Developing a Reliable, Fast, General-Purpose Hallucination Detection and Mitigation Service

Song Wang, Xun Wang, Jie Mei, Yujia Xie,
Sean Muarray, **Zhang Li**, **Lingfeng Wu**, **Si-Qing Chen**, **Wayne Xiong**
Microsoft
{sonwang, xunwang, jimei, yujiaxie,}@microsoft.com
{murraysean, z67li, lingfw, sqchen, weixi}@microsoft.com

Abstract

Hallucination, a phenomenon where large language models (LLMs) produce output that is factually incorrect or unrelated to the input, is a major challenge for LLM applications that require accuracy and dependability. In this paper, we introduce a reliable and high-speed production system aimed at detecting and rectifying the hallucination issue within LLMs. Our system encompasses named entity recognition (NER), natural language inference (NLI), span-based detection (SBD), and an intricate decision tree-based process to reliably detect a wide range of hallucinations in LLM responses. Furthermore, we have crafted a rewriting mechanism that maintains an optimal mix of precision, response time, and cost-effectiveness. We detail the core elements of our framework and underscore the paramount challenges tied to response time, availability, and performance metrics, which are crucial for real-world deployment of these technologies. Our extensive evaluation, utilizing offline data and live production traffic, confirms the efficacy of our proposed framework and service.

1 Introduction

In the rapidly evolving landscape of natural language processing (NLP), large language models (LLMs) have marked a significant leap forward, unlocking new horizons of capabilities and potentials. However, alongside their remarkable advancements, LLMs bring forth substantial challenges, with "hallucination" standing out as a particularly problematic issue. Hallucination in this context refers to instances when an LLM produces output that is either factually incorrect or not anchored in the supplied input, thus compromising the model's reliability and the credibility of its applications. Therefore, the importance of confronting and mitigating hallucinations in LLM deployments cannot be overstated.

Detecting and mitigating hallucinations present tough challenges, actively explored in recent research as evidenced by several survey papers (Ji et al., 2023; Huang et al., 2023; Tonmoy et al., 2024). There are also different levels of hallucinations spanning from minor inconsistencies to blatant fabrications, and they can have different effects for different applications and users. Against this backdrop, our work delves into the nuances of hallucinations within LLMs, placing a special emphasis on **intrinsic hallucinations** i.e. errors that can be checked against reference inputs.

Developing a **general-purpose, fast and accurate** hallucination detection and mitigation service is an extremely difficult task given the existing state-of-the-art technologies. To this end, we present a pragmatic solution as shown in Figure 1, which includes three modules: **multi-source detection**, **iterative rewriting** and **multi-source verification**. We will discuss the components in details in Section 4.

Our contributions are as follows. First, we present a novel detection system capable of detecting different types of hallucinations with high accuracy. The system operates in real-time (low latency) and is suitable for large-scale applications (low cost). This approach leverages multiple hallucination detection methods—including named entity recognition (NER), natural language inference (NLI), and span-based detection—and ensembles multiple AI feedbacks using Gradient Boosting Decision Trees (GBDT).

Second, we propose a rewriting system for hallucination removal utilizing large language models (LLMs). After testing various strategies, we developed an effective rewriting solution that balances quality and latency.

Third, we conducted comprehensive experiments, analyses, and evaluations, demonstrating that our methods are effective and providing insights valuable for other researchers and industry
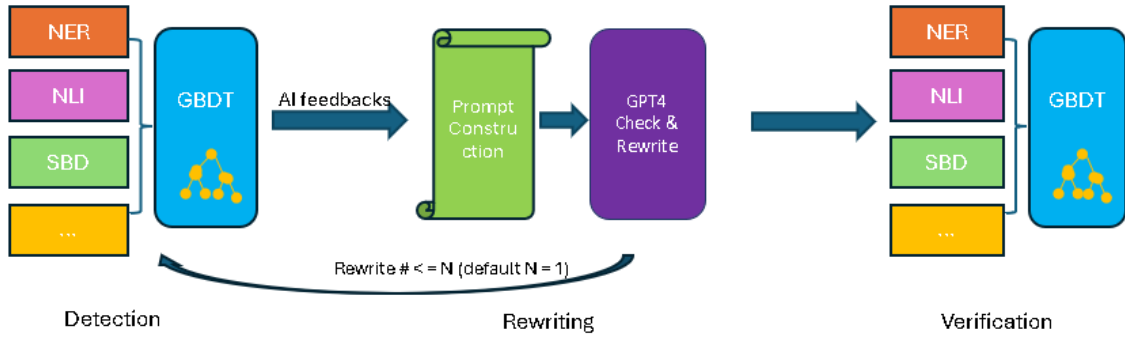
Figure 1: End-2-end hallucination detection and mitigation system

2 Related Work

Hallucination Taxonomy A widely-adopted classification of hallucination is the intrinsic-extrinsic dichotomy(Dziri et al., 2021; Huang et al., 2021). Intrinsic hallucination occurs when LLM outputs contradict the provided input, such as prompts. Conversely, extrinsic hallucination occurs when LLM outputs cannot be verified by the information in the input. Recently, reasearchers have proposed more fine-grained taxonomies (Pagnoni et al., 2021; Mishra et al., 2024).

We largely followed the categories in (Pagnoni et al., 2021) with modifications that reflect the nature and causes of hallucinations in LLM outputs and we also developed a guidelines based on this taxonomy for annotators.

Hallucination Detection Conventional methods of detecting hallucination can be classified into two types: token-based and sentence-based. The former aims to find hallucinated tokens while the latter is to identify the sentences with hallucinations. Various methods have been developed for identifying hallucinations and most of them leverage the pre-trained LLMs fine tuned on task-specific data (Liu et al., 2021; Dziri et al., 2021; Cao et al., 2021; Zha et al., 2023). More recently, LLM with prompt-based methods are also widely used (Manakul et al., 2023; Lei et al., 2023). Though the prompt-based methods require little to no tuning and have competitive performance, they tend to have higher cost and higher latency.

In our study, we've harnessed the strengths of both methods by employing LLM-based detection

for labeling data and creating an ensemble of tailored traditional models such as NER, NLI, and span-based sequence labeling. Our designed detection service conducts a detailed examination of the input text's semantic and syntactic attributes, allowing it to detect various kinds of hallucinations across different granularities and categories.

Hallucination Mitigation Hallucination mitigation is to correct the identified hallucinations in the generated responses by LLMs. There are many other ways to reduce hallucination during post-generation. For example RARR (Thorne and Vlachos, 2021) trained a T5 model using retrieved evidence to generate corrected responses. More recently, researchers have been leveraging LLMs to better utilize hallucination feedback and generate corrections (Mündler et al., 2024; Dhuliawala et al., 2023; Lei et al., 2023).

In our work, our rewriter is also LLM-based, leveraging LLM's self refinement through feedback and reasoning. The key difference is that we have to take into consideration the cost and latency, which demands fewer output tokens, while ensuring the mitigation performance.

3 Hallucination Taxonomy and GPT4-based detection

3.1 Hallucination Taxonomy

We started with developing a hallucination taxonomy by manually analyzing various hallucinated model outputs, mainly from some internal summarization systems, which is based on a state-of-the-art encoder-decoder model about 1B parameters.

We randomly collected 500 samples from production systems and benchmark systems, including ChatGPT, and manually identified the **34** hallucinated outputs (two of the authors). In Table 1, we

Error Category	Description	Examples	
		Intrinsic	**Extrinsic**
Semantic Frame Errors:			
Entity	Error in the primary argument (or attributes) of the predicate	The **Juvae in July 2019 event** will feature a dance orchestra. *Correct: **The Juvae event** will feature a dance orchestra in July 2019.*	We read the story of **Symeon**, a paralyzed ... *Correct: We read the story of a paralyzed ...*
Predicate	Error in the predicate (or its attributes) of the summary	The program **will be** a presentation ... *Correct: The program **was** a presentation*	The questions appear on ... and the answers **are listed on** ... *Info: No information about where are the answers*
Circumstance	Error in the circumstance (time, location) around the predicate	On January 27, **2018**, the Holocaust ... *Correct: Dec. **2018**: The Holocaust Remembrance Day on January 27 ..*	The event will be in **October** ... *Correct: The event will be **next month***
Discourse Errors:			
Coreference	Pronoun with wrong or not antecedent	while **she** is a college student *Info: No information about person gender*	
Discourse Link	Error in how multiple statements are linked	The State Ports Authority has been unable to sell the property. The State Budget and Control Board **subsequently** delegated the responsibility. *Correct: Swap sentences order*	
Pragmatic Errors:			
Interpretation	Error in the general meaning of the message in the summary	**This document is a contract** ... *Correct: **This document is a revision of a contract***	

Table 1: Hallucination taxonomy and examples. We largely followed the categories in (Pagnoni et al., 2021) with modifications that reflect the nature and causes of hallucinations in LLM outputs.
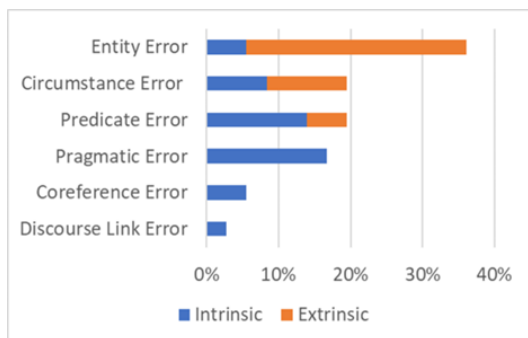

Table 2: Hallucination distribution

list our taxonomy and examples of hallucinations, which show that hallucination is of different types and root causes. Most of them are due to semantic frame errors, but discourse errors and pragmatic errors also exist.

In Table 2, we list the distribution of different types of hallucinations. As stated above, we focus mainly on **intrinsic hallucinations**, which are errors that can be verified from the source document. Although most of the existing hallucination detection solutions are entity-based which treat new entities in the generated summary as hallucinations, from table 2, we found only a small portion, 5% of intrinsic hallucinations are attributable to new entities. This observation motivates us to develop an ensemble-based solution that extends NER-based methods to recall more hallucination errors.

3.2 GPT4-based Detection

We developed a GPT4-based hallucination detection as follows (we cannot share the prompt due to proprietary limitations): First, we transform the LLM outputs so that individual sentences are placed on separate lines. We then instruct GPT-4 to evaluate each sentence by comparing it against the source document and provide reasons if a sentence is determined to be a hallucination. If any sentence within the output is identified as hallucinated, the entire output is considered hallucinated. This approach utilizes the Chain of Thought (CoT) technique, and has been shown to outperform existing methods on multiple datasets; for instance, see the results on SummAC in Table 4. Other studies (Lei et al., 2023; Wei et al., 2024) have also demonstrated the effectiveness of leveraging GPT-4's reasoning capabilities for hallucination detection.

Additionally, we examined discrepancies between human annotation and our GPT-4-based evaluation using 20 model outputs from a benchmark dataset of 1,400 samples (see Table 3). Our analysis indicates that while GPT-4 tends to have higher false positive rates, human annotators often show higher false negative rates. Nonetheless, GPT-4's labeling is comparable to human efforts in overall error counts and can enhance annotator productivity by combining the model's high recall with human precision. However, the study's limited sample size and the specific design of prompts may constrain the generalizability of these findings, indicating a need for further research.

	False Positive	False Negative	Total
Annotators	1	10	11
GPT4	4	5	9

Table 3: Error analysis of the inconsistency of GPT4 and human,

4 Hallucination Detection and Mitigation

In this section, we introduce the hallucination checking method as shown in Figure 1 tailored for intrinsic hallucinations, which employes an ensemble method that leverages multiple techniques—including named entity recognition (NER), natural language inference (NLI), and sequence labeling—to detect inaccuracies. None of these models are LLM-based mainly for two main reasons: First, we aim to achieve real-time detection with high accuracy suitable for large-scale applications. The substantial costs associated with calling LLM models are not feasible. Second, LLMs are not effective at detecting hallucinations in their own outputs. Previous works have shown that LLMs tend to believe their outputs are correct and are difficult to persuade otherwise (Farquhar et al., 2024; Quevedo et al., 2024).

4.1 Hallucination detection methods

NER-based detection Named Entity Recognition (NER) aims to identify and categorize key information (entities) in text. By applying NER analysis to the input data, we can spot possible entities that are present in the LLM outputs but not supported by the source document—that is, hallucinations. We are using a well-known NER service[1] which returns both entity types and their confidence scores. We apply this NER service to detect hallucinations, and Figure 2 shows the common entity types among the hallucinated LLM outputs. Additionally, we conducted NER analysis on benchmark datasets in the target domain to determine the entity types and confidence thresholds for our NER detection implementation.

NLI-based detection In natural language inference (NLI), given two input text snippets—a premise and a hypothesis—the task is to predict their relationship: entailment, contradiction, or neutral. In principle, this aligns with the goal of hallucination detection. However, in most existing NLI datasets (Bowman et al., 2015; Williams
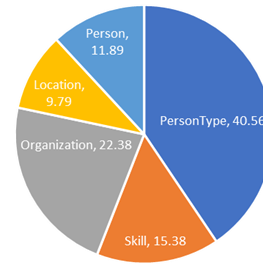
Figure 2: Hallucination distribution over entity types

et al., 2018; Nie et al., 2020; Schuster et al., 2021), the premises and hypotheses are short (one or two sentences). To address this, we included a new document-sentence dataset (Kamoi et al., 2023) and used GPT-4, as described in Section 3.2, to label a diverse set of document-summary pairs from various public and internal sources. Finally, we fine-tuned the pre-trained DeBERTa encoder (He et al., 2021) on these combined datasets. This model detects hallucinations based on the semantic relationship between the document and summary.

Span-based detection NLI provides sentence- or summary-level hallucination detection, while NER is restricted to a predefined set of named entities. To explore a more general fine-grained hallucination detection, we train a token-level hallucination detection model to provide more detailed AI feedback, such as highlighting hallucinated text spans.

Starting with the dataset labeled by GPT-4 for the NLI model, as described in Section 3.2, we further ask GPT-4 to highlight the hallucinated text spans if the text contains hallucinations. As shown in Figure 3, we initiate model training with a pre-trained Replaced Token Detection (RTD) head from DeBERTa (He et al., 2021). We adapt this model using the GPT-4 generated data to determine if a token is part of a hallucinated span. We refer to this model as the Span-Based Detection model (SBD).
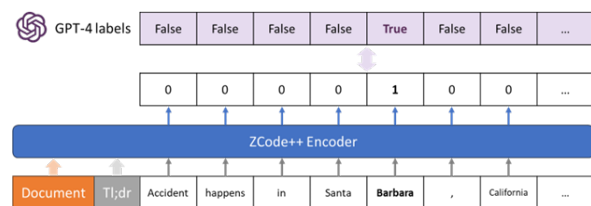

Figure 3: Span-based hallucination detection

4.2 Multi-Source Ensemble

We believe that combining multiple sources is essential to further boost performance while minimizing costs. This approach is also convenient when we want to use a single score to control thresholds, allowing us to prioritize high precision during real implementation.

For this part, we collected about 10,000 training examples, where the labels (hallucinated vs. not hallucinated) are produced by GPT-4 detection, and the features are entities and their confidence scores from the NER, confidence score from NLI, and the confidence score from SBD. We adopt the **Gradient Boosting Decision Tree** to leverage the diverse AI feedback and fine-tune a model using scikit-learn to generate a single numerical value indicating the confidence or likelihood that the text to be checked (i.e., response, summary, or single sentence) is hallucinated.

4.3 GPT4-based rewriting

With AI feedback from different sources, we formulated prompts based on a pre-defined template to guide GPT-4 in correcting hallucinations. We explored two distinct rewriting prompts:

Rewriting prompt v1. This does an exhaustive CoT reasoning or analysis to identify hallucinations in the text. It then does a complete rewriting to correct the hallucination while maintaining the coherence.

Rewriting prompt v2. This prompt reduces the extent of CoT reasoning and opts to skip rewriting if no hallucinations are detected. When rewriting is necessary, it focuses solely on the hallucinated sentences rather than the entire text. This approach ensures only the essential changes are made to the original content.

5 Main Results

5.1 Detection Results

We compare the different detection methods' performance on the following datasets:

Internal benchmark dataset. This internal benchmark dataset has N=1400 examples consisting of 200 representative documents/transcripts x 7 systems of summaries. The ground-truth label is collected by our hired independent data vendors.

Public benchmark dataset: SummAC. This dataset, as detailed in (Laban et al., 2022) has N=1700 examples collected from six datasets focused on summary inconsistency detection, with

ground-truth labels provided by humans in each datasets. We also reference the best results from that paper as baselines.

We evaluated our detection methods against GPT-4 with a focus on both accuracy and latency. As shown in Table 4, NER tends to have lower recall but higher precision in public SummAC (Laban et al., 2022). It's notable that of the three tailored models, the **SBD** method outperforms the rest in all metrics, showcasing the effectiveness of detection at the token level. In Table 5, Compared

Method	Internal benchmark			Public - SummAC		
	Precision	Recall	F1	Precision	Recall	F1
SummAC	-	-	-	85.68	63.36	74.79
NER	49.61	39.69	44.10	90.21	57.19	70.00
NLI	53.93	59.08	56.38	86.37	71.66	78.33
SBD	60.96	66.77	63.73	88.08	73.08	79.88
GPT4	60.39	66.15	63.14	89.57	74.32	81.24

Table 4: Performance of different methods on the internal benchmark and public SummAC. **SBD** methods is very competitive and worth further exploring.

with latency of GPT4 on the internal benchmark dataset, our finetuned NLI and SBD models enjoy significant latency advantages.

Method	Latency (s/request)
NLI model	1.2s (on V100 GPU)
SBD model	1.3s (on V100 GPU)
GPT4 w/ CoT	7.9s (per API call)

Table 5: Latency of finetuned models and the GPT4-based method.

5.2 Mitigation Results

Here is the evaluation setting:

Benchmark dataset. This is an internal testset of 200 samples, derived from an application or feature team. This dataset includes 100 documents with two summaries from genearated GPT4 per document: one in paragraph format and one in bullet-point format.

Metrics. We evaluated performance using two key metrics: *mitigation rate* measures the percentage of hallucination being successfully corrected fixed, as determined by GPT4-detection; and *GPT-4 output tokens length* is serving for as proxy metrics for evaluating the latency and cost.

As shown in Table 6, Rewriting Prompt v2 is great token efficiency and achieves good a balanced

trade-offce among rewriting quality, latency, and cost.

Rewriting Prompt	Mitigation Rate	GPT-4 Output Tokens (avg)
No rewrite	0.0	244 (original output)
Prompt v1	66.0%	587
Prompt v2	44.7%	130

Table 6: Rewriting to balance the quality and latency.

5.3 Performance in Production

We have developed two pipelines for production usage: 1) the detection-only pipeline and 2) the detection with mitigation pipeline. They have been integrated into LLM-based products to mitigate the customer's complaints about hallucination as in Figure 4. In practice, we start with the detection-only pipeline and block the hallucinated context from affecting the customers. Gradually, we move on to the full pipeline of detection and mitigation.
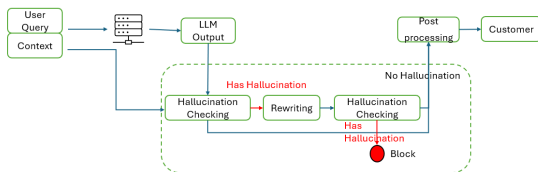

Figure 4: User experience of Hallucination Detection & Mitigation Pipeline in Production

Accurately measuring the effectiveness of hallucination detection and mitigation in real-world applications is a complex task. To address this, we adopt two approaches: 1) offline human evaluation using a production-related benchmark dataset and 2) GPT4-based online monitoring based on the actual production traffic.

For the offline human evaluation, we applied our detection and rewriting pipeline to a production-related testset consisting of bullet-point style summaries comprising of 630 individual keypoints. We have 25 (i.e. 4.0% of 630) in total detected as containing hallucinations. An independent human data labeler to verified that 15 of the 25 dectections were accurate (i.e. check precision is **60.0%**). Additionally, the human labeler to check if rewriting fixes the hallucinations, with 10 out 15 are accurately fixed (i.e. rewriting effective rate is **66.7%**)

In the online monitoring approach, we sampled a portion of production traffic - comprising pairs of <LLM input, LLM ouput > - for evaluation. GPT4

checked outputs for hallucinations and assessed if the rewriting process was necessary. Using GPT4's judgement as a benchmark, we can observed that the precision of ensemble detection is **above 80%** (i.e. 80% of the time is consistent with GPT4's judgement) when the detection rate is about 3% and rewriting success rate is **above 50%**. In Table 7, we use statistics based on one-month production traffic to show both the detection-only pipeline and full pipeline of detection and mitigation can effectively reduce hallucinations, albeit an acceptable increase in latency.

Pipeline	Rewriting Rate (% traffic entering GPT4-rewriter)	Block rate (% of output are hallucinated)	Latency (s)		
			P50	P95	P98
Detection-only	0.0%	3.2%	1.2	2.0	2.4
Detection & Mitigation	3.2%	0.58%	1+0	2.0+0	2.4+**4.7**

Table 7: Online monitoring and comparisons of the pipelines based on production traffic.

6 Challenges and Future Work

6.1 Measurement of Effectiveness in Production

Accurately measuring the effectiveness of hallucination detection and mitigation, as well as the value they bring to customers in a production environment, is very challenging. We have designed a system that applies mirrored traffic to various pipelines and uses GPT4 to assess hallucination rates and the overall quality of rewritten content. However, the GPT4-based measurement has limitations and ensuring the reliability of these measurements and their alignment with human judgment remains an ongoing challenge, necessitating continuous refinement and validation.

6.2 Handling Multilingual and Long Source Documents

We have incorporated major non-English training datasets into our NLI and SBD models to support multilingual use cases and are utilizing a segmentation-based approach to manage long source documents. However, handling inputs and outputs in different languages and their extensive combinations remains challenging. Additionally, developing effective models for processing long source documents continues to be an open research problem, requiring further exploration and innovation.

6.3 Deep Customization Needs for Hallucination Handling

Different user circumstances call for tailored adjustments to hallucination handling. For example, to meet the production needs, we've calibrated our ensemble-based detection for greater precision with a reduced block rate (or trigger rate) to avoid the availability issue, while also adjusting our rewriting for decreased latency at the cost of some mitigation power. However, there might be another setting, where we need a different balance of quality, latency and cost. Also, adapting to special domain or task or handling a specific types of hallucinations can also be great directions to explore.

7 Conclusions

In this paper, we introduce a novel framework that can detect and mitigate intrinsic hallucinations, characterized by outputs not supported by grounding documents in LLMs. Our detection approach leverages the combined strengths of NER, NLI, and novel sequence labelling (SBD), and Decision Tree to detect as much as hallucination as possible. We further developed an effective LLM-based mitigation solution that blance the quality and latency.

We detail the core elements of our framework and underscore the paramount challenges tied to response time, availability, and performance metrics, which are crucial for real-world deployment of these technologies. Our extensive evaluation, utilizing offline data and live production traffic, confirms the efficacy of our proposed framework and service.

8 Ethical Considerations

Ethical considerations are paramount in the development and deployment of hallucination mitigation systems. Ensuring transparency in detection and mitigation processes, providing clear explanations for decisions, and safeguarding user data are essential components of our ethical framework. Balancing these ethical imperatives with technical and operational demands is a complex but necessary challenge.

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages

632–642, Lisbon, Portugal. Association for Computational Linguistics.

Meng Cao, Yue Dong, and Jackie Chi Kit Cheung. 2021. Hallucinated but factual! inspecting the factuality of hallucinations in abstractive summarization. *arXiv preprint arXiv:2109.09784*.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2023. Chain-of-verification reduces hallucination in large language models. *Preprint*, arXiv:2309.11495.

Nouha Dziri, Andrea Madotto, Osmar Zaïane, and Avishek Joey Bose. 2021. Neural path hunter: Reducing hallucination in dialogue systems via path grounding. *arXiv preprint arXiv:2104.08455*.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. *Preprint*, arXiv:2006.03654.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *Preprint*, arXiv:2311.05232.

Yichong Huang, Xiachong Feng, Xiaocheng Feng, and Bing Qin. 2021. The factual inconsistency problem in abstractive text summarization: A survey. *arXiv preprint arXiv:2104.14839*.

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.

Ryo Kamoi, Tanya Goyal, Juan Diego Rodriguez, and Greg Durrett. 2023. Wice: Real-world entailment for claims in wikipedia. *Preprint*, arXiv:2303.01432.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022. SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.

Deren Lei, Yaxi Li, Mengya Hu, Mingyu Wang, Vincent Yun, Emily Ching, and Eslam Kamal. 2023. Chain of natural language inference for reducing large language model ungrounded hallucinations. *Preprint*, arXiv:2310.03951.

Tianyu Liu, Yizhe Zhang, Chris Brockett, Yi Mao, Zhifang Sui, Weizhu Chen, and Bill Dolan. 2021. A token-level reference-free hallucination detection

benchmark for free-form text generation. *arXiv preprint arXiv:2104.08704*.

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *Preprint*, arXiv:2303.08896.

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. *Preprint*, arXiv:2401.06855.

Niels Mündler, Jingxuan He, Slobodan Jenko, and Martin Vechev. 2024. Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation. *Preprint*, arXiv:2305.15852.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial nli: A new benchmark for natural language understanding. *Preprint*, arXiv:1910.14599.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4812–4829, Online. Association for Computational Linguistics.

Ernesto Quevedo, Jorge Yero, Rachel Koerner, Pablo Rivas, and Tomas Cerny. 2024. Detecting hallucinations in large language model generation: A token probability approach. *arXiv preprint arXiv:2405.19648*.

Tal Schuster, Adam Fisch, and Regina Barzilay. 2021. Get your vitamin C! robust fact verification with contrastive evidence. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 624–643, Online. Association for Computational Linguistics.

James Thorne and Andreas Vlachos. 2021. Evidence-based factual error correction. *Preprint*, arXiv:2012.15788.

S. M Towhidul Islam Tonmoy, S M Mehedi Zaman, Vinija Jain, Anku Rani, Vipula Rawte, Aman Chadha, and Amitava Das. 2024. A comprehensive survey of hallucination mitigation techniques in large language models. *ArXiv*, abs/2401.01313.

Jerry Wei, Chengrun Yang, Xinying Song, Yifeng Lu, Nathan Hu, Jie Huang, Dustin Tran, Daiyi Peng, Ruibo Liu, Da Huang, Cosmo Du, and Quoc V. Le. 2024. Long-form factuality in large language models. *Preprint*, arXiv:2403.18802.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. Alignscore: Evaluating factual consistency with a unified alignment function. *arXiv preprint arXiv:2305.16739*.