w DPO)	67.6	57.4	62.1	74.0	74.6	74.3	69.1	41.0	51.5	72.1	65.0	68.4
Finetuned (Predict only)	**69.8**	63.5	**66.5**	**79.1**	**80.4**	**79.7**	**71.2**	49.2	**58.2**	**76.0**	71.3	**73.6**

Table 4: Sentence Level Hallucination Detection Performance on RAGTruth Test Set. The P, R, and F respectively denote Precision, Recall, and F1 Score.

DATASET	METHOD	GPT-4 Turbo	Human
RAGTruth Dataset	Qwen	41.1	33.2
	RAG-HAT	**57.3**	**40.8**
WebGLM 1000	Qwen	39.5	-
	RAG-HAT	**58.5**	-

Table 5: Answer Quality Win Rates: 1,000-Example WebGLM Set and RAGTruth Test Set

PAIRED METHOD	WIN RATE (GPT-4 Turbo)
RAG-HAT (full) :: (w/o defensive, w/ OCP)	**51.5**
RAG-HAT (full) :: (w/o defensive, w/o OCP)	**54.1**

Table 6: Impact of Training Dataset Composition on Answer Quality: Pairwise Comparison

in reducing hallucinations.

5.6 Answer Quality

In response to concerns that DPO training might lead the model to sacrifice answer quality and richness in order to reduce hallucinations, we conducted evaluations to assess the quality of the generated responses.

GPT-4 is prompted to compare response quality in pairs. The evaluations indicate that the DPO-trained model delivers better answer quality compared to the original model, as shown in Table 5. Specifically, it achieved a 57% win rate on the RAGTruth test set, compared to the original model's 41% win rate. On the WebGLM dataset, the trained model achieved a 59% win rate, outperforming the original model's 40% win rate.

6 Analysis

6.1 Impact of Defensive Advice and Overly Cautious Penalization (OCP)

We conducted a set of ablation experiments on WebGLM to demonstrate the effectiveness of defensive advice as described in Section 4.1.1, as well as the data augmentation by random deletion of one sentence from "chosen" examples as described in Section 4.3.2. As is illustrated in Table 6, the experiments of both data generation strategies are beneficial in improving the answer quality.

The model trained on the full dataset achieved a 51.5% win rate, outperforming the model trained without defensive advice, and a 54.1% win rate trained without both. Notably, the win rate increased by 2.6% upon the removal of OCP, underscoring the efficacy of our penalization strategy.

6.2 Effectiveness of the Two Stage Detection Model Training

To substantiate the necessity of adopting the two-stage training approach for our detection model—where the model outputs prediction labels directly rather than engaging in reasoning using Chain of Thought (CoT) (Wei et al., 2023) style—we compared fine-tuning results on the Llama-3-8B using both training strategies.

As shown in Table 4, training the model to generate hallucination descriptions and prediction together consistently yielded suboptimal results compared to training the model to output the prediction label only. Even when we incorporated DPO training—sampling outputs from a previously supervised fine-tuned model to build a preference dataset based on prediction correctness and then conducting subsequent DPO training—the final classification performance remained suboptimal.

We speculate this is due to the training methodology of auto-regressive models. If hallucination descriptions and labels are generated together, the optimization of the prediction label might be diluted by the other tokens from the hallucination description, leading the model to converge to a suboptimal point for label predictions.

7 Industry Application

As a local information provider, NewsBreak is actively exploring various applications of Retrieval-Augmented Generation (RAG) systems within our business model. Our primary focus is on using the RAG system to gather fragmented local data and leveraging large language models (LLMs) to organize this information into coherent formats, such as Question-Answering systems or highly informative resources.

We are currently experimenting with the integration of RAG-HAT into our RAG system to enhance the accuracy and relevance of the local information we provide. Users can access a wide range of information through our platform, particularly about local entities (e.g., restaurants, auto shops), local safety (e.g., crime reports), and community events. Thus, we incorporate substantial amounts of local merchant data, news articles, and other sources into our training datasets.

Techniques like RAG-HAT significantly reduce the risk of unintentionally disseminating misinformation, which is critical for protecting our reputation. Additionally, they enable our product managers to plan more advanced RAG applications with confidence, mitigating potential legal and reputational risks associated with hallucinations.

8 Conclusion

In this work, we introduce a hallucination-aware tuning pipeline RAG-HAT, which contains three parts: detection, rewriting and mitigation. The detection component identifies hallucinations with human-readable interpretations. The rewriting component allows us to automatically generate a preference dataset, enabling the use of DPO to train models to hallucinate less. Specialized data augmentation techniques are designed to reduce hallucinations without compromising the model's answer quality. Benchmarks demonstrated that RAG-HAT significantly reduced the hallucination rate while enhancing answer quality simultaneously.

Limitions

Due to limited computational resources, we did not test our method on larger LLMs, such as Llama3-70B. Furthermore, we did not evaluate our model in domains requiring expert knowledge, such as finance and medicine, due to the lack of annotators with specific domain expertise.

References

AI@Meta. 2024. Llama 3 model card.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. Ms marco: A human generated machine reading comprehension dataset.

Zhaorun Chen, Zhuokai Zhao, Hongyin Luo, Huaxiu Yao, Bo Li, and Jiawei Zhou. 2024. Halc: Object hallucination reduction via adaptive focal-contrast decoding.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2023. Chain-of-verification reduces hallucination in large language models.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. 2022. Training compute-optimal large language models.

Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. 2023. Challenges and applications of large language models.

Ryo Kamoi, Sarkar Snigdha Sarathi Das, Renze Lou, Jihyun Janice Ahn, Yilun Zhao, Xiaoxin Lu, Nan Zhang, Yusen Zhang, Ranran Haoran Zhang, Sujeeth Reddy Vummanthala, Salika Dave, Shaobo Qin, Arman Cohan, Wenpeng Yin, and Rui Zhang. 2024. Evaluating llms at detecting errors in llm responses.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *CoRR*, abs/2001.08361.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks.

Xiao Liu, Hanyu Lai, Hao Yu, Yifan Xu, Aohan Zeng, Zhengxiao Du, Peng Zhang, Yuxiao Dong, and Jie Tang. 2023. Webglm: Towards an efficient web-enhanced question answering system with human preferences.

Alejandro Lozano, Scott L Fleming, Chia-Chun Chiang, and Nigam Shah. 2023. Clinfo.ai: An open-source retrieval-augmented large language model system for answering medical questions using scientific literature.

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

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! Topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium.

Cheng Niu, Yuanhao Wu, Juno Zhu, Siliang Xu, Kashun Shum, Randy Zhong, Juntong Song, and Tong Zhang. 2024. Ragtruth: A hallucination corpus for developing trustworthy retrieval-augmented language models.

OpenAI. 2024. Chatgpt (gpt-4 turbo). Large language model.

Qwen. 2024. Introducing qwen1.5.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model.

Selvan Sunitha Ravi, Bartosz Mielczarek, Anand Kannappan, Douwe Kiela, and Rebecca Qian. 2024. Lynx: An open source hallucination evaluation model.

Jon Saad-Falcon, Omar Khattab, Christopher Potts, and Matei Zaharia. 2024. Ares: An automated evaluation framework for retrieval-augmented generation systems.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Scott Wen tau Yih. 2023. Trusting your evidence: Hallucinate less with context-aware decoding.

Kurt Shuster, Jack Urbanek, Emily Dinan, Arthur Szlam, and Jason Weston. 2021. Dialogue in the wild: Learning from a deployed role-playing game with humans and bots. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 611–624, Online. Association for Computational Linguistics.

Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher D. Manning, and Chelsea Finn. 2023. Finetuning language models for factuality.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and finetuned chat models.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. 2022. Star: Bootstrapping reasoning with reasoning.

Boyu Zhang, Hongyang Yang, Tianyu Zhou, Muhammad Ali Babar, and Xiao-Yang Liu. 2023. Enhancing financial sentiment analysis via retrieval augmented large language models. In *Proceedings of the Fourth ACM International Conference on AI in Finance*, ICAIF '23, page 349–356, New York, NY, USA. Association for Computing Machinery.

Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, et al. 2023. Pytorch fsdp: experiences on scaling fully sharded data parallel. *arXiv preprint arXiv:2304.11277*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena.

A Appendix

Task	# Inst.	# Resp.	HALLUCINATION		
			# Resp.	% Resp.	# Span
Question Answering	989	5934	1724	29.1%	2927
Data-to-text Writing	1033	6198	4254	68.6%	9290
Summarization (CNN/DM)	628	3768	1165	30.9%	1474
Summarization (Recent News)	315	1890	521	27.6%	598
Overall	2965	17790	7664	43.1%	14289

Table 7: The basic statistics of RAGTruth. Here "Resp." stands for "Answer".

Given an instruction:
{instruction}
Consider these responses:
A: {answer_a}
B: {answer_b}
In this context, "better" refers to:
- The accuracy of each response in reflecting the details of the instruction.
- The extent to which each response adheres to the guidelines provided in the instruction.
Based on these criteria, assess which response is better.
Provide your analysis in this format:
Analysis: [Your Analysis Here]
Better Response: A or B or TIE

Table 8: Prompt for Evaluating Answer Quality

Given a prompt with reference:
{prompt}
and a sentence:
"{sentence}"
which is from the generated answer:
"{full_answer}"
Please find whether there are hallucinations in the generated sentence (not the whole answer)
Hallucinations Definition:
1. conflict: instances where the generative content presents direct contradiction or opposition to the original input;
2. baseless info: instances where the generated content includes information which is not substantiated by or inferred from the original input.
You response should be a binary label, where:
True means there are hallucinations in the generated sentence.
False means there are no hallucinations in the generated sentence.
Now please answer in the following format exactly:
Pred: True or False

Table 9: Prompt for Sentence Level Hallucination Detection without Description. We use this prompt to evaluate the hallucination rate in LLM's response.

Given a prompt with reference:

{prompt}

and a sentence:

"{sentence}"

which is from the generated answer:

"{full_answer}"

Please find whether there are hallucinations in the generated sentence (not the whole answer)

Hallucinations Definition:

1. conflict: instances where the generative content presents direct contradiction or opposition to the original input;

2. baseless info: instances where the generated content includes information which is not substantiated by or inferred from the original input.

You response should be in two parts:

1. Analysis: This part should reflect your thinking process. Provide the explanation for your final conclusion.

2. Defensive Advice: If you are confident that there are no hallucinations, which part of it might others mistakenly believe to be hallucinated, and how would you respond to their challenges? Conversely, if others think the information is accurate but you believe it contains hallucinations, which part would you challenge, and how would you argue your case?

3. Final Conclusion: Your final conclusion, it should be a binary label: True or False.

Now please answer in the following format exactly:

Analysis(1 paragraph): [NO NEW LINE]...

Defensive Advice(1 paragraph): [NO NEW LINE]...

Final Conclusion: [NO NEW LINE]...

Table 10: Prompt for Sentence Level Hallucination Detection with Description. We use this prompt to synthesize training data for detection model.

Given an answer produced by an LLM (Large Language Model) according to the following prompt:

{prompt}

Here is the LLM-generated answer:

"{full_answer}"

A report identifies these hallucinations:

{hallucination_reports}

Please revise the LLM's answer with minimal modifications necessary to:

1. Correct any hallucinations identified in the report. You may rewrite parts of the answer to ensure coherence.

Note, if you think no modifications need to be made, just repeat the given LLM-generated answer.

Format your response as follows:

Modifications plan: [NO NEW LINE, ONE PARAGRAPH]

Revised answer:

Table 11: Rewrite Prompt For LLM Response Classified as Hallucinated

```
Given an answer produced by an LLM (Large Language Model) according to the following prompt:
{prompt}
Here is the LLM-generated answer:
"{full_answer}"
A report outlines these concerns and potential confusion points:
{defensive_advice}
Please revise the LLM’s answer with minimal modifications necessary to:
1. Enhance the rigor of the answer based on Report. Focus only on sentence-level modifications without adding new sentences or new information.
Note:
You don’t need to solve all the concerns or confusion points listed in the report, pick the sentences you think are necessary to revise.
If you think no modifications need to be made, just repeat the given LLM-generated answer.
Format your response as follows:
Modifications plan: [NO NEW LINE] For sentence bx, ...; For sentence bx, ...; ...
Sentences you need to modify: ["b1", ..., "bn"] or [](empty_list)
Revised answer:
```

Table 12: Rewrite Prompt For LLM Response Classified as Not Hallucinating