HAD: HAllucination Detection Language Models Based on a Comprehensive Hallucination Taxonomy

Fan Xu[1], Xinyu Hu[1], Zhenghan Yu[1], Li Lin[1], Xu Zhang[1], Yang Zhang[2], Wei Zhou[2], Jinjie Gu[3], Xiaojun Wan[1]
[1]Wangxuan Institute of Computer Technology, Peking University
[2]Alibaba Group [3]Fudan University
{xufan2000,huxinyu,zhangxu,wanxiaojun}@pku.edu.cn, {efsotr_l,zhenghanyu}@stu.pku.edu.cn
{yaoling.zy,jinjie.gujj}@antgroup.com, zhouwei546138922@126.com

Abstract

The increasing reliance on natural language generation (NLG) models, particularly large language models, has raised concerns about the reliability and accuracy of their outputs. A key challenge is hallucination, where models produce plausible but incorrect information. As a result, hallucination detection has become a critical task. In this work, we introduce a comprehensive hallucination taxonomy with 11 categories across various NLG tasks and propose the **HA**llucination **D**etection (**HAD**) models[1], which integrate hallucination detection, span-level identification, and correction into a single inference process. Trained on an elaborate synthetic dataset of about 90K samples, our **HAD** models are versatile and can be applied to various NLG tasks. We also carefully annotate a test set for hallucination detection, called **HADTest**, which contains 2,248 samples. Evaluations on in-domain and out-of-domain test sets show that our HAD models generally outperform the existing baselines, achieving state-of-the-art results on HaluEval, FactCHD, and FaithBench, confirming their robustness and versatility.

1 Introduction

The rapid advancement of natural language generation (NLG) has been largely driven by the development of large language models (LLMs) (Li et al., 2024). These models, characterized by their ability to process and generate human-like text, have found applications across diverse domains (Chkirbene et al., 2024), including content creation, customer support, and personalized education. Despite their impressive performance, LLMs still have limitations. One of the most concerning issues is the phenomenon known as **hallucination**, in which models produce outputs that seem credible but are factually incorrect or unfaithful to the provided context (Huang et al., 2023). This problem poses significant challenges for users relying on LLMs for accurate information, raising critical concerns about the trustworthiness and accountability of AI-generated content. As LLMs continue to evolve and integrate into various applications, addressing hallucination is paramount to ensuring the reliability of these technologies in real-world scenarios.

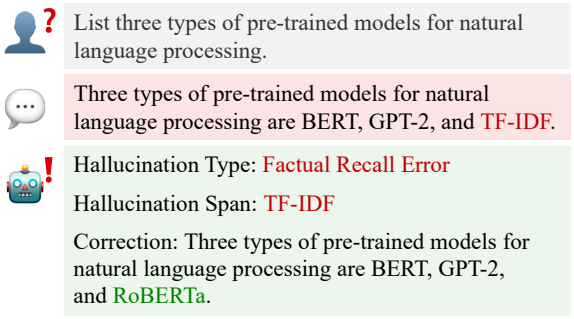

Figure 1: Example of the hallucination detection process.

Hallucination in NLG can be broadly categorized into two primary dimensions: factuality and faithfulness (Huang et al., 2023). **Factuality** refers to the degree to which the generated content aligns with real-world facts. Meanwhile, **faithfulness** refers to the relationship between the generated text and the input, as well as the consistency within the output itself. It emphasizes the importance of following the instructions and accurately representing the information provided to the model. Both aspects of hallucination impact the overall reliability of LLMs, so it's essential to understand and address both dimensions.

Despite the significant advancements in research on hallucination in LLMs, existing works still have notable limitations. Many studies focus solely on factuality (Chen et al., 2024) or faithfulness (Niu et al., 2023), neglecting the comprehensive assessment of both aspects. Additionally, several prior studies target specific tasks (Gu et al., 2024; Min

[1]https://github.com/pku0xff/HAD

et al., 2023), which restricts their applicability and generalizability. Furthermore, the taxonomy they employed is typically coarse-grained, lacking the nuanced granularity necessary for detailed analysis and effective application across diverse contexts.

In this work, we propose a more comprehensive and fine-grained hallucination taxonomy, covering both faithful and factual aspects. Our taxonomy is organized into three hierarchical levels and defines 11 specific types of hallucinations. Recognizing that NLG tasks vary in their requirements for these dimensions (Ji et al., 2023), we map each task type to corresponding hallucination categories. Based on this correspondence, we synthesize a set of 90,172 training data samples by modifying correct task data to include various types of hallucinations across multiple NLG tasks. The synthetic data consists of hallucinated outputs and the spans of hallucinations as its key components.

Our **HA**llucination **D**etection (**HAD**) models are obtained by fine-tuning `Qwen2.5-7B-Instruct` and `Qwen2.5-14B-Instruct` (Team, 2024) with synthetic data, enabling them to jointly perform type classification, span-level identification, and correction within a unified inference. To obtain high-quality test data, we manually annotate the **HADTest** dataset, including diverse hallucination samples paired with an equal number of correct samples, with 2,248 samples in total. We evaluate our models and other existing baselines through both in-domain and out-of-domain tests. The results show that HAD models outperform both general-purpose base models and specified hallucination detection models. The in-domain results demonstrate their functionality, while the out-of-domain results highlight their accuracy and generalizability. The datasets and models will be publicly available soon.

Overall, our main contributions can be summarized as follows:

1. We present a fine-grained hallucination taxonomy that encompasses both factuality and faithfulness, considering more subtle issues related to hallucinations.
2. Aligned with our hallucination taxonomy, we create a large-scale, multi-task training dataset along with a high-quality test set, **HADTest** for the hallucination detection and correction.
3. We develop the **HAD** models that are capable of handling hallucination categorization, span-level detection, and correction within a single inference. These models are applicable to various NLG tasks and have achieved state-of-the-art performance on multiple benchmarks.

2 Hallucination Taxonomy

We define **hallucination** as the phenomenon where the generated output appears credible but is factually incorrect or not faithful to the provided context, following the taxonomy proposed in previous work (Huang et al., 2023). Typically, A natural language generation task consists of:

- **Instruction:** Description of the task, indicating the task type and specific constraints, such as the subject matter, format, and length of the generated output.
- **Input context:** The relevant context for generating the output, such as the source document in summarization, translation, or contextual QA.
- **Task Output:** The result generated to complete the task.

We categorize hallucinations hierarchically based on the nature of the inconsistencies and the content involved, with 11 fine-grained categories at the third level. Our taxonomy is more comprehensive than existing ones, covering a broader range of scenarios. To demonstrate the empirical relevance and coverage of this taxonomy, we map its categories to related concepts and terminologies discussed in prior works, as presented in Appendix A.

2.1 Faithfulness Hallucination

Faithfulness hallucinations stem from inconsistencies between the input content and generated content, or from issues within the generated content, without relying on external factual information. These hallucinations can be classified into three main types: instruction inconsistency, input context inconsistency, and internal inconsistency.

A. Instruction Inconsistency This occurs when generated content fails to align with the given instruction, not meeting task, content, or format requirements.

1. **Task Type Inconsistency (TTI)**: The output represents a different task type than instructed. Deviations within the same task type, such as violating specific requirements, are excluded.
2. **Task Requirement Inconsistency (TRI)**: The output doesn't meet specified task requirements, such as format, length, subject, or tone. This is due to a failure to follow instructions, rather than to contradictions in content.

B. Input Context Inconsistency This happens when the generated content contradicts or misinterprets the input context.

3. **Contradiction with Input Content (CwIC)**: The output contradicts the input, presenting information incompatible with the context, often due to failure to recall or misunderstanding the provided information.
4. **Baseless Information (BI)**: In tasks that require strict adherence to the given context, the output contains unsupported information. Tasks that seek new information do not suffer from this problem.
5. **Information Omission (IO)**: When the task requires a complete and accurate representation of the provided context, the output omits details presented in the input.

C. Internal Inconsistency This arises when generated content contains contradictions, logical errors, or structural problems, leading to incoherent or implausible results.

6. **Contradiction within Output Content (CwOC)**: The output includes contradictory statements or flawed reasoning.
7. **Structural Incoherence (SI)**: The output contains redundant, repetitive, or disjointed statements that do not enhance clarity or value.

2.2 Factuality Hallucination

Factuality hallucinations occur when the generated content contains inaccuracies, distortions, or fabrications that do not align with external reality. According to whether they can be directly refuted by established world knowledge, factuality hallucinations can be categorized into fact contradiction and fact fabrication.

D. Fact Contradiction Fact contradiction happens when the generated content directly contradicts established knowledge.

8. **Factual Recall Error (FRE)**: The generated text contains an incorrect atomic fact due to the model's inability to accurately recall or access relevant knowledge.
9. **Factual Inference Error (FIE)**: The content contains incomplete or misinterpreted facts, such as confusion between time periods, individuals, or events, omission of key details, or errors in the order of causes and effects, leading to the facts being presented incorrectly.

E. Fact Fabrication Fact fabrication refers to content presenting unverifiable information not based on real-world knowledge, excluding artistic or creative fiction.

10. **Fabricated Entity (FE)**: The generated output introduces entirely new, fabricated entities such as concepts, names, or objects that lack any real-world basis.
11. **Fictional Attribution (FA)**: The generated output fabricates information about real entities, such as unverified claims or quotes, that cannot be directly confirmed or disproven by reliable sources. Unlike a Fabricated Entity, this error does not introduce entirely new entities.

3 Data Construction

The goal of our HAD models is to provide a fine-grained detection and classification of hallucination, and then correct it. Given a **task input** and a **task output**, the model is required to classify the hallucination into a fine-grained **type**, identify the precise **span** of the hallucinated content, and provide a **correction** to the output. The task input is a combination of instruction and input context, considering that in many scenarios, people do not explicitly distinguish them. Our models aim to apply to multiple NLG tasks and hallucination types, which places high demands on data diversity. In the following subsections, we will introduce how we acquire the training and test data.

3.1 Source Data Selection

Different tasks can lead to various types of hallucinations (Ji et al., 2023). We adopt ELI5 (Fan et al., 2019) as the data source for the long-form QA task. From the Super-NaturalInstruction (SNI) dataset (Wang et al., 2022), we sample data for tasks including story writing, poem writing, paraphrasing, data-to-text, summarization, contextual QA and short-form QA. For math reasoning, the data source is GSM8K (Cobbe et al., 2021). Data of dialogue task comes from the FaithDial dataset (Dziri et al., 2022). Additionally, we use Alpaca (Taori et al., 2023) for the general instruction following task.

3.2 Hallucination Synthesis

Hallucination Injection For each dataset, we manually assess which hallucination types may occur based on the task definition. All hallucination data are constructed with GPT-4o(`gpt-4o-2024-08-06`) by disturbing the correct

output based on the type definition and 3-shot examples. The examples differ by the hallucination type. To improve the variety, the examples for each data item are sampled independently. The prompt template can be found in Table 10 in Appendix B. We sample 10,000 source data for each hallucination type, set the temperature to 1.0, and sample 5 hallucination candidates for each data item.

Automatic Filtering After the hallucination injection, we check and filter the candidates with a few criteria, which vary between hallucination types. Some issues may exist in the injection stage, including the failure to follow instructions, misunderstanding the hallucination type, or misidentifying the hallucination span. To address these issues, we write both general and task-specific criteria and prompt GPT-4o to analyze whether all the criteria are met. The prompt template and criteria for data verification are shown in Table 11 and 12 in Appendix B. The pass rate at the filtering stage is 82.3%. After filtering, there are 90088 hallucination data left.

3.3 Test Data Annotation

We annotate a test set, HADTest, consisting of 2,248 samples. Initially, 1,240 samples are selected by sampling 20 data items per task and hallucination type. Annotation is conducted by two volunteers who are familiar with the field of LLMs. Each sample is assessed independently by two annotators to ensure it meets predefined criteria and correctly matches the hallucination type. Disagreements are resolved by editing the output to meet the criteria. Guidelines of annotation can be found in Appendix B. The raw test set had a pass rate of 66.37% and an inter-annotator agreement of 80.56%. After filtering and editing, the final HADTest consists of 1,124 hallucinated samples, balanced with an equal number of non-hallucinated samples. Detailed statistics are shown in Figure 2.

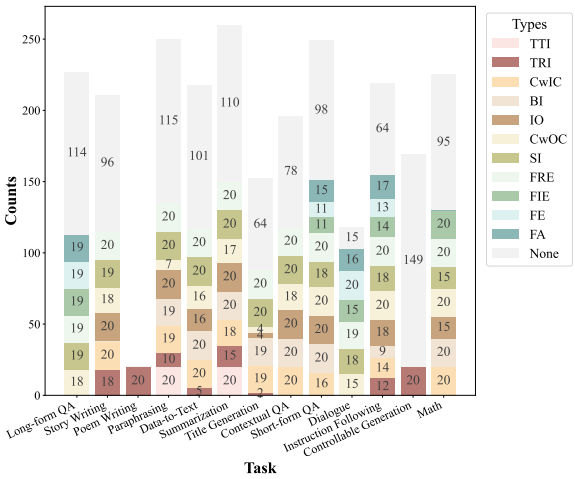

Figure 2: Statistics of our constructed HADTest.

4 Experiments

4.1 Settings

Training Setting We fine-tune the models end-to-end to enable the models to jointly perform hallucination categorization, span-level detection, and correction. The prompt template and response template are provided in Table 13 in Appendix B, based on the task formulation outlined in Section 3. The hallucination types and spans are synthesized following the procedure described in Section 3.2, while the correction is derived from the original ground-truth output of the source datasets without further processing. We sampled 90,172 data points for training and 869 for development. The data ratio in the hallucination dataset reflects both the frequency of hallucination types in real scenarios and their detection difficulty. Specifically, there are about 9,000 samples for each factual hallucination type, around 5,000 for input context inconsistencies, and roughly 2,500 for other types. Positive data are sampled at half the amount of hallucination data while maintaining consistent task distributions.

We use `Qwen2.5-14B-Instruct` as the primary base model, and experiment across different sizes and architectures with `Qwen2.5-7B-Instruct` and `Llama-3.1-8B-Instruct`. Supervised fine-tuning is conducted for one epoch with a learning rate of `1e-5` and a batch size of 256 on 4 H100 GPUs. At the test stage, the temperature is set to 0.

Additionally, we introduce the `HAD-14B-Binary` model, which simplifies hallucination detection to a binary classification task (hallucination vs. non-hallucination). It uses the same data and hyperparameters as the original model. The prompt and response templates are provided in Table 14 in Appendix B.

In-Domain Evaluation We assess three aspects: classification, span detection, and correction. Metrics include Accuracy, Balanced Accuracy, Macro-F1 for classification, and word-level Precision, Recall, F1 for span detection and correction. The latter is evaluated only on hallucinated samples.

Out-of-Domain Evaluation We also test our models on the following external datasets: **HaluEval** (Li et al., 2023) for general hallucination detection, **FactCHD** (Chen et al., 2024) for fact-conflicting hallucinations, and **FaithBench** (Bao et al., 2024b) for hallucinations in summarization task. We simplify hallucination detection to binary classification and report Accuracy for HaluEval, Micro F1 for FactCHD, and Balanced Accuracy and Macro F1 for FaithBench as suggested in the original papers.

Baselines We compare our models against leading LLMs like **GPT-4o**, **GPT-4o mini**, and **DeepSeek V3**, as well as other hallucination detection methods and models including **SelfCheckGPT** (Manakul et al., 2023), **LYNX 8B** (Ravi et al., 2024), **ANAH-v2** (Gu et al., 2024), **FAVA-Model** (Mishra et al., 2024), and **HHEM-2.1-Open** (Bao et al., 2024a). More details about the implementation of baselines can be found in Appendix C.

4.2 Results

Overall, HAD-14B and HAD-14B-Binary outperform the baseline models across multiple evaluation metrics and test sets. Table 1 presents the in-domain test results. HAD-14B delivers superior performance across multiple metrics, achieving a binary classification accuracy of 89.10% and a fine-grained classification accuracy of 83.05%. HAD-14B also excels in span identification, with an F1 score of 76.01%, and in correction, with an F1 score of 77.97%.

As seen in Table 2, HAD models outperform or are comparable with both closed-source, large-scale baseline models and other hallucination detection methods across most OOD test sets, highlighting their robustness. HAD-7B, despite its smaller size, still outperforms baseline models on certain test sets. There is also a trade-off between detection accuracy and functionality, as the performance of HAD-14B-Binary outperforms that of HAD-14B on several test sets. Additional experimental results, ablation studies, error analysis, and case examples are provided in Appendix D.

4.3 Human Evaluation

We additionally conduct a human evaluation to assess the generated corrections. We sample 110 items from HADTest (10 per hallucination type, covering all 11 types), where the model correctly identifies the hallucination type. A qualified human annotator with expertise in large language models then evaluated each correction along six dimensions: (1) overall acceptability, (2) factual accuracy, (3) style preservation, (4) coherence and fluency, (5) whether secondary errors are introduced, and (6) whether overcorrection occurs.

The results indicate that 96 of the 110 corrections (87.3%) are assessed as "Acceptable". In addition, 106 corrections (96.4%) are free of secondary errors, while 109 cases (99.1%) avoid overcorrection of the original text. These results demonstrate that the HAD model's corrections are qualitatively safe. The model reliably fixes the hallucinated content without introducing new errors, altering the original style, or disrupting text coherence. More details can be found in Appendix E.

5 Knowledge Augmentation

Given the limitations of language models, such as the cutoff time of their training data and their constrained knowledge base, retrieval augmentation has become a crucial tool for detecting hallucinations, especially factual ones. Thus, we integrate retrieved information into the hallucination detection process by simply inserting background knowledge into the `task input` part. We concatenate the task input and task output with a "`\n`" as the query and use the `contriever-msmarco` model (Izacard et al., 2021) to retrieve relevant paragraphs from the Wikipedia DPR corpus. For each query, the top 5 documents are selected.

We test knowledge augmentation for HAD-14B and report the F1 score of four fact hallucination types for the HADTest dataset in Table 3. As expected, knowledge augmentation can largely improve the model performance on fact-checking, on both in-domain and out-of-domain test sets. The improvement in the Factual Inference Error is the most obvious, indicating that the model is particularly dependent on external knowledge when dealing with the association of multiple facts.

Given that HAD models are built on 7B and 14B base models with limited knowledge, augmenting them with external information is crucial and effective for fact-based domains. Our experiments demonstrate a simple yet powerful approach to implementing this.

Model	Binary	Fine-Grained			Span			Correction		
	Acc	Acc	BA	F1	Precision	Recall	F1	Precision	Recall	F1
FAVA-Model	59.43	-	-	-	34.69	50.35	36.56	70.84	65.41	66.29
GPT-4o	63.08	40.30	46.61	41.64	56.41	73.03	57.79	63.22	68.93	62.90
GPT-4o mini	60.19	29.76	32.00	29.37	48.00	59.03	46.91	57.99	65.17	58.81
DeepSeek-V3	58.50	39.72	38.48	38.47	46.58	60.45	48.15	40.41	63.23	45.37
HAD-14B	**89.10**	**83.05**	77.38	**76.29**	78.96	78.27	76.01	78.87	78.16	77.97

Table 1: The in-domain test results on the HADTest dataset include evaluations for binary classification, fine-grained classification, span identification, and correction. A dash ("–") indicates that the model does not support the corresponding function.

Model	HaluEval-dial	HaluEval-gen	HaluEval-QA	HaluEval-summ	FactCHD	FaithBench	
	Acc	Acc	Acc	Acc	Micro F1	BA	Macro F1
SelfCheckGPT	48.20	21.30	36.30	49.70	54.54	50.00	41.26
LYNX 8B	60.25	74.48	85.70*	68.45	44.88	56.71	44.36
ANAH-v2	57.00	59.98	81.21*	57.33	57.44	51.03	26.68
FAVA-Model	57.82	73.37	62.16	50.42	44.62	53.18	41.64
HHEM-2.1-Open	-	-	-	37.86	-	55.68*	40.86*
GPT-4o	73.55	81.30	86.33	71.43	51.89	56.29*	40.75*
GPT-4o mini	71.71	81.18	84.04	69.80	39.20	52.13	35.29
DeepSeek-V3	77.85	**81.81**	50.73	72.09	62.98	52.94	32.79
HAD-7B	82.24	78.41	83.63	81.54	63.68	44.63	44.30
HAD-8B	**84.55**	75.40	81.40	76.55	64.16	55.46	55.55
HAD-14B	82.33	79.94	86.65	81.36	**66.82**	51.85	45.45
HAD-14B-Binary	82.16	76.55	**92.37**	**84.63**	60.18	**57.74**	**54.97**

Table 2: Model performance on the out-of-domain test sets. Note that the results with '*' is copied from the original papers. In the ANAH-v2 model test, we classify samples with the model output of "unverifiable" or "nofact" as 0.5 non-hallucinated and 0.5 hallucinated samples (highlighted in gray).

Method	FRE	FIE	FE	FA
HAD-14B	35.90	26.87	81.48	70.48
+knowledge	37.84	52.17	97.99	87.88

Table 3: F1-score of in-domain test results with knowledge augmentation.

6 Related Work

6.1 Hallucination Detection Model

Training a language model to detect hallucinations is a cost-effective approach to some extent. Several recent works have made significant strides in this area by proposing hallucination detection models and benchmarks. For instance, FAVABench (Mishra et al., 2024) is a dataset designed to evaluate hallucinations in large language models. This work classifies six types of hallucinations and proposes FAVA, a retrieval-augmented model trained on Llama2-Chat 7B to detect and edit hallucinations. It is the closest to ours, but is limited to factual hallucinations with mandatory retrieval and focuses solely on fact-seeking tasks. Several other works, including RAGTruth (Niu et al., 2023), FactCHD (Chen et al., 2024) and ANAH-v2 (Gu et al., 2024) adopt the same workflow that propose a hallucination dataset and then train an accompanying hallucination detection model. LYNX (Ravi et al., 2024), a series of models, is designed for large-scale hallucination detection in contextual question-answering tasks within an RAG framework. These models, while performing excellently on specific tasks or domains, face significant challenges when dealing with diverse data from various tasks or types of hallucinations. In contrast, our HAD is designed to handle a broader range of tasks and hallucination types, offering greater versatility and applicability across a variety of use cases.

6.2 Taxonomy of Hallucination

To better understand and address hallucination in natural language generation, establishing a detailed taxonomy is essential. Several surveys (Ji et al., 2023; Ye et al., 2023) divide hallucinations into two parts: intrinsic hallucinations, where the output contradicts the source content, and extrinsic hallucinations, where the output cannot be verified by the source content. In another sur-

vey (Zhang et al., 2025), hallucinations are categorized into three types based on where the outputs conflict: input-conflicting, context-conflicting, and fact-conflicting. Though these taxonomies are valuable for further investigations, they lack a fine-grained taxonomy of hallucinations. We inherit and further expand the taxonomy from (Huang et al., 2023), providing a more detailed analysis of the hallucination problem.

6.3 Hallucination Benchmark

As hallucination has become a significant area of investigation into large language models (LLMs), numerous benchmarks have been developed. For example, HalluQA (Cheng et al., 2023) contains three types of questions (misleading, misleading-hard, and knowledge), aiming to evaluate hallucination in Chinese LLMs. Moreover, some benchmarks are developed to assess the ability of LLMs to detect hallucinations. HaluEval (Li et al., 2023) consists of 5,000 user queries and 30,000 task-specific samples generated from automatic generation and human annotation. FELM (Chen et al., 2023) can be used to assess LLMs' factuality in five domains, including world knowledge, science and technology, mathematics, writing and recommendation, and reasoning. Although these studies are compelling, there is still a need for a large-scale, multi-task hallucination dataset for training. In our study, we aim to provide a dataset covering multiple NLG tasks and hallucination categories.

7 Conclusion

In conclusion, we propose a fine-grained hallucination taxonomy with 11 fine-grained categories and construct a large dataset for training hallucination detection models covering multiple NLG tasks. Additionally, we manually annotate a test set called HADTest. Based on the hallucination taxonomy and data, we introduce HAD models for hallucination detection. Beyond simply detecting whether there are hallucinations in the text, HAD models can also classify fine-grained hallucination types, locate their spans, and correct them. Our models outperform existing general-purpose models as well as hallucination detection models, offering improved functionality, detection accuracy, and applicability across different NLG tasks.

8 Limitations and Future Work

Although our proposed HAD models achieve good performance on both in-domain and out-of-domain benchmarks, several limitations remain. First, the current formulation is restricted to detecting single-span hallucinations, whereas real-world outputs often contain multiple or overlapping erroneous spans, as well as different hallucination types within the same task output. A possible remedy is to first decompose outputs into smaller units before applying the HAD model. Future work could extend this framework to jointly detect multiple, potentially overlapping hallucination spans in an end-to-end manner, without relying on explicit unit decomposition.

Second, the training pipeline depends heavily on synthetic hallucination data, which, despite its scalability, cannot fully capture the diversity and complexity of naturally occurring cases. However, annotating natural hallucinations is costly and difficult to control in terms of distribution, making this trade-off between synthetic and natural data persistent and difficult to resolve. A valuable next step would be incorporating limited natural hallucination annotations via semi-supervised learning.

On the methodological level, the functionality of our model is primarily driven by the composition and diversity of the data. We have not fully explored how to coordinate between multiple tasks and various types of hallucinations to achieve optimal results. How to design mechanisms that better coordinate across tasks and hallucination types is worth further investigation.

Acknowledgments

This work was supported by Beijing Natural Science Foundation (L253001), Key Laboratory of Science, Technology and Standard in Press Industry (Key Laboratory of Intelligent Press Media Technology) and National Engineering Research Center of New Electronic Publishing Technologies. We appreciate the anonymous reviewers for their helpful comments. Xiaojun Wan is the contact author.

References

Forrest Bao, Miaoran Li, Rogger Luo, and Ofer Mendelevitch. 2024a. HHEM-2.1-Open.

Forrest Sheng Bao, Miaoran Li, Renyi Qu, Ge Luo, Erana Wan, Yujia Tang, Weisi Fan, Manveer Singh

Tamber, Suleman Kazi, Vivek Sourabh, and 1 others. 2024b. Faithbench: A diverse hallucination benchmark for summarization by modern llms. *arXiv preprint arXiv:2410.13210*.

Shiqi Chen, Yiran Zhao, Jinghan Zhang, I-Chun Chern, Siyang Gao, Pengfei Liu, and Junxian He. 2023. Felm: Benchmarking factuality evaluation of large language models. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Xiang Chen, Duanzheng Song, Honghao Gui, Chengxi Wang, Ningyu Zhang, Jiang Yong, Fei Huang, Chengfei Lv, Dan Zhang, and Huajun Chen. 2024. Factchd: Benchmarking fact-conflicting hallucination detection. In *Proceedings of the 33rd International Joint Conference on Artificial Intelligence*.

Qinyuan Cheng, Tianxiang Sun, Wenwei Zhang, Siyin Wang, Xiangyang Liu, Mozhi Zhang, Junliang He, Mianqiu Huang, Zhangyue Yin, Kai Chen, and 1 others. 2023. Evaluating hallucinations in chinese large language models. *arXiv preprint arXiv:2310.03368*.

Zina Chkirbene, Ridha Hamila, Ala Gouissem, and Unal Devrim. 2024. Large language models (llm) in industry: A survey of applications, challenges, and trends. In *2024 IEEE 21st International Conference on Smart Communities: Improving Quality of Life using AI, Robotics and IoT (HONET)*, pages 229–234. IEEE.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

David Dale, Elena Voita, Loïc Barrault, and Marta R Costa-Jussà. 2022. Detecting and mitigating hallucinations in machine translation: Model internal workings alone do well, sentence similarity even better. *arXiv preprint arXiv:2212.08597*.

Nouha Dziri, Ehsan Kamalloo, Sivan Milton, Osmar Zaiane, Mo Yu, Edoardo M. Ponti, and Siva Reddy. 2022. FaithDial: A faithful benchmark for information-seeking dialogue. *Transactions of the Association for Computational Linguistics*, 10:1473–1490.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

Yuzhe Gu, Ziwei Ji, Wenwei Zhang, Chengqi Lyu, Dahua Lin, and Kai Chen. 2024. Anah-v2: Scaling analytical hallucination annotation of large language models. *arXiv preprint arXiv:2407.04693*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and 1 others. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning.

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

Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. HaluEval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6449–6464, Singapore. Association for Computational Linguistics.

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.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore. Association for Computational Linguistics.

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, Singapore. Association for Computational Linguistics.

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*.

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*.

Kellin Pelrine, Mohammad Taufeeque, Michał Zając, Euan McLean, and Adam Gleave. 2023. Exploiting novel gpt-4 apis. *arXiv preprint arXiv:2312.14302*.

Vikas Raunak, Arul Menezes, and Marcin Junczys-Dowmunt. 2021. The curious case of hallucinations in neural machine translation. *arXiv preprint arXiv:2104.06683*.

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*.

Vipula Rawte, Swagata Chakraborty, Agnibh Pathak, Anubhav Sarkar, S.M Towhidul Islam Tonmoy, Aman Chadha, Amit Sheth, and Amitava Das. 2023. The troubling emergence of hallucination in large language models - an extensive definition, quantification, and prescriptive remediations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2541–2573, Singapore. Association for Computational Linguistics.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, and 16 others. 2022. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Hongbin Ye, Tong Liu, Aijia Zhang, Wei Hua, and Weiqiang Jia. 2023. Cognitive mirage: A review of hallucinations in large language models. *arXiv preprint arXiv:2309.06794*.

Jifan Yu, Xiaozhi Wang, Shangqing Tu, Shulin Cao, Daniel Zhang-Li, Xin Lv, Hao Peng, Zijun Yao, Xiaohan Zhang, Hanming Li, Chunyang Li, Zheyuan Zhang, Yushi Bai, Yantao Liu, Amy Xin, Nianyi Lin, Kaifeng Yun, Linlu Gong, Jianhui Chen, and 16 others. 2024. Kola: Carefully benchmarking world knowledge of large language models. *Preprint*, arXiv:2306.09296.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Chen Xu, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2025. Siren's song in the ai ocean: A survey on hallucination in large language models. *Preprint*, arXiv:2309.01219.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.

A Supplementary Materials for Hallucination Taxonomy

We build our hallucination taxonomy based on prior works about LLMs, as shown in Table 8, each category can be find similar concepts in related research. This mapping provide important supporting for the coverage and rationality of our taxonomy. To help better understand the meaning of each hallucination type, we provide examples in Table 7.

B Prompt Templates and Guidelines

Here, we present our prompt templates and guidelines for different stages to enhance the reproducibility of our work.

During the data construction stage, we first inject hallucinations (Table 10) and then filter out low-quality data (Table 11). The guidelines (Table 12) are applied both for the automatic filtering of training data and for the manual annotation of HADTest.

In the training and evaluation stage, we employ the same prompt and response template (Table 13) for HAD-7B, HAD-8B, and HAD-14B. Table 14 presents the template for HAD-14B-Binary, which differs slightly from the previous one. When evaluating baseline large language models on HADTest, we adopt a few-shot prompting strategy (Table 15).

C Implementation of Baselines

When testing base models on our in-domain HADTest dataset, we provide hallucination-type descriptions and fixed 2-shot examples in the prompt (Table 15). One example is correct and another is hallucinated. For out-of-domain test datasets, we adopt the same prompts as specified in the original papers corresponding to these datasets, with the temperature parameter set to 1.

For SelfCheckGPT (Manakul et al., 2023), we use `Qwen2.5-14B-Instruct` as the passage generation model and sample 5 responses for each prompt. The passage-level score is the average of sentence-level scores. Data samples with passage-level scores less than 0.5 are labeled as "Hallucination". Besides, the four hallucination detection models (LYNX 8B (Ravi et al., 2024), ANAH-

v2 (Gu et al., 2024), FAVA-Model (Mishra et al., 2024), HHEM-2.1-Open(Bao et al., 2024a)) share a consistent input format consisting of three parts: *context*, *query*, and *response*. For summarization tasks, we place the input documents in the context part. For other test sets, we simply place the task input in the query part and leave the context part empty.

D Supplementary Experimental Results

D.1 Supplementary In-Domain Test Results

Since several baselines do not support fine-grained classification on HADTest, we compare these baselines with our models using the binary classification task and report their accuracies in Table 5. Detailed evaluation results of HAD-14B on the HADTest dataset are presented in Table 4. Furthermore, Table 9 shows three examples of HAD-14B's predictions, demonstrating the model's ability to detect and correct different types of hallucinations.

Category	Precision	Recall	F1-Score	Size
TTI	49.30	87.50	63.06	40
TRI	50.56	44.12	47.12	102
CwIC	81.90	63.33	71.43	150
BI	86.18	83.46	84.80	127
IO	74.42	67.37	70.72	95
CwOC	78.26	71.05	74.48	152
SI	93.89	85.79	89.66	197
FRE	41.18	31.82	35.90	66
FIE	69.23	16.67	26.87	54
FE	90.16	74.32	81.48	74
FA	97.37	55.22	70.48	67
No Hallu	86.76	86.30	86.53	1124
overall	74.93	63.91	66.88	2248

Table 4: Detailed evaluation results of HAD-14B on HADTest. The overall metrics are macro average of all the categories.

Model	Acc
SelfCheckGPT	45.73
LYNX 8B	63.83
ANAH-v2	54.76
FAVA-Model	59.43
HAD-7B	87.46
HAD-8B	86.12
HAD-14B	**89.10**
HAD-14B-Binary	87.77

Table 5: Evaluation of binary hallucination classification on HADTest.

D.2 Ablation Study

The hallucination verification step in Section 3.2 aims to improve data quality. To evaluate its impact, we randomly sample an equal number of data points with the same distribution from the raw hallucination data and train a separate model for comparison. We evaluate it with the fine-grained classification task and the macro F1 on our test set is 75.65%, which is lower than the HAD-14B's accuracy of 76.29%.

D.3 Error Analysis

We use the results of HAD-14B for this section and analyze the in-domain test results across different types. The confusion matrix is shown in Figure 3. The F1 scores of *Factual Recall Error* and *Factual Inference Error* are relatively low (35.90 and 26.87, respectively) because these errors are often mistakenly labeled as "no hallucination." These failures are mainly due to insufficient background knowledge, particularly in smaller models. Besides, there's noticeable confusion between similar categories, such as *Contradiction with Input Content* and *Contradiction within Output Content*, as well as between *Factual Inference Error* and *Factual Recall Error*.

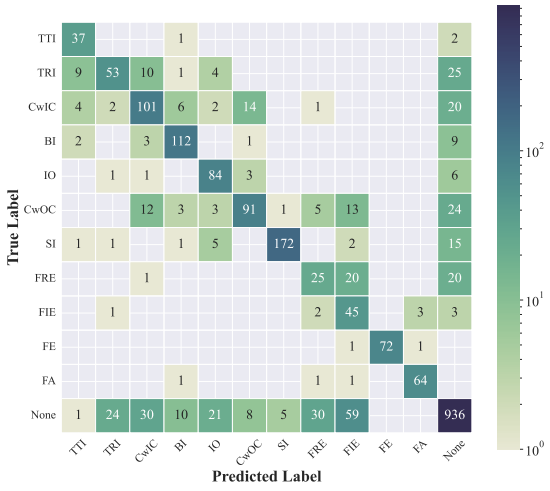

Figure 3: Confusion matrix of the test results generated by HAD-14B.

In the HaluEval-general dataset, false negatives are mainly classified as factual recall errors or task requirement inconsistencies. Additionally, this dataset has a wider range of task types and requirements and contains responses generated by real LLMs, which makes hallucination detection more challenging, resulting in more false positives.

In summary, the error analysis highlights that HAD-14B still struggles with fact contradiction hallucinations, which are largely attributed to a lack of background knowledge, especially in smaller models. The analysis also suggests areas for improvement, such as better handling of diverse tasks and enhancing the model's ability to distinguish between different types of hallucinations.

E Details on Human Evaluation

In this section, we present more details about the human evaluation process for the model-generated corrections described in Section 4.3. The annotation is performed by a single volunteer annotator. We provide a comprehensive annotation guideline that defines the task, data format, and evaluation protocol for assessing the quality of automatically generated hallucination corrections. Each sample includes the original input, the hallucinated output, the ground-truth hallucination span and correction, as well as the model's predicted hallucination span and the correction to be evaluated. The annotator is required to judge the model's correction along six dimensions, each rated on a predefined categorical scale:

- overall acceptability: whether the corrected text is a usable replacement;
- factual accuracy: whether the correction introduces factually correct information, with a special option for non-factual hallucinations;
- style preservation: how well the original tone and register are maintained;
- coherence and fluency: internal grammatical and logical quality;
- introduction of new errors: whether the correction inadvertently adds errors to previously correct content;
- overcorrection: whether the modification unnecessarily alters content beyond the hallucinated span.

When new errors or overcorrections are identified, annotators are required to document them in a free-text notes field. The guideline further instructs annotators to focus exclusively on correction quality rather than hallucination detection accuracy, to compare the predicted correction against both the original hallucinated output and the ground-truth correction, and to use conservative labels and provide notes in ambiguous cases. Table 6 presents the detailed results.

Dimension	Label	Count
Acceptability	Acceptable	96 (87.3%)
	Partially Acceptable	6 (5.5%)
	Unacceptable	8 (7.3%)
Factual Accuracy	Accurate	34 (30.9%)
	Mostly Accurate	4 (3.6%)
	Inaccurate	2 (1.8%)
	N/A	70 (63.6%)
Style Preservation	Well Preserved	96 (87.3%)
	N/A	14 (12.7%)
Coherence	Coherent	109 (99.1%)
	Incoherent	1 (0.9%)
Secondary Error	No	106 (96.4%)
	Yes	4 (3.6%)
Overcorrection	No	109 (99.1%)
	Yes	1 (0.9%)

Table 6: Label distribution of human evaluation across six quality dimensions for 110 model-generated corrections.

F AI Usage Disclosure

In this work, we utilize generative AI to assist with data processing and to refine our manuscript. Throughout the use of AI tools, we carefully review and revise the generated content to ensure the accuracy and reliability of our work.

G Ethical Considerations

We carefully consider the ethical aspects of our work. We assess that the present study carries minimal risk, primarily because it neither involves sensitive data nor engages human subjects. All hallucinatory content in our paper and datasets is explicitly annotated to guarantee the transparent and responsible utilization of these resources.

Our research exclusively employs publicly accessible datasets and pretrained models, and our application of these resources strictly adheres to the licenses. The data utilized in this work contains no personally identifiable information or other forms of sensitive content; we operate under the presumption that the original providers of these datasets have conducted appropriate anonymization and content filtering procedures.

Additionally, the artifacts (including the constructed datasets and associated models) developed through this study will be released, with their distribution terms remaining consistent with the licensing agreements of the used resources.

Category	Input	Output
Task Type Inconsistency	In this task, you will be given a list of numbers. The goal is to sum all the numbers in the list. [180.44, 159.979, 32.8, 227.334, 32.8]	To remove duplicates from the list, we need to identify numbers that appear more than once and remove them. "32.8" appears twice in the list. Here is the list with duplicatest removed: [180.44, 159.979, 32.8, 227.334].
Task Requirement Inconsistency	Classify the following statement into a high level semantic category: "The stock markets are surging"	Market Trends/Stock Market Performance
Contradiction with Input Content	Describe the concept of Imposter Syndrome.	Imposters Syndrome is a psychological phenomenon in which people feel a persistent fear of being exposed as a fraud.
Baseless Information	Given a sentence and an input, correct the grammar and spelling mistakes in the sentence. "The dog barked loudley at it's foe."	The dog barked loudly at its foe, a squirrel perched confidently on a nearby branch.
Information Omission	Given a sentence and an input, correct the grammar and spelling mistakes in the sentence. "The dog barked loudley at it's foe."	The dog barked loudly.
Contradiction within Output Content	Create a signature drink.	I suggest a White Grapefruit Mule. This refreshing summer drink is made with fresh white grapefruit juice, vodka, ginger beer, and a splash of lime juice. Enjoy your Red Grapefruit Mule!
Structure Incoherence	Create a recipe for mashed potato pancakes.	Mashed Potato Pancakes Recipe Ingredients:- 1 cup all-purpose flour 1/4 teaspoon ground cloves- 1/4 cup milk- 1/4 cup milk- 1/4 cup milk
Factual Recall Error	Where is Messi playing now?	Barcelona.
Factual Inference Error	Why do older cartoons always show a lighter color of an object that's about to move?	It's discussed in detail here. The most efficient way to animate older cartoons, images are rendered as lines on a black background digitally, then coloured by computer. The colors were deliberately faded in the areas of motion to give the illusion of speed.
Fabricated Entity	Why do warmer climates have turquoise, clear seas and white beaches? Why is my sea so gross in comparison?	If you are imagining tropical islands, the answer lies in the Sapphire Lattice. This underwater crystal network, unique to warmer climates, refracts sunlight in a way that turns the water turquoise and enhances the whiteness of the sand.
Fictional Attribution	Who does Mr. Jones represent in Animal Farm?.	Tsar Nicholas II, who was also known for his involvement in the secret negotiations with the British government to establish a new form of monarchy in Eastern Europe."

Table 7: Examples of different hallucination categories. The red text indicates the hallucination span.

Type Name	Related Work	Related Term
Task Type Inconsistency	(Huang et al., 2023) (Pelrine et al., 2023)	Instruction Inconsistency Incorrect Summarization
Task Requirement Inconsistency	(Zhou et al., 2023)	Instruction Following Failure
Contradiction with Input Content	(Huang et al., 2023) (Zhang et al., 2025)	Context Inconsistency Input-Conflicting Hallucination
Baseless Information	(Niu et al., 2023) (Raunak et al., 2021)	Evident/Subtle Introduction of Baseless Information Detached Hallucinations
Information Omission	(Dale et al., 2022)	Undertranslation
Contradiction within Output Content	(Zhang et al., 2025) (Huang et al., 2023)	Context-Conflicting Hallucination Logical Inconsistency
Structural Incoherence	(Raunak et al., 2021)	Oscillatory Hallucinations
Factual Recall Error	(Yu et al., 2024)	Failure on Knowledge Memorization
Factual Inference Error	(Yu et al., 2024) (Rawte et al., 2023)	Failure on Knowledge Applying Time Wrap
Fabricated Entity	(Rawte et al., 2023) (Mishra et al., 2024)	Generated Golem Invented Information
Fictional Attribution	(Rawte et al., 2023)	Virtual Voice

Table 8: Mapping of Hallucination Taxonomy Categories to Related Works and Terminologies

Task Information
Task Input:
Rewrite the passage from the perspective of a different character:
Sophie was exhausted after a long day at school.
Task Output:
Tom found Sophie exhausted.

Detection Result
Hallucination Type:
Information Omission
Hallucination Span:
after a long day at school.
Correction:
Tom found Sophie exhausted after a long day at school.

Task Information
Task Input:
List three types of pre-trained models for natural language processing.
Task Output:
Three types of pre-trained models for natural language processing are BERT, GPT-2, and TF-IDF.

Detection Result
Hallucination Type:
Factual Recall Error
Hallucination Span:
TF-IDF
Correction:
Three types of pre-trained models for natural language processing are BERT, GPT-2, and RoBERTa.

Task Information
Task Input:
Given two noun phrases (arguments) and relationship between them, form a sentence that expresses theses arguments with the given relationship.
Relationship: 'be locate in', Argument/Subject 1: 'fort lauderdale', Argument/Subject 2: 'broward county'
Task Output:
201 Southwest 5th Avenue Fort Lauderdale from Broward County Convention Center , which is located at the northern end of Port Everglades , the world 's second largest cruise port, largerst cruise port.

Detection Result
Hallucination Type:
Structural Incoherence
Hallucination Span:
largerst cruise port.
Correction:
201 Southwest 5th Avenue Fort Lauderdale from Broward County Convention Center , which is located at the northern end of Port Everglades , the world 's second largest cruise port .

Table 9: Examples of the detection results from HAD-14B.

Instruction
Given a pair of task input and output, your objective is to create an error data by intentionally modifying the given task output. Inject the error exactly as the error type description, without introducing any other modifications. The error should be restricted to a single error span, which is the part of the task output that you modify. Do not include any other errors or changes outside of the designated error span. Provide the modified output and the error span in your response.

Error Type Description
{hallucination_description}

Example
Task Input:
{eg_task_input}
Task Output:
{eg_task_output}
Modified Output:
{eg_modified_output}
Error Span:
{eg_error_span}

{more_examples}

Example
Task Input:
{task_input}
Task Output:
{task_output}

Table 10: Prompt template for hallucination injection in dataset construction stage.

Instruction
Given a task input, a task output containing an error, and a specified span that represents the erroneous part, your goal is to evaluate whether the task output and specified span correspond to the specified error type, based on the provided criteria. For each criterion, provide an analysis that explains how the task output and specified span either satisfy or fail to meet it. Finally, aggregate all the analysis carefully, and conclude with "Conclusion: Yes" if all criteria are met, or "Conclusion: No" if they are not.

Error Type Description
{error_type_description}

Criteria
{error_type_criteria}

Example
Task Input:
{task_input}

Task Output:
{hallucinated_output}

Specified Span:
{hallucinated_span}

Your Judgement

Table 11: Prompt template for automatic hallucination filtering in dataset construction stage.

General:
The task output contains an error in the specified span.
There are no other errors in the task output except for the specified span, which could encompass the entire task output.

Task Type Inconsistency:
The task input specifies one task type, but the task output corresponds to a different task type.
The error should lie in the mismatch of task type, not in the failure to meet specific task constraints.

Task Requirement Inconsistency:
The task input contains specific requirements, such as constraints on length, format, tone, or wording.
The error is limited to a failure to meet these specific requirements. The task output should align with both the task input (excluding specific requirements) and general world knowledge.

Contradiction with Input Content:
The error should involve a contradiction between the task output and the content provided in task input.
The error can be refuted by task input, without requiring additional external information or factual knowledge.
The task output should maintain coherence within itself.

Baseless Information:
The task requires that the correct output should be directly based on the input information, without introducing any new or unsupported information.
The error should introduce information not present in the task input.
The task output must not contain information that conflicts the task input.

Information Omission:
The task output should omit necessary information, resulting in an incomplete or incorrect response.
The information contained within the specified span should be included in the task input but excluded from the task output.
The task output should maintain coherence within itself.

Contradiction within Output Content:
The error occurs within the output itself, where two or more parts of the output contradict each other.
The task output should be consistent with both the task input and general world knowledge.

Structural Incoherence:
The error should pertain to the structure of the output, such as improper conjunctions, incomplete texts, or meaningless repetition.
The information provided in the task output is not necessarily incorrect, but the structure hinders clarity or coherence.

Factual Recall Hallucination:
The task requires factual accuracy based on real world knowledge.
The error should be limited to a single atomic fact.
The error should not introduce any newly fabricated entities or events.
The task output should maintain internal coherence and consistency with the task input.

Factual Inference Hallucination:
The task requires factual accuracy based on real world knowledge.
The error should involve multiple facts that go beyond a single atomic fact (like a single entity or relationship)).
The error should not introduce any newly fabricated entities or events.
The task output should maintain internal coherence and consistency with the task input.

Fabricated Entity:
The task requires factual accuracy based on real world knowledge.
The error introduces a completely fabricated entity that is not part of established world knowledge.
The task output should maintain internal coherence and consistency with the task input.

Fictional Attribution:
The task requires factual accuracy based on real world knowledge.
The task output should not introduce any newly fabricated entities.
The error should not directly conflict with established real-world knowledge, but it can be refuted through careful analysis and reasoning.
The task output should maintain internal coherence and consistency with the task input.

Table 12: Criterion for automatic hallucination filtering and manual annotation.

Instruction
Given a pair of task input and task output, your goal is to detect whether the task output contains any hallucination.
If a hallucination is present, specify the type of hallucination, identify the hallucination span, and provide the correct version of the output.

Example
Task Input:
{task_input}

Task Output:
{task_output}

Your Detection

Hallucination Type:
{hallucination_type}

Hallucination Span:
{hallucination_span}

Correction:
{correction}

Table 13: Prompt and response templates for HAD-14B and HAD-7B.

Instruction
Given a pair of task input and task output, your goal is to detect whether the task output contains any hallucination.
If a hallucination is present, identify the hallucination span and provide the correct version of the output.

Example
Task Input:
{task_input}

Task Output:
{task_output}

Your Detection

Hallucination Label:
{hallucination_type}

Hallucination Span:
{hallucination_span}

Correction:
{correction}

Table 14: Prompt and response templates for HAD-14B-Binary.

Instruction
Given a pair of task input and task output, your goal is to detect whether the task output contains any hallucination.
If a hallucination is present, specify the type of hallucination based on the type description, identify the hallucination span, and provide the correct version of the output.

Hallucination Type Description
Task Type Inconsistency: The generated output represents a different type of task than what was specified in the instruction. This does not include deviations within the same task type, such as violations of detailed requirements or specifications.
Task Requirement Inconsistency: The generated output does not align with the task requirements outlined in the instruction, including key aspects such as the expected format, length, subject matter, or tone. Note that this error stems from not following the task requirement, rather than from inconsistency with the input content.
Contradiction with Input Content: The generated output contradicts with the provided input content, presenting information or statements that are incompatible with the context given. This may result from a failure to accurately recall the input content, or from misunderstandings and confusion about the information provided.
Baseless Information: The generated output contains baseless information that are not supported by the input context, whereas the task requires the model to generate output that strictly adheres to the information provided in the input. Note that tasks seeking new information do not encounter this issue.
Information Omission: The generated output fails to include certain details or information present in the input, whereas the task requires the modelś output to fully and accurately capture all the information provided in the input context.
Contradiction within Output Content: The generated content contains internal inconsistencies where statements directly oppose each other, or where the reasoning is logically flawed.
Structural Incoherence: The generated output contains redundant or repetitive statements that do not enhance the clarity or value of the content, or when the output is incomplete or disjointed. This does not apply instances where the incoherence is used purposefully for stylistic effect or rhetorical emphasis.
Factual Recall Error: The generated text contains incorrect atomic facts due to the modelś inability to accurately recall or access relevant knowledge. Note that the inaccuracy is limited to a single atomic fact, rather than multiple facts.
Factual Inference Error: The generated content contains incomplete or misinterpreted facts. Common phenomena include confusion between different time periods, individuals, or events; omissions of critical conditions or contextual information; and errors in the logical sequence of events or processes. As a result, the modelś reasoning appears to be based on seemingly factual information, but it ultimately leads to an erroneous or unreliable output.
Fabricated Entity: The generated content contains entirely new and fabricated entities that do not exist in the real world, including invented concepts, names, or objects that have no basis in reality or prior knowledge.
Fictional Attribution: The generated content fabricates information about real entities, including unverified or fabricated claims, statements, or quotes, which cannot be supported or directly refuted by established facts or reliable sources. Unlike the "fabricated entity" type, this error does not introduce entirely new entities.

Example
{example_1}

Example
{example_2}

Example
Task Input:
{task_input}

Task Output:
{task_output}

Table 15: Prompt template for baseline large language models.