PROBE: PROcess-Based BEnchmark for Hallucination Detection

Anonymous ACL submission

Abstract

Hallucination detection remains a significant challenge for large language models. Existing agentic applications rely on LLMs to self-assess the factuality of their outputs using single-step "LLM-as-a-judge" prompts. However, even when equipped with ground truth information, current LLMs still fall short in detecting hallucinations, and this one-shot evaluation offers neither the transparency nor the granularity needed to diagnose where and why the detection fails. To address this gap, we introduce PROBE (**Pro**cess-based **Be**nchmark for Hallucination Detection), a comprehensive benchmark that breaks down hallucination detection into four critical steps: claim decomposition, evidence finding, evidence evaluation, and hallucination localization, and evaluates each step individually. PROBE consists of 12,000 test cases across three task types—summarization, question answering, and style transfer. Critically, we demonstrate that when hallucination detection is treated as a multi-step process, all models achieve considerably better performance. Through extensive evaluation, we show that current LLMs struggle chiefly with evidence finding, and that finetuning on our released training data substantially improves performance on this step. PROBE represents a significant step toward more transparent, diagnosable, and robust hallucination detection systems.

1 Introduction

In recent years, language models have made remarkable progress across many language modeling tasks, including text generation (Li et al., 2024), text summarization (Zhang et al., 2024), and question answering (Allemang and Sequeda, 2024). However, one of the key challenges in deploying LMs to perform these tasks in real-world applications is their tendency to hallucinate facts. This typically entails the model generating content that is not grounded in factual information or provided

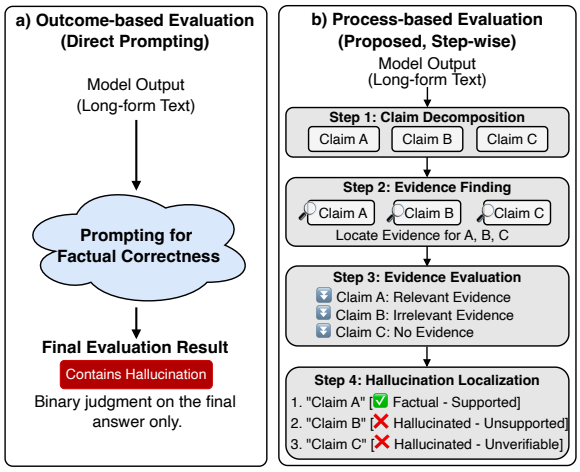


Figure 1: Comparison of hallucination detection approaches. Our proposed step-wise evaluation provides fine-grained insights for factual hallucination detection.

context (Niu et al., 2023) – behavior that is deeply rooted in the nature of LM generation: even when given authoritative source documents, the model is not inherently constrained to produce outputs that strictly follow the facts in its input. As a result, detecting factual hallucinations has become a critical problem for ensuring the safety and trustworthiness of LM-based systems.

In terms of factual hallucination detection, various agentic applications have emerged in which LLMs are prompted to self-reflect or self-judge their own outputs to improve factuality (Hu et al., 2024; Song et al., 2024; Mishra et al., 2024). In these applications, the LLM is invoked as a judge to identify factual inconsistencies between its generated content and the provided ground-truth information. Correspondingly, a range of evaluation benchmarks has been developed to measure LLMs' hallucination detection capabilities at both the word/span level and the sample level. For example, RAGTruth (Niu et al., 2023) offers a corpus tailored for detecting word-level hallucinations in RAG settings, whereas HaluBench (Ravi et al., 2024) evaluates hallucination recognition at the

sample level using binary labels to indicate whether a model's output contains hallucinations.

Despite growing interest in this direction, existing approaches suffer from a fundamental limitation. Current benchmarks focus primarily on outcome-based evaluation, assessing only whether the final answer is factually correct yet providing no insight into why this detection process succeeds or fails. This limitation becomes more pronounced in long-context scenarios, where direct prompting is insufficient for reliably identifying hallucinations (Wei et al., 2024). In general, existing evaluation methods treat hallucination detection as a monolithic task, overlooking the distinct cognitive capabilities required for effective fact-checking: the ability to decompose complex claims, find relevant evidence, evaluate evidence relevance, and precisely localize unsupported claims.

Motivated by this, we introduce **PROBE** (**PRO**cess-Based **BE**nchmark for Hallucination Detection), a comprehensive benchmark for evaluating hallucination detection capabilities through a multi-step assessment process. PROBE comprises 12,000 test cases across three diverse task types, including summarization, question answering, and style transfer, with 118,628 annotated claims with detailed annotations at both claim and evidence levels. Our benchmark decomposes hallucination detection into four critical steps: (1) *claim decomposition*, evaluating the model's ability to break down complex statements into verifiable atomic claims; (2) *evidence finding*, assessing whether models can locate truthful and relevant evidence; (3) *evidence evaluation*, measuring the capability to differentiate relevant from irrelevant evidence; and (4) *hallucination localization*, determining the exact span of hallucinated content.

Our key contributions are:

1. We introduce PROBE, the first large-scale process-based benchmark suite for fine-grained hallucination detection across multiple tasks.

2. We introduce a comprehensive evaluation framework that breaks down hallucination detection into four distinct capabilities, enabling more precise diagnosis of model strengths and weaknesses.

3. We build a high-quality dataset spanning three categories of real-world LLM tasks, featuring annotated claim-evidence pairs and precise-annotated hallucinations.

4. We conduct extensive experiments evaluating current LLMs on each step of the process, identifying evidence finding as the primary performance bottleneck and establishing detailed step-wise baselines.

5. We show that process-based evaluation yields more actionable insights than one-shot LLM-as-a-judge prompting and that step-wise supervision enables more effective model improvement. In particular, we demonstrate that finetuning on our released training data substantially enhances model performance on the hardest steps.

2 Motivation

This paper investigates a new paradigm for evaluating the hallucination detection capabilities of large language models. Specifically, our work is motivated by three key observations:

- **Granular Diagnosis Matters**: Decomposing hallucination detection into well-defined stages enables fine-grained diagnosis of model failures. This breakdown makes it possible to pinpoint where a model goes wrong, such as inaccurate claim extraction, weak evidence retrieval, or incorrect evidence evaluation, providing actionable insights for improving accuracy.

- **Process Evaluation Enhances Detection Accuracy**: A decomposed hallucination detection process yields substantially higher detection accuracy than one-shot outcome-based evaluation, as shown in Figure 4b. By explicitly modeling the reasoning pipeline, models achieve more faithful and robust assessments of factual consistency.

- **Stronger Learning through Process Supervision**: Outcome-based supervision offers only a coarse and often ambiguous training signal. In contrast, granular step-level annotations supply rich process supervision that guides models toward mastering each component of the hallucination detection workflow, leading to more effective learning and better generalization.

3 PROBE Construction

The term hallucination in the context of LLMs is commonly used in two distinct ways. It may refer to nonfactual content, where model outputs are not

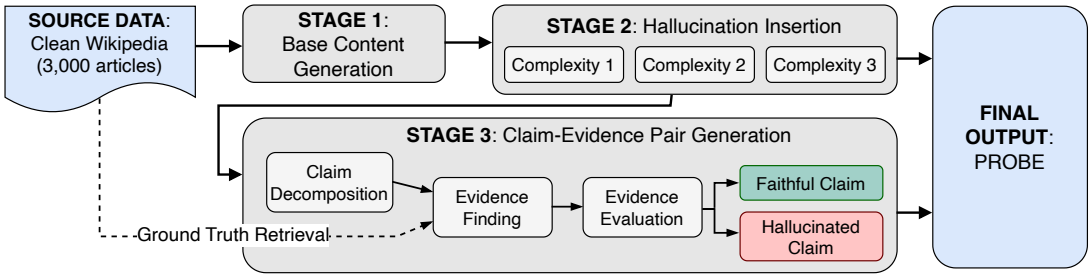


Figure 2: PROBE generation pipeline. (1) Base content generation, where responses are generated using LLMs with natural prompts; (2) Hallucination insertion, in which three levels of hallucinations are synthetically injected into the base content while preserving linguistic consistency; and (3) Claim-evidence pair generation, where frontier models are leveraged to find evidence from the ground truth and vote on evidences.

grounded in real-world knowledge, or to unfaithful or inconsistent content, where the generated text fails to adhere to the provided input. PROBE focuses on the latter setting in which an LLM must complete a task such as summarization, question answering, or style transfer based on a given passage or reference, as in Retrieval-Augmented Generation (RAG) (Lewis et al., 2020). Under this definition, a statement is considered unfaithful when it lacks support from the source material.

Figure 2 illustrates the pipeline used to generate and annotate our dataset, which includes three types of poisoned content: summaries, question–answer pairs, and style-transferred texts. Across all task types, the generation process follows a consistent pattern involving baseline content generation, hallucination injection, and claim-evidence pair generation.

3.1 Task Definition and Content Generation

We consider three widely used generation tasks under RAG settings: summarization, question answering, and style transfer. We randomly sample 1,000 articles for each task from the training split of the Clean Wikipedia dataset (Foly et al., 2025), resulting in 3,000 source documents. For summarization, LLMs are prompted to produce a concise summary of each source document. For question answering, we first prompt the language model to generate a question about the provided text that can be answered with a small number (2-4) of concrete facts. The model is then asked to answer each generated question using only the retrieved passages corresponding to the article. For style transfer, the goal is to rewrite the article in a different style while preserving its meaning. We consider four target styles: blog post, lecture notes, FAQ, and textbook. For each article, one style is selected at random, and the LLM is prompted to rewrite the original Wikipedia text accordingly, yielding approximately 25% of samples in each style category. In this step, as well as in the subsequent hallucination insertion step, we use GPT-OSS-120B (OpenAI, 2025) for generation, balancing quality with computational efficiency. All prompts used to generate outputs for these tasks are provided in the Appendix B.

3.2 Hallucination Insertion

To precisely control the location of hallucinations, we construct unfaithful hallucinations by synthetically injecting poisoned semantics, i.e., statements that appear plausible and factual but are in fact unsupported by the source context. We define a semantic perturbation as a subsequence that remains semantically coherent within the generated answer yet lacks any retrievable supporting evidence in the groundedness setting, thereby constituting a claim-level unfaithful hallucination.

PROBE comprises three blocks of samples that enable evaluation for summarization, question answering, and style transfer over Wikipedia articles. Each block contains 1,000 hallucination-free baseline samples and 3,000 samples with synthetically injected plausible factual hallucinations of three complexity levels. Complexity 1 samples contain one factual hallucination, usually as a direct mention incorporated in the otherwise accurate summary. Complexity 2 samples contain two factual hallucinations, either as direct mentions or as fabricated factual statements presented as supporting or logically extending true claims. Complexity 3 samples include three factual hallucinations introduced in the same manners as Complexity 1 and 2, with the additional property that genuinely true statements may be derived through logical inference from the hallucinated claims. Annotations describing spans corresponding to hallucinated claims are provided in the dataset.

Task	# Instance	# Claims	# Claims		Response Length		Hallucination Length		
			Hallucinated	Truth	Mean	Max	Mean	Max	% Resp.
Summarization	3,000	41,558	9,908	31,560	1,697	5,331	223	2,832	15.13%
Question Answering	3,000	12,018	5,489	6,529	167	1,305	135	562	47.42%
Style Transfer	3,000	65,052	10,216	54,836	2,060	8,437	332	1,063	16.93%
Overall	**12,000**	**118,628**	**25,613**	**92,925**	**1,308**	**8,437**	**230**	**2,832**	29.96%

Table 1: Claim-level statistics of PROBE. We report hallucinated claims (from injected segments) and truthful claims supported by strong multi-model consensus. Here "Resp." stands for "Response" and % Resp. denotes the proportion of hallucination regarding the whole response.

In general, PROBE consists of three subsets designed to evaluate factuality in summarization, question answering, and style transfer over Wikipedia articles. Each subset contains 1,000 hallucination-free baseline samples and 3,000 samples with synthetically injected plausible factual hallucinations, evenly distributed across three complexity levels.

3.3 Claim-Evidence Pair Generation

Claim Decomposition. To construct high-quality claim-evidence pairs, we first decompose each sample into a set of atomic claims. We define a *claim* as the smallest unit of information that can be independently evaluated against a given context. This formulation follows prior work on hallucination and factuality evaluation, which similarly decomposes model outputs into claims for fine-grained assessment (Min et al., 2023; Akbar et al., 2024; Hu et al., 2024). Each claim is associated with the specific text segment in the ground truth source that expresses it. For this step, we invoke a language model (Llama-3.1-70B (AI@Meta, 2024)) in parallel to perform claim decomposition. Claims originating from synthetically injected hallucination segments are directly labeled as hallucinated, since no supporting evidence exists in the source document. For claims derived from baseline content, we then perform the following evidence finding step to retrieve evidence from original Wikipedia sources.

Evidence Finding. In this step, we retrieve candidate evidence spans from the original Wikipedia articles using four frontier LLMs, including Llama-3.1-70B (AI@Meta, 2024), GPT-4o-mini (Hurst et al., 2024), Mixtral-8×22B (Mistral.ai, 2024), and Claude-Sonnet-4.5 (Anthropic, 2025). We take the union of all retrieved evidence spans as the candidate evidence set. Here we observe that LLaMA-3.1-70B outperforms LLaMA-3.3-70B on our task, which we attribute to its stronger alignment with explicit task instructions. In contrast, LLaMA-3.3-70B appears to favor broader generalization, which does not translate as effectively to our instruction-sensitive setting. Accordingly, we adopt LLama-3.1-70B throughout this paper.

Evidence Evaluation. Then, each evidence candidate is independently evaluated by the same four models to determine whether it supports the corresponding claim, producing a binary judgment. A piece of evidence is accepted as supporting if at least 3 out of 4 models agree, i.e., >=0.75 consensus. The resulting verified evidences form the claim's supporting evidence set. Claims with one or more verified evidence spans are labeled as truth claims, and claims without any supporting evidence are labeled as hallucinated claims. This procedure yields an accurate and robust claim–evidence dataset, with faithful claims grounded by validated evidence and hallucinated claims precisely identified through controlled perturbation.

3.4 Benchmark Statistics

We presented detailed statistics of PROBE in Table 1. Compared to existing datasets for hallucination detection, PROBE is considerably large in scale. In total, the benchmark contains 12,000 generated responses and 118,628 claims extracted from 9,000 samples across summarization, question answering, and style transfer tasks. Specifically, we focus on two claim categories: hallucinated claims, which originate from synthetically injected segments, and high-confidence grounded claims, which are supported by evidence validated with strong multi-model consensus. Overall, PROBE includes 25,613 hallucinated claims (21.7%) and 92,925 high-confidence grounded claims (78.3%).

The dataset exhibits substantial diversity in response length and hallucination characteristics across tasks. Summarization and style transfer produce substantially longer responses (mean lengths of 1,697 and 2,060 characters, respectively), resulting in relatively low hallucination ratios at the

response level (15.13% and 16.93%). In contrast, question answering responses are shorter on average but contain a much higher proportion of hallucinated content (47.42%), reflecting the brittleness of factual precision in concise answers. Overall, hallucinated spans average 230 characters in length and account for 29.96% of responses, underscoring the necessity of fine-grained, claim-level evaluation rather than coarse response-level judgments.

4 Evaluation

4.1 Setup

Hallucination Detection Algorithms. We evaluate both direct prompting baselines and our proposed process-based hallucination detection method. For direct prompting, large language models are instructed in a single step to identify hallucinated content within a generated response. In contrast, our process-based approach decomposes hallucination detection into four sub-steps, including claim decomposition, evidence finding, evidence evaluation and hallucination localization. All hallucination detection prompts are manually designed and applied uniformly across frontier models, including Llama-3.1-70B (AI@Meta, 2024), GPT-4o-mini (Hurst et al., 2024), Mixtral-8×22B (Mistral.ai, 2024), and Claude-Sonnet-4.5 (Anthropic, 2025). Detailed prompt templates and implementation details are provided in the Appendix.

Data Split. All detection algorithms are evaluated on the same evaluation set of PROBE. This evaluation set is constructed by randomly sampling 100 instances from each task type: summarization, question answering, and style transfer. The remaining data is used exclusively for fine-tuning language models to enhance its capability in evidence finding and evidence evaluation, ensuring no overlap between training and evaluation samples.

Evaluation Metrics. Unlike prior work that evaluates hallucination detection at the response level or word level, we adopt a fine-grained claim-wise evaluation protocol. Specifically, we compute claim-level matching between the predicted hallucination claims and the ground-truth annotated hallucination claims. Based on this overlap, we report precision, recall, and F1 score at the character level, which more accurately reflects a model's ability to localize hallucinated content within long and complex responses. Unlike prior work that evaluates hallucination detection at the response or word level, we adopt a fine-grained, claim-wise evaluation protocol. Specifically, we match predicted hallucinated claims against ground-truth annotated hallucination claims at the claim level. Based on the resulting matchs, we compute precision, recall, and F1, which more accurately measure a model's ability to localize hallucinated content within long and complex responses.

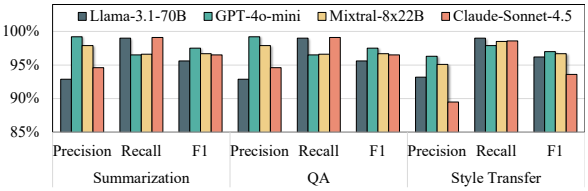


Figure 3: The claim decomposition performance of LLMs on PROBE. Frontier LLMs perform strongly on this task, achieving high recall and laying a solid foundation for step-wise evaluation.

4.2 Step-Wise Results of PROBE

Claim Decomposition. The first step in our evaluation pipeline assesses the quality of claim decomposition, which determines whether a model can correctly identify the set of claims present in a generated response. We directly prompt the LLMs to decompose each response into a set of atomic claims. The extracted claims are then matched against ground-truth claims annotated in PROBE.

As shown in Figure 3, both precision and recall are consistently high across tasks, especifically for the recall (consistently > 95%). This indicates that modern LLMs are effective at identifying atomic claims in generated text, and that claim decomposition itself is a relatively easy subtask compared to later stages such as evidence finding and evaluation. The strong performance on claim decomposition thus provides a reliable foundation for downstream steps in the process-based evaluation pipeline, enabling more meaningful analysis of evidence finding and evaluation capabilities.

Evidence Finding. The second evaluation stage examines the ability of LLMs to find supporting evidences for a given claim. For each claim that is correctly matched in the claim decomposition step, the model is prompted to identify evidences from the ground-truth Wikipedia article. We evaluate evidence finding using two complementary metrics. *Partial Match* measures whether the model retrieves at least one correct supporting evidence span for a claim, while *Complete Match* requires the model to retrieve all annotated supporting evidences associated with that claim.

As shown in Table 2, frontier models achieve

Table 2: The evidence finding (EF) and evidence evaluation (EE) accuracy of LLMs on PROBE. "Partial" denotes retrieving at least one correct supporting evidence, while "Complete" denotes retrieving all required supporting evidence. SFT Llama-3.1-8B refers to the fine-tuned Llama-3.1-8B model. Overall, current frontier LLMs exhibit sub-optimal performance on both EF and EE, whereas the fine-tuned model consistently performs better.

Models	Partial Evidence Finding			Complete Evidence Finding			Evidence Evaluation		
	Summarization	QA	Style Transfer	Summarization	QA	Style Transfer	Summarization	QA	Style Transfer
Llama-3.1-70B	82.7	**88.6**	82.9	69.2	49.7	69.0	77.5	72.3	81.9
GPT-4o-mini	78.7	80.1	79.2	64.2	41.6	66.5	79.9	70.6	79.2
Mixtral-8x22B	77.0	82.0	77.9	63.4	43.5	66.0	80.3	71.6	77.9
Claude-Sonnet-4.5	81.6	86.4	81.6	68.1	49.1	69.1	81.6	69.8	81.6
SFT Llama-3.1-8B	**83.2**	87.4	**84.1**	**70.3**	**55.1**	**71.4**	**82.9**	**80.7**	**83.1**

relatively high partial match scores (about 80%), indicating that LLMs are often able to locate at least one relevant evidence. However, complete match performance remains substantially lower, even below 50% for QA task, revealing a significant gap in the LLM's ability to comprehensively retrieve all necessary evidence. Moreover, we observe that QA achieves the highest partial evidence finding accuracy but the lowest complete evidence finding accuracy. This is because claims extracted from QA tend to have more supporting evidence than those from summarization or style transfer, owing to the higher fact density in QA. These results highlight evidence finding as a major bottleneck in the hallucination detection process. While models can often identify some relevant context, they struggle to exhaustively recover all supporting evidence, which limits reliable downstream evidence evaluation and hallucination localization.

Evidence Evaluation. The third stage evaluates the ability of LLMs to judge whether candidate evidence supports a given claim using a binary voting mechanism. To isolate evidence evaluation from retrieval quality, we reuse the candidate evidences obtained in the EF step and perform no additional retrieval. For *truth claims*, a prediction is considered correct if the evaluator votes *support* for its truth evidences. For *hallucinated claims*, a prediction is considered correct if the evaluator votes *non-support* for possibly retrieved evidences. As shown in Table 2, the overall accuracy of evidence evaluation is sub-optimal. Even the advanced Claude-Sonnet-4.5 tends to misclassify evidences, achieving a low accuracy of only 69.8%. This indicates that, even when candidate evidence is provided, LLMs struggle to reliably assess evidence sufficiency and relevance, making evidence evaluation a non-trivial bottleneck in the hallucination detection pipeline.

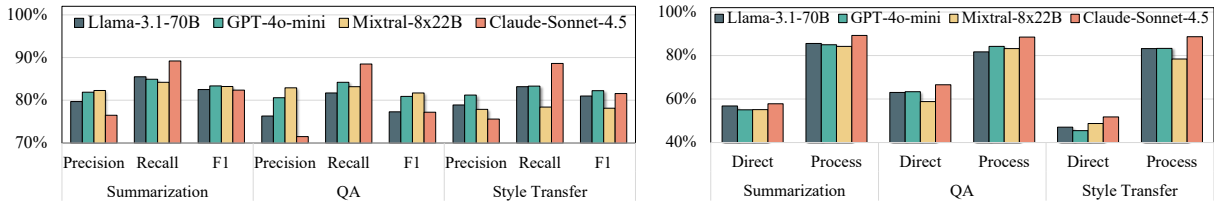
Hallucination Localization. The final stage evaluates the model's ability to localize hallucinated content in the generated response. This step reuses the hallucination predictions from the evidence evaluation stage, where claims without supporting evidence are classified as hallucinated. To assess localization accuracy, we match predicted hallucinated claims to ground-truth hallucination annotations, reusing the semantic claim matching results in the first step. Therefore, no additional model inference is performed in this step.

We report precision, recall, and F1 score for hallucination localization. As shown in Figure 4a, all LLMs achieve high recall (above 80%), indicating that step-wise reasoning effectively captures most hallucinated content, but their precision remains relatively low, reflecting limitations in the evidence evaluation step. Notably, the Claude-Sonnet-4.5 model achieves the highest recall (88.8%) but the lowest precision (74.5%), due to its conservative voting behavior in the evidence evaluation step. Overall, GPT-4o-mini demonstrates the best balance, achieving the highest average F1 score.

4.3 Hallucination Detection Performance

We compare hallucination detection performance between direct prompting and our process-based evaluation method. We focus primarily on recall, as it directly reflects a model's ability to identify hallucinated content within generated responses.

As shown in Figure 4b, direct prompting yields very low recall (below 40%), indicating that LLMs struggle to reliably detect hallucinations in long-form generations when asked to do so in a single step. In contrast, the process-based method consistently achieves substantially higher recall, generally exceeding 80% and reaching up to 90% in the best case. These results demonstrate that decomposing hallucination detection into explicit reasoning steps is critical for effectively identifying hallucinations in long and complex model outputs, and strongly motivate the use of process-based evaluation over one-shot LM-as-a-judge prompting.



(a) (b)

Figure 4: (a) The final step-wise hallucination localization performance of LLMs on PROBE. All models achieve high recall yet relatively low precision, reflecting limitations in correctly identifying supported evidence. (b) Comparison of direct prompting and process-based evaluation. The process-based evaluation consistently achieves higher recall than direct prompting.

4.4 Finetuning Performance

We utilize the Llama-3.1-8B (AI@Meta, 2024) model as the backbone model for fine-tuning to improve evidence finding and evidence evaluation capabilities. We perform full-parameter fine-tuning with a learning rate of $2e^{-5}$. Training uses the Adam optimizer (Kingma, 2014) with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, together with a cosine learning rate scheduler and a 2% warm-up over the total training steps. The fine-tuned model is initialized from a HuggingFace checkpoint and trained on 8 NVIDIA A100 (80GB) GPUs using Fully Sharded Data Parallel (Zhao et al., 2023).

The fine-tuning results are reported in Table 2. Leveraging our high-quality claim–evidence training dataset, the fine-tuned Llama-3.1-8B achieves the best performance on both evidence finding and evidence evaluation tasks. These results demonstrate the effectiveness of our data in enhancing model capabilities for hallucination detection.

5 Related Work

5.1 Granularity of Hallucinations

In terms of hallucination-checking granularity, response-level evaluation (Li et al., 2023) determines only whether an entire model output contains hallucinations. While such binary judgments suffice for short queries, they become uninformative and often lead to false negatives when hallucinations occur locally within long-form responses. To achieve finer granularity, prior work (Niu et al., 2023) introduces span-level or word-level hallucination annotations, labeling only specific regions of text as hallucinated. However, hallucination-span boundaries are structurally difficult to define, making it challenging to construct high-quality demonstrations for in-context learning. To address these limitations, in PROBE, we propose using claims as the basic unit of hallucination checking. Claims provide fine-grained, semantically coherent, and more clearly separable units of meaning. When hallucinated information appears in lengthy LLM responses, evaluating individual extracted claims improves hallucination detection accuracy (Akbar et al., 2024). Figure 5 illustrates how claim-level hallucination annotation contrasts with coarser (response-level) and finer but less structured (span-level) alternatives.

5.2 Hallucination Detection Benchmarks

With the rise of LLMs, the detection of hallucinations has become increasingly challenging, necessitating the development of high-quality datasets for LLM evaluation. RAGTruth (Niu et al., 2023) presents a corpus specifically designed for analyzing word/span-level hallucinations in RAG settings. Their dataset comprises nearly 18,000 naturally generated responses with meticulous manual annotations at both response and word levels. The FACTS Grounding Leaderboard (Jacovi et al., 2025) focuses on evaluating models' ability to generate responses fully grounded in provided context documents while fulfilling users' requests. This approach uses automated judge models to assess factuality with respect to a given context, though the evaluation remains primarily outcome-focused. HaluBench (Ravi et al., 2024) sources examples from existing benchmarks such as HaluEval (Li et al., 2023) and RAGTruth (Niu et al., 2023), yielding a large collection of model-generated and human-annotated hallucinated samples across diverse domains. Both FactScore (Min et al., 2023) and LongFact (Wei et al., 2024) propose to decompose a model's response into atomic facts and evaluate the proportion of facts supported by reliable sources or web search. FastFact (Wan et al., 2025) advances this pipeline by integrating confidence-based pre-verification, significantly reducing the

Table 3: Comparison of PROBE with existing hallucination detection benchmarks. PROBE is the first to provide process-based evaluation at claim-level granularity, enabling detailed diagnosis of detection capabilities.

Feature	PROBE (Ours)	RAGTruth	FACTS Grounding	HaluBench	HaluEval	FactScore
Evaluation Level	**Claim-level**	Span-level	Response-level	Response/Span-level	Response-level	Claim-level
Process-Based	**Yes**	No	No	No	No	No
Task Diversity	Summarization, QA **Style Transfer**	QA, Data-to-Text, Summarization	Long-form generation	QA domains	QA, Summarization, Dialogue	Biography generation
Annotation	**Step-wise**	Span-level	Binary	Binary	Binary	Fact-level
Scale	12,000 samples **(120,404 claims)**	14,289 spans (17,790 responses)	1,719 examples	Large-scale	30,000+ samples	10,000 facts

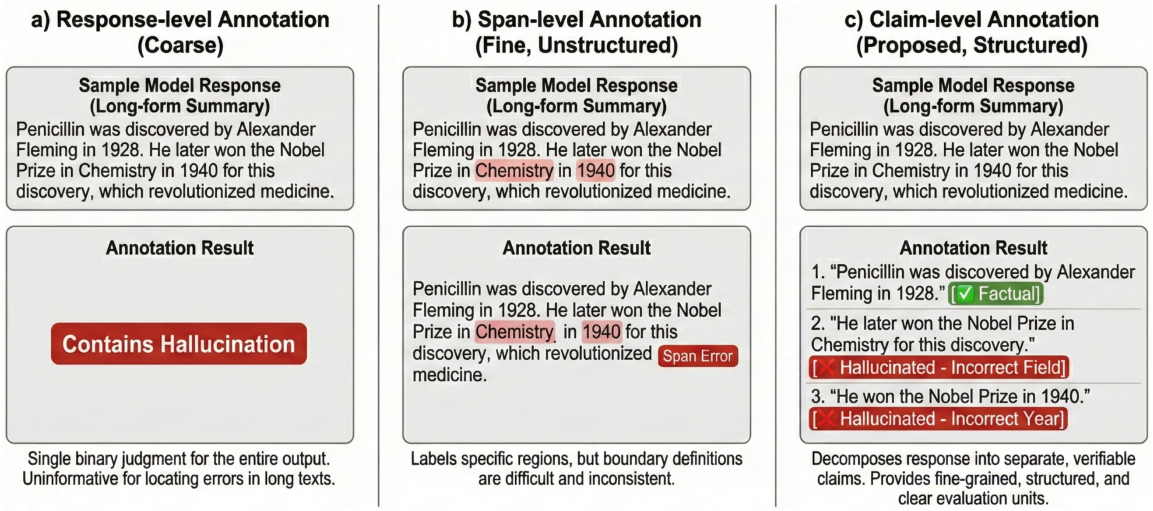


Figure 5: Comparison of hallucination annotation granularities. Claim-level annotation offers a balance of granularity and structure, overcoming the limitations of coarse response-level and unstructured span-level methods, especially for long-form model responses.

cost of web searching. While these benchmarks cover a wide range of domains and annotation granularities, none explicitly evaluate the multi-step process involved in hallucination detection. Table 3 presents a detailed comparison between PROBE and existing hallucination detection benchmarks.

5.3 Process-Based Evaluation

The importance of process-based evaluation has been recognized across multiple domains. In mathematical reasoning, PRM800K (Lightman et al., 2023) provides step-wise annotations for model-generated solutions. CriticBench (Lin et al., 2024) evaluates language models' abilities to critique solutions and correct mistakes. MathCheck (Zhou et al., 2024) synthesizes solutions containing erroneous steps for evaluation purposes. ProcessBench (Zheng et al., 2024) introduces a benchmark for identifying process errors in mathematical reasoning, demonstrating the value of step-wise evaluation for complex reasoning tasks. Their work shows the importance of process supervision and step-wise evaluation in the reasoning process. Our work falls into this category by providing a structured framework for evaluating each component of the fact-checking process.

6 Conclusion

In this paper, we introduce PROBE, a process-based benchmark for evaluating the ability to identify hallucinations in long-form text generation. Unlike prior outcome-based evaluations, PROBE enables an in-depth, step-by-step analysis of LLM capabilities in hallucination detection, providing concrete insights into why models fail to detect hallucinations. Through extensive evaluations on frontier LLMs, we make two key observations: (1) LLMs consistently struggle with evidence finding and evidence evaluation, and (2) step-wise evaluation achieves superior performance compared to direct prompting. Furthermore, we demonstrate that fine-tuning Llama on PROBE yields competitive results, suggesting that high-quality supervision can enable the development of specialized models that outperform general-purpose LLMs in process-based halluciantion detection. Through PROBE, we aim to establish a new standard for hallucination detection by moving beyond outcome-based metrics toward process-oriented assessment, enabling more transparent and reliable LLM deployments.

7 Limitations

Due to limited computational resources, we do not fine-tune larger LLMs, such as LLaMA-3.1-70B, using PROBE. Moreover, our approach relies on multiple language model invocations to assess hallucination detection, resulting in higher latency compared to direct prompting. This overhead could be partially mitigated through confidence-based pre-selection during evidence finding. Finally, we limit our benchmark to English and non-expert domains, as extending PROBE to other languages or specialized domains (e.g., finance and medicine) would require additional domain expertise. Future work could extend PROBE to multilingual and domain-specific settings by leveraging suitable multilingual models and domain-specific language models.

8 Ethical Considerations

This work is in full compliance with the Ethics Policy of the ACL. We acknowledge that responses generated by LLMs in this study may contain inaccuracies. Aside from this, to the best of our knowledge, there are no additional ethical issues associated with this paper.

References

AI@Meta. 2024. Llama 3 model card.

Shayan Ali Akbar, Md Mosharaf Hossain, Tess Wood, Si-Chi Chin, Erica M Salinas, Victor Alvarez, and Erwin Cornejo. 2024. Hallumeasure: Fine-grained hallucination measurement using chain-of-thought reasoning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15020–15037.

Dean Allemang and Juan Sequeda. 2024. Increasing the llm accuracy for question answering: Ontologies to the rescue! *arXiv preprint arXiv:2405.11706*.

Anthropic. 2025. Introducing claude sonnet 4.5.

Sabine Foly, Jingshu Liu, Jean-Gabriel Barthelemy, Gaëtan Caillaut, Raheel Qader, Mariam Nakhle, and Arezki Sadoune. 2025. Clean-wikipedia-english-articles.

Xiangkun Hu, Dongyu Ru, Lin Qiu, Qipeng Guo, Tianhang Zhang, Yang Xu, Yun Luo, Pengfei Liu, Yue Zhang, and Zheng Zhang. 2024. Refchecker: Reference-based fine-grained hallucination checker and benchmark for large language models. *arXiv preprint arXiv:2405.14486*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Alon Jacovi, Andrew Wang, Chris Alberti, Connie Tao, Jon Lipovetz, Kate Olszewska, Lukas Haas, Michelle Liu, Nate Keating, Adam Bloniarz, and 1 others. 2025. The facts grounding leaderboard: Benchmarking llms' ability to ground responses to long-form input. *arXiv preprint arXiv:2501.03200*.

Diederik P Kingma. 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, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Junyi Li, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. Halueval: A large-scale hallucination evaluation benchmark for large language models. *arXiv preprint arXiv:2305.11747*.

Junyi Li, Tianyi Tang, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2024. Pre-trained language models for text generation: A survey. *ACM Computing Surveys*, 56(9):1–39.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Zicheng Lin, Zhibin Gou, Tian Liang, Ruilin Luo, Haowei Liu, and Yujiu Yang. 2024. Criticbench: Benchmarking llms for critique-correct reasoning. *arXiv preprint arXiv:2402.14809*.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100.

Abhika Mishra, Akari Asai, Vidhisha Balachandran, Yizhong Wang, Graham Neubig, Yulia Tsvetkov, and Hannaneh Hajishirzi. 2024. Fine-grained hallucination detection and editing for language models. *arXiv preprint arXiv:2401.06855*.

Mistral.ai. 2024. Mixtral-8x22b.

Cheng Niu, Yuanhao Wu, Juno Zhu, Siliang Xu, Kashun Shum, Randy Zhong, Juntong Song, and Tong Zhang. 2023. Ragtruth: A hallucination corpus for developing trustworthy retrieval-augmented language models. *arXiv preprint arXiv:2401.00396*.

OpenAI. 2025. gpt-oss-120b & gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

Selvan Sunitha Ravi, Bartosz Mielczarek, Anand Kannappan, Douwe Kiela, and Rebecca Qian. 2024. Lynx: An open source hallucination evaluation model. *arXiv preprint arXiv:2407.08488*.

Juntong Song, Xingguang Wang, Juno Zhu, Yuanhao Wu, Xuxin Cheng, Randy Zhong, and Cheng Niu. 2024. Rag-hat: A hallucination-aware tuning pipeline for llm in retrieval-augmented generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1548–1558.

Yingjia Wan, Haochen Tan, Xiao Zhu, Xinyu Zhou, Zhiwei Li, Qingsong Lv, Changxuan Sun, Jiaqi Zeng, Yi Xu, Jianqiao Lu, and 1 others. 2025. Fastfact: Faster, stronger long-form factuality evaluations in llms. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 23814–23854.

Jerry Wei, Chengrun Yang, Xinying Song, Yifeng Lu, Nathan Hu, Jie Huang, Dustin Tran, Daiyi Peng, Ruibo Liu, Da Huang, and 1 others. 2024. Long-form factuality in large language models. *Advances in Neural Information Processing Systems*, 37:80756–80827.

Yang Zhang, Hanlei Jin, Dan Meng, Jun Wang, and Jinghua Tan. 2024. A comprehensive survey on process-oriented automatic text summarization with exploration of llm-based methods. *arXiv preprint arXiv:2403.02901*.

Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, and 1 others. 2023. Pytorch fsdp: experiences on scaling fully sharded data parallel. *arXiv preprint arXiv:2304.11277*.

Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2024. Processbench: Identifying process errors in mathematical reasoning. *arXiv preprint arXiv:2412.06559*.

Zihao Zhou, Shudong Liu, Maizhen Ning, Wei Liu, Jindong Wang, Derek F Wong, Xiaowei Huang, Qiufeng Wang, and Kaizhu Huang. 2024. Is your model really a good math reasoner? evaluating mathematical reasoning with checklist. *arXiv preprint arXiv:2407.08733*.

A Example of Hallucinations

Base Answer from QAs
Q: What are the geographical areas that the term "Siam" is used to refer to, and what is the former name of Thailand? A: The term "Siam" is used to refer to the historical region of Central Thailand, usually including Southern Thailand, as well as other places such as Siam, Ohio, Siam area in Bangkok, and Rue de Siam in Brest, France. The former name of Thailand is Siam.
Hallucinated Example with **Complexity 1**
Q: What are the geographical areas that the term "Siam" is used to refer to, and what is the former name of Thailand? A: The term "Siam" is used to refer to the historical region of Central Thailand, usually including Southern Thailand, as well as other places such as Siam, Ohio, Siam area in Bangkok, and Rue de Siam in Brest, France. Historically, Siam was also used as a name for a short-lived French colony in Southeast Asia during the 19th century.
Hallucinated Example with **Complexity 2**
Q: What are the geographical areas that the term "Siam" is used to refer to, and what is the former name of Thailand? A: The term "Siam" is used to refer to the historical region of Central Thailand, usually including Southern Thailand, as well as other places such as Siam, Ohio, Siam area in Bangkok, and Rue de Siam in Brest, France. Additionally, Siam was also used as a name for a short-lived French colony in Southeast Asia during the 19th century. Furthermore, the term "Siam" is also used to refer to a specific region in the southern part of Laos. The former name of Thailand is Siam.
Hallucinated Example with **Complexity 3**
Q: What are the geographical areas that the term "Siam" is used to refer to, and what is the former name of Thailand? A: The term "Siam" is used to refer to the historical region of Central Thailand, usually including Southern Thailand, as well as other places such as Siam, Ohio, Siam area in Bangkok, and Rue de Siam in Brest, France. Historically, Siam was also used as a name for a short-lived French colony in Southeast Asia during the 19th century. Additionally, the term "Siam" is also used to refer to a specific region in the southern part of Laos. The former name of Thailand is Siam, and interestingly, the name "Siam" was even considered as an alternative name for the country of Cambodia during its independence movement in the mid-20th century.

Table 4: Examples of the three types of examples with different level of hallucination.

B Generation Prompt

The benchmark generation process follows a well-defined pipeline that can be divided into three distinct stages. Here we show the prompt for each step.

B.1 STAGE 1: Base Content Generation

Summarization: LLMs are prompted to produce a concise summary of each source document.

Listing 1: Summary Generation System Prompt

```
You are a helpful assistant that generates a summary of a factual text. Your job is
    ↪  to extract the main facts from the text and present them in a concise
    ↪ manner. You must not hallucinate. You must not make up any facts. You must
    ↪ not use any information that is not in the text.
```

```
The summary should be one or a few paragraphs and be detailed -- containing all
    ↪ important names, dates, numbers, or facts about actions taken or events that
    ↪  took place. Do not output any other text than the summary.
```

Listing 2: Summary Generation User Prompt

```
Summarize the following text:
{TEXT}

Reminders:
- You must not hallucinate.
- You must not make up any facts.
- You must not use any information that is not in the text.
- The summary should be one or a few paragraphs and be detailed -- containing all
    ↪ important names, dates, numbers, or facts about actions taken or events that
    ↪  took place.
- Do not output any other text than the summary.
```

Question and Answering: We first prompt the language model to generate a question about the provided text that can be answered with a small number (2-4) of concrete facts. The model is then asked to answer each generated question using only the retrieved passages corresponding to the article.

Listing 3: Question Generation System Prompt

```
You are a helpful assistant that generates questions about factual text. Your job
    ↪ is to create a question that requires a brief answer containing only a
    ↪ handful of concrete facts (e.g., one date, one location, and one name, or
    ↪ two dates, etc.). The question should be based on the initial segment of the
    ↪  text and should require an answer that includes 2-4 specific, concrete
    ↪ facts.

The question should be specific enough to require only a few factual statements,
    ↪ not long-winded explanations. Do not output any other text than the question.
    ↪
```

Listing 4: Question Generation User Prompt

```
Generate a question about the following text that requires a brief answer with only
    ↪  a handful of concrete facts:
{TEXT}

Reminders:
- The question should require an answer with only 2-4 concrete facts
- The question should be based on the initial segment of the text
- The answer should include specific facts like dates, locations, names, or numbers
- The question should be specific enough to require only a few factual statements
- Do not output any other text than the question.
```

Listing 5: Answer Generation System Prompt

```
You are a helpful assistant that generates brief answers to questions about factual
    ↪  text. Your job is to provide a concise answer that includes only the
    ↪ specific factual claims requested, based on the text. You must not
    ↪ hallucinate. You must not make up any facts. You must not use any
    ↪ information that is not in the text.
```

```
The answer should be brief, use simple English, and contain roughly one sentence
    ↪ per fact being asked. Avoid long-winded paragraphs. Do not output any other
    ↪ text than the answer.
```

Listing 6: Answer Generation User Prompt

```
Answer the following question based on the text:
QUESTION: {QUESTION}
TEXT: {TEXT}

Reminders:
- You must not hallucinate.
- You must not make up any facts.
- You must not use any information that is not in the text.
- The answer should be brief, use simple English, and contain roughly one sentence
    ↪ per fact being asked.
- Avoid long-winded paragraphs and unnecessary details.
- Do not output any other text than the answer.
```

Style Transfer: The goal is to rewrite the article in a different style while preserving its meaning. We consider four target styles: blog post, lecture notes, FAQ, and textbook. For each article, one style is selected at random, and the LLM is prompted to rewrite the original Wikipedia text accordingly, yielding approximately 25% of samples in each style category.

Listing 7: Blog Post Style Transfer Prompt

```
You are a helpful assistant that converts Wikipedia articles into engaging blog
    ↪ post style. Your job is to rewrite the factual content in a more
    ↪ conversational, engaging blog post format while preserving all the factual
    ↪ information. You must not hallucinate. You must not make up any facts. You
    ↪ must not use any information that is not in the text. If you cannot fit
    ↪ information into the blog post format naturally, you may omit it, but never
    ↪ make up new information.

The blog post should be engaging, conversational, and accessible while maintaining
    ↪ factual accuracy. Use a personal tone, include transitions, and make it
    ↪ readable for a general audience. Do not output any other text than the blog
    ↪ post.
---
Convert the following Wikipedia article into an engaging blog post style:

{TEXT}

Reminders:
- You must not hallucinate.
- You must not make up any facts.
- You must not use any information that is not in the text.
- If you cannot fit information into the blog post format naturally, you may omit
    ↪ it, but never make up new information.
- The blog post should be engaging, conversational, and accessible while
    ↪ maintaining factual accuracy.
- Use a personal tone, include transitions, and make it readable for a general
    ↪ audience.
- Do not output any other text than the blog post.
```

Listing 8: Textbook Style Transfer Prompt

```
You are a helpful assistant that converts Wikipedia articles into textbook excerpt style. Your job is to rewrite the factual content in a clear, educational textbook format while preserving all the factual information. You must not hallucinate. You must not make up any facts. You must not use any information that is not in the text. If you cannot fit information into the textbook format naturally, you may omit it, but never make up new information.

The textbook excerpt should be clear, well-structured, educational, and suitable for students. Use formal academic language, include clear headings or sections, and present information in a logical, educational manner. Do not output any other text than the textbook excerpt.
----
Convert the following Wikipedia article into a clear textbook excerpt style:

{TEXT}

Reminders:
- You must not hallucinate.
- You must not make up any facts.
- You must not use any information that is not in the text.
- If you cannot fit information into the textbook format naturally, you may omit it, but never make up new information.
- The textbook excerpt should be clear, well-structured, educational, and suitable for students.
- Use formal academic language, include clear headings or sections, and present information in a logical, educational manner.
- Do not output any other text than the textbook excerpt.
```

Listing 9: Lecture Notes Style Transfer System Prompt

```
You are a helpful assistant that converts Wikipedia articles into lecture notes style. Your job is to rewrite the factual content in a concise, note-taking format while preserving all the factual information. You must not hallucinate. You must not make up any facts. You must not use any information that is not in the text. If you cannot fit information into the lecture notes format naturally, you may omit it, but never make up new information.

The lecture notes should be concise, well-organized, and easy to follow. Use bullet points, numbered lists, and clear structure. Include key concepts, important dates, and main points in a format suitable for students taking notes. Do not output any other text than the lecture notes.
---
Convert the following Wikipedia article into concise lecture notes style:

{TEXT}

Reminders:
- You must not hallucinate.
- You must not make up any facts.
- You must not use any information that is not in the text.
```

```
- If you cannot fit information into the lecture notes format naturally, you may
    ↪ omit it, but never make up new information.
- The lecture notes should be concise, well-organized, and easy to follow.
- Use bullet points, numbered lists, and clear structure.
- Include key concepts, important dates, and main points in a format suitable for
    ↪ students taking notes.
- Do not output any other text than the lecture notes.
```

Listing 10: FAQ Style Transfer System Prompt

```
You are a helpful assistant that converts Wikipedia articles into FAQ (Frequently
    ↪ Asked Questions) format. Your job is to rewrite the factual content as a
    ↪ series of questions and answers while preserving all the factual information.
    ↪  You must not hallucinate. You must not make up any facts. You must not use
    ↪ any information that is not in the text. If you cannot fit information into
    ↪ the FAQ format naturally, you may omit it, but never make up new information.
    ↪

The FAQ should be organized as a series of relevant questions with clear, factual
    ↪ answers. Questions should be natural and cover the main topics from the
    ↪ article. Answers should be concise but informative. Do not output any other
    ↪ text than the FAQ.
---
Convert the following Wikipedia article into FAQ (Frequently Asked Questions)
    ↪ format:

{TEXT}

Reminders:
- You must not hallucinate.
- You must not make up any facts.
- You must not use any information that is not in the text.
- If you cannot fit information into the FAQ format naturally, you may omit it, but
    ↪  never make up new information.
- The FAQ should be organized as a series of relevant questions with clear, factual
    ↪  answers.
- Questions should be natural and cover the main topics from the article.
- Answers should be concise but informative.
- Do not output any other text than the FAQ.
```

B.2 STAGE 2: Hallucination Insertion

To precisely control hallucination locations, we generate unfaithful hallucinations by synthetically injecting poisoned semantics into the text.

Listing 11: Hallucination Invention System Prompt for Summaries

```
You are an adversarial assistant that invents new facts about a text. Your job is
    ↪ to invent {NEW_FACT_COUNT} new facts about the text that are not in the text
    ↪  (and are not a logical consequence of any information in the text) but are
    ↪ plausible or believable. The facts should be formulated as statements that
    ↪ can be evaluated as true or false. Do not output any text other than the
    ↪ facts separated by newlines.
```

Listing 12: Hallucination Invention User Prompt for Summaries

```
Invent {NEW_FACT_COUNT} new facts about the following text. Output the facts
    ↪ separated by newlines -- each fact should be one a single line. Do not
    ↪ output any other text than the facts separated by newlines.
{TEXT}

Reminders:
 - The facts should be formulated as statements that can be evaluated as true or
     ↪ false.
 - The facts should be plausible or believable.
 - The facts should not be in the text or be a logical consequence of any
     ↪ information in the text.
 - Do not output any other text than the facts separated by newlines.
```

Listing 13: Answer Editor System Prompt

```
You are a helpful assistant that edits an answer to include new facts that are
    ↪ provided. Your job is to edit the answer in such a way that it includes new
    ↪ facts that are provided and are not in the original answer. The inclusion of
    ↪  these new facts must be seamless and natural, and should appear across the
    ↪ whole text rather than just being added at the end. Keep the answer brief
    ↪ and simple, with roughly one sentence per fact. Do not output any other text
    ↪  than the edited answer. Do not hallucinate = do not make up any new facts.
    ↪ You must use only the original answer text and the newly provided facts.
```

Listing 14: Answer Editor User Prompt

```
Edit the following ANSWER to include the NEW FACTS that are provided. The inclusion
    ↪  of these new facts must be seamless and natural, and should appear across
    ↪ the whole text rather than just being added at the end. Keep the answer
    ↪ brief and simple, with roughly one sentence per fact. Do not output any
    ↪ other text than the edited answer. Do not hallucinate. Do not make up any
    ↪ new facts. You must use only the original ANSWER text and the newly provided
    ↪  facts.

ANSWER:
{ANSWER}

NEW FACTS:
{FACTS}

Reminders:
 - The inclusion of these new facts must be seamless and natural, and should appear
     ↪  across the whole text rather than just being added at the end.
 - Keep the answer brief and simple, with roughly one sentence per fact asked.
 - Do not output any other text than the edited answer.
 - Do not hallucinate = do not make up any new facts.
 - You must use only the original answer text and the newly provided facts.
 - Try not to leave out any information from the original answer text.
```

B.3 STAGE 3: Claim-Evidence Pair Generation

Step 1: Claim Decomposition.

Listing 15: Claim decomposition prompt

```
You are AssertionDecomposer, a helpful text processing assistant that takes a text
    ↪ and returns a detailed list of all assertions that the text makes, no matter
    ↪  how small.

The TEXT you are given is an output of generation, e.g., an informed answer to a
    ↪ question, a summary/essay/report on a document or a piece of text, a
    ↪ modification of a text including some stylistic changes, etc.. You can
    ↪ assume that the TEXT you will be given is the output of a generation based
    ↪ on some information sources that have been provided in order to produce the
    ↪ TEXT.

Your job is to find all assertions that the TEXT (including its section headings,
    ↪ tables, figure captions, or other non-textual elements) makes, no matter how
    ↪  small, taking particular care to note down all facts that the text makes
    ↪ including any numbers, dates, names, or claims about something happening.
    ↪ You will then output a list of assertions in the following Markdown-inspired
    ↪  format:

> SEGMENT#1
Assertion of a fact formulated from SEGMENT#1.
> SEGMENT#2
Assertion of a fact formulated from SEGMENT#2.
...
```

Step 2: Evidence Finding

Listing 16: Evidence finding prompt

```
Find segments/sentences in the ground truth source that are relevant to the
    ↪ assertion, i.e., that support, contradict, or are otherwise directly
    ↪ relevant to the assertion. Think step by step, then produce your final
    ↪ output by marking it with "FINAL OUTPUT:".

<ASSERTION>
{ASSERTION}
</ASSERTION>

<GROUND_TRUTH_SOURCE>
{GROUND_TRUTH_SOURCE}
</GROUND_TRUTH_SOURCE>

Output format:
Thinking: [Your thinking]
FINAL OUTPUT:
Segment #1
...
```

Step 3: Evidence Evaluation

Listing 17: Evidence evaluation prompt

```
Evaluate whether the following evidence supports the given assertion.

ASSERTION:
```

```
{ASSERTION}

EVIDENCE EXCERPT:
{EVIDENCE}

SURROUNDING CONTEXT:
{SURROUNDING_CONTEXT}

Please analyze whether the evidence excerpt ALONE supports the assertion. Provide
    ↪ your reasoning and then give a clear PASS or FAIL verdict. The analysis must
    ↪  rely on the evidence, not the context. The context is informative to help
    ↪ you better understand the assertion and the evidence, but it is not a crutch
    ↪  to justify the assertion from which to make far-stretching logical leaps.
    ↪ When reasoning, do not rely on any statements in the context to make your
    ↪ case. Use the context only to enhance your understanding of the assertion
    ↪ and the evidence, and never to justify the assertion.

REASONING: [Your analysis here relying on the evidence alone]
VERDICT: [PASS or FAIL]
```