On A Scale From 1 to 5: Quantifying Hallucination in Faithfulness Evaluation

Anonymous ACL submission

Abstract

Hallucination has been a popular topic in natural language generation (NLG). In real-world applications, unfaithful content can result in bad data quality or loss of trust from end users. Thus, it is crucial to fact-check before adopting NLG for production usage, which can be expensive if done manually. In this paper, we investigate automated faithfulness evaluation in guided NLG. We developed a rubrics template and use large language models (LLMs) to score the generation into quantifiable scales. We compared popular LLMs as well as the widely adopted natural language inference (NLI) models in scoring quality and sensitivity. In addition, we developed methods to generation synthetic unfaithful data, as well as a heuristics to quantify the percentage of hallucination. Our results on 4 travel-domain industry dataset show that GPT-4 (Achiam et al., 2023) can provide accurate judgement and explanation on whether a source and a generation are factually consistent. Furthermore, we found that tuning NLI models on synthetic data can improve performance. Lastly, we present insights on latency and cost for deploying such system.

1 Introduction

The recent advancement of large language models (LLM) has made it easier than ever to build NLP applications. With LLM's powerful natural language generation (NLG) ability, one can easily leverage prompt engineering to generate texts. However, pretrained transformer decoders are prone to hallucinations. Previously it was shown that the performance of LLM's decoding strategies (e.g. beam search) have a dependency on the type of the generation task (Wiher et al., 2022). As a result, LLMs can sometimes generate completed "made-up" sequences, in open-ended generation like story telling and close-ended generation like summarization (Maynez et al., 2020). Many recent methods have been proposed to mitigate hallucination, including retrieval-augmented generation (Lewis et al., 2020), chain-of-thoughts (Wei et al., 2022), tree-of-thoughts (Yao et al., 2024), self-consistency (Wang et al., 2023b), self-reflection (Ji et al., 2023b), controlled decoding (Mudgal et al., 2023), instruction tuning (Liu et al., 2023a). Nevertheless, the current state-of-the-art does not guarantee that hallucination can be 100% prevented. To apply NLG in industry applications, oftentimes having a "faithful enough" model is not sufficient to productionize an LLM-driven experience. With the aim of adding another layer of security to assure the accuracy of LLM generated content, we investigate an automated approach to effectively score degrees of faithfulness in guided text generation.

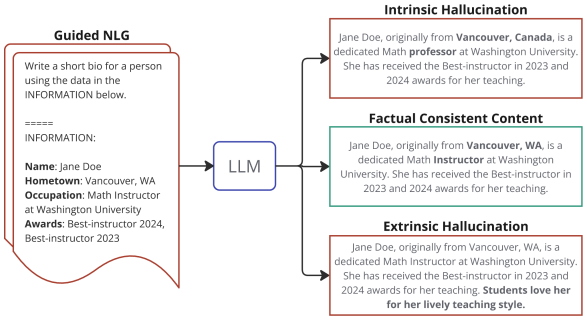


Figure 1: Guided NLG showcasing a text generation example. Given a prompting template and grounding data, hallucination can happen in intrinsic or extrinsic ways. Both can be harmful to a product because the generated contents can either be wrong or unverified.

Figure 1 illustrates a simple example of guided generation scenarios. Given guidelines (the prompt template) and grounding data (the source facts), an LLM model can hallucinate and generate either intrinsic hallucinating content in which the facts contradict that of the source or extrinsic hallucinating content in which new facts are added that cannot be verified from the source (Ji et al., 2023a). It should be noted that extrinsic hallucination does not necessarily contain incorrect information with

respect to world knowledge. It is possible that the LLM interpreted the additional facts based on its internal knowledge. On the other hand, faithfulness is considered an antonym to "hallucination", which is defined as staying factually consistent with the provided source (Ji et al., 2023a). A higher faithfulness score thus indicates lower hallucinations. We define faithfulness of a generated text with respect to its source as the following:

- ***reference***: a text containing the source or ground-truth information, often longer
- ***hypothesis***: a text to check against the reference, which is derived by a model

The hypothesis is factually consistent with the reference if all factual elements in the hypothesis can be traced back to the reference. A faithfulness score measures: 1) to what extent the hypothesis can be verified by the reference; and 2) to what extent the LLM hallucinates when generating the hypothesis from the reference.

We investigated two approaches, namely using LLMs with a grading rubrics and natural language inference (NLI) models, to derive a faithfulness score for a given reference and hypothesis pair. We experimented with 5 LLMs and 3 variants of NLI models on 4 travel-domain dataset. We studied the score variation by comparing various types of hypothesis: gold hypothesis, sentence-level gold hypothesis, intrinsic hypothesis, and extrinsic hypothesis. In addition, we explored the model sensitivity in picking up different levels of hallucination. Our main contributions are: First, we showed that LLMs can accurately distinguish faithfulness apart from hallucination, and that reasoning during grading help LLMs gain sensitivity towards hallucination. Second, we presented methods to generate synthetic hallucination by type. Third, we demonstrated tuning with synthetic data can improve NLI model's performance. Lastly, we illustrated the progression of scores based on the percentage of hallucinating content.

2 Related Works

NLI for NLG evaluation NLI based models has been used widely in factual consistency evaluation. Derived originally from the text entailment (Dagan et al., 2005), given a premise, a model should classify the premise to be an "entailment", "neutral", or "contradiction" to a hypothesis (Bowman et al., 2015). Laban et al. (2022) proposed SummaC (Laban et al., 2022) which demonstrated that a fact check with NLI-based models on summarization performed better when using sentence-level granularity in both reference and hypothesis. Honovich et al. (2022) introduced TRUE benchmark for factual consistency evaluation, which showed that the NLI-based binary classifier could retain state-of-the-art performance over summarization, dialogue, paraphrasing, and fact checking tasks. Falsesum (Utama et al., 2022) is a data generation pipeline that utilized OpenIE and a fine-tuned T5-base (Raffel et al., 2020) model to create synthetic factually inconsistent data for improving NLI-based classifier. The "neutral" and "contradiction" labels were together treated as "non-entailment".

LLM for NLG evaluation. Recently, many studies have also proven that an LLM could be used as a judge for evaluating NLP tasks (Zheng et al., 2023). Wang et al. (2023a) showed that ChatGPT can perform as well as human judges in NLG evaluation, in which a broader generation quality was graded between the reference and the hypothesis. Liu et al. (2023b) proposed G-Eval which used a chain-of-thought framework to evaluate NLG by granular categories. The faithfulness component was drafted as a binary classification for factual inconsistency identification. Chiang and Lee (2023) expanded on G-Eval and demonstrated that grading with explanation can improve the LLMs evaluation accuracy in summarization. Although the factual consistency was scored on a 1-5 scale, LLM's internal knowledge was used as criteria. Chang et al. (2024) showed that LLM can detect hallucinations as a binary classification problem in RAG systems in a zero-shot setting, and that training on synthetic data can further improve the detection accuracy. Gekhman et al. (2023) presented TrueTeacher which improved both LLM- and NLI-based factual consistency model accuracy by training on LLM-generated synthetic datasets.

Previous works on NLI models focus on classification and lack of explainability, while LLM-based evaluation is still emerging which has not arrived formalization. As far as we know, few studies have tried to quantify the degree of faithfulness. Furthermore, little documentation about how NLI models compare to LLM evaluation is available for domain specific real-world use cases. Finally, comparison in terms of what percentage of hallucinations affect the final scores has not been explored extensively.

Dataset Name	Number of Data Pairs	Reference #Avg Tokens	Reference # Avg Sents	Hypothesis # Avg Tokens	Hypothesis # Avg Sents
ConvoAS	97	500.77	10.67	142.86	5.52
ConvoTS	135	745.91	15.24	50.07	2.33
ReviewTS	128	120.82	6.57	15.54	1.16
JsonTG	108	159.52	4.54	108.64	4.16

Table 1: Number of gold reference and hypothesis pairs, average tokens, and average number of sentences by data type. Number of tokens are estimated based on a $cl100k_base$ tokenizer. It should be noted that because the references in JsonTG are structured JSON data, the number of sentences is not applicable. We instead included the length of the decomposition for reference. There are a total of 4,503 pairs including synthetic data.

3 System

We evaluated faithfulness scoring models on 4 travel-domain industry datasets. It should be noted that our dataset are taken from production and only contains gold data. We discuss means to create synthetic hallucination dataset in Section 3.1. Statistics of the dataset can be found in Table 1.

3.1 Dataset

ConvoAS is an abstractive summarization dataset. Our source data consists of customer support transcripts between 2 participants, a traveler and an agent. GPT-4o (OpenAI, 2024b) was used to generate a summary for the entire transcript. Information in the summary such as date, price, and location needs to be traceable from the source.

ConvoTS is a topic-specific summarization dataset. Similar to ConvoAS, this dataset also contains customer support transcripts, except 3 topic-specific summaries were generated for each transcript. It should be noted that many redundant turns exist because the chat happened between a traveler and a virtual agent through a guided service. GPT-4o (OpenAI, 2024b) was used for this use case.

ReviewTS is a topic-specific summarization dataset from reviews. Each reference consists of multiple review snippets related to a topic of interest (e.g. pool). A fine-tuned Mistral 7B (Jiang et al., 2023) model was used to extract and summarize the snippets. The summary is restricted to less than 100 characters. Because of the concise nature of this data, full-length and sentence-level hypothesis variants contain large overlaps. However, we kept the same setup for consistency.

JsonTG contains key-valued pairs represented in JSON format for stylized text generation. This task aims at generating property headlines and descriptions given the grounding information such as amenities. The LLM, Claude3 Haiku (Anthropic, 2024a), was allowed to come up with expressive verbs and adjectives, but the grounding information must be traceable from the source JSON data.

Synthetic Hallucination Data. Due to insufficient hallucination examples, we drafted guidelines for GPT-4o to generate a synthetic counterpart from the gold dataset to study model performance. Since the hallucination could take the form of either intrinsic, for contradictory facts, or extrinsic for extra knowledge, we mimicked both scenarios. For the former, we prompted GPT-4o to modify a piece of fact in the original sentence; and for the latter, we instructed the model to incorporate a piece of new world knowledge while the original sentence remains unchanged. Each dataset thus resulted in 4 varying hypothesis versions, namely gold full-length, gold sentence-level, intrinsic sentence-level, and extrinsic sentence-level. Furthermore, we simulated a numeric percentage at the sentence level to measure how much effect the portion of unfaithful content has on the evaluation scores. We took a ConvoAS subset with a 5-sentence hypothesis (51 samples). For every gold hypothesis, we incrementally swapped out 1 to 5 random sentences by its counterpart. The percentage of sentences that contain unfaithful information can thus be treated to mimic the "hallucination percentage" with respect to the overall hypothesis. It should be noted that it is inherently difficult to justify "the number of facts" contained in downstream tasks like abstractive summarization because the granularity of the content is subjective to the use case and the boundary between different "pieces of facts" can be fuzzy. While the proposed heuristic approach is far from perfection, it is sufficient to serve our goal of learning the scoring trend and model sensitivity.

Table 2 illustrates synthetic hallucination examples created from the gold sentences.

Dataset	Type	Sentence
ConvoAS	Gold	The total charge for the booking is 1,078.84 CAD.
	Intrinsic	The total charge for the booking is **899.50** CAD.
	Extrinsic	The total charge for the booking is 1,078.84 CAD, **which is roughly equivalent to the cost of a new iPhone 14.**
ReviewTS	Gold	Walking distance to the beach.
	Intrinsic	**A short drive** to the beach.
	Extrinsic	Walking distance to the beach, **where the local farmers' market is held every Saturday morning.**
JsonTG	Gold	Cozy 1-bedroom vacation home steps to One World Trade Center
	Intrinsic	Spacious **3-bedroom apartment** miles away from One World Trade Center
	Extrinsic	Cozy 1-bedroom vacation home steps to One World Trade Center, **and just a short walk from the historic Trinity Church**

Table 2: Examples of hallucinating sentences generated by prompting gpt-4o to 1) replace a piece of fact in gold hypothesis (intrinsic); and 2) expand on the gold hypothesis to add a piece of new world knowledge (extrinsic).

3.2 Entailment-based Evaluation

We adopted Vectara's open-source HHEMv1.0[1] model (Bao et al., 2023). HHEM is a 184.4 million parameter lightweight Deberta model (He et al., 2021) fine-tuned on various NLI dataset, including TRUE (Honovich et al., 2022) and SummaC (Laban et al., 2022). Given a pair of text *(reference, hypothesis)*, HHEM scores the level of entailment between the pair to a value in the range [0, 1], with 1 being entailment and 0 being contradictory.

One limitation is that HHEM model has a max input size of 512 tokens, which is prone to long text pairs. By default, truncation is applied to overflowing tokens starting at the end of the texts. In this case, the hypothesis will be truncated first, which will lead to information loss and worse case, totally wrong evaluation. To minimize information loss for maximized accuracy, we implemented a sliding window segmentation to the reference. Overlapping tokens were taken into consideration to include the previous context. The size of the current segment is computed dynamically based on tokens in the hypothesis as shown below:

[CLS] overlap + current segment [SEP] hypothesis [SEP] [PAD]

The hypothesis is then evaluated against every reference segment and a final score for each pair is max aggregated across all segments. The reason behind segmenting the reference rather than the hypothesis is that the reference is often longer and contains redundancy. It is less sensible to segment the hypothesis, in which the information is typically already compressed, as it will result in more data loss. For the size of the overlapping tokens, we intuitively set it to 32 tokens which corresponds to around 25 words. This length is slightly longer than the length of an average sentence, which is said to be around 15-20 words.

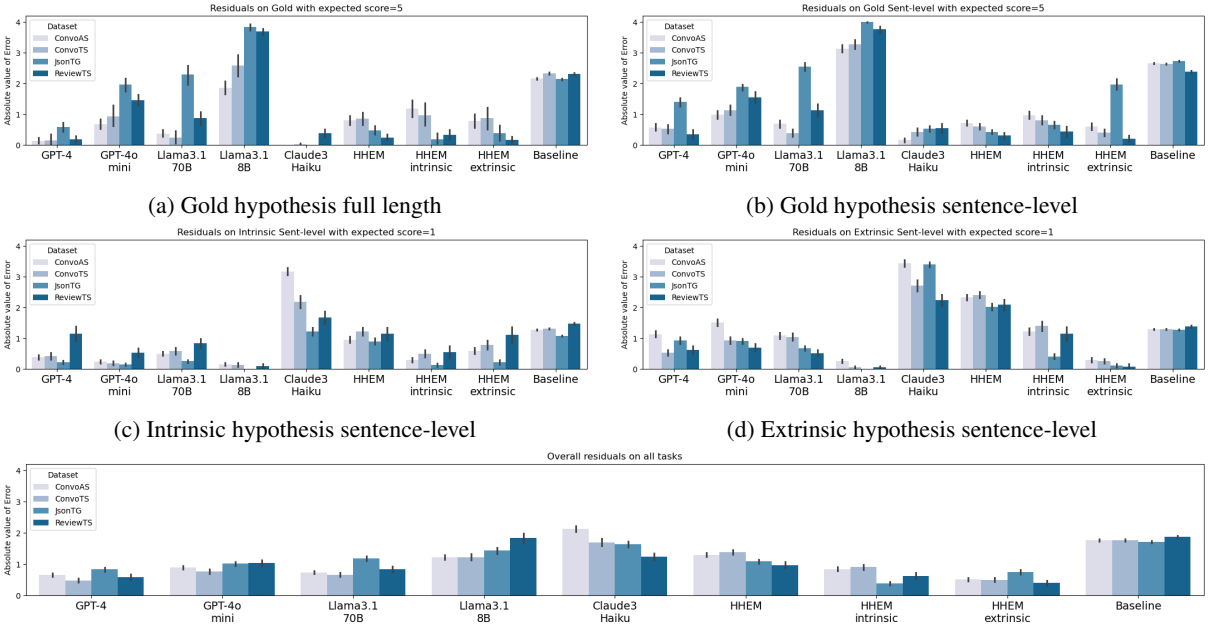
In addition to testing the capability of HHEM out-of-the-box, we also fine-tuned 2 versions of the HHEM model with an extension of JsonTG. The preliminary accuracy on JsonTG was only promising when the JSON was decomposed into plain texts. While it is expected that encoder models perform poorly on unseen data structures; and that pursuing a JSON decomposition heuristic is feasible on a case-by-case basis, we were curious whether fine-tuning on stringified JSON can generalize the model to this data structure while preserving the same level of performance on other data.

3.3 Rubrics-based LLM Evaluation

Following the accuracy metric defined in HELM (Liang et al., 2022), we crafted a rubrics-based approach to use LLM as a judge to score the faithfulness of the generated contents.

Given a *(reference, hypothesis)* pair, the model was instructed with a system prompt to evaluate four aspects: 1) factual consistency which checks if all facts, e.g. numeric values and proper nouns, in the hypothesis can be traced back to the reference; 2) adjective regularity which examines if the adjectives used in the hypothesis are synonymous with their corresponding counterparts in the reference; 3) knowledge congruence which evaluates whether extrinsic information is injected to the hypothesis;

[1] https://github.com/vectara/hallucination-leaderboard/tree/hhem-1.0-final



(a) Gold hypothesis full length

(b) Gold hypothesis sentence-level

(c) Intrinsic hypothesis sentence-level

(d) Extrinsic hypothesis sentence-level

(e) All Tasks

Figure 2: Residuals computed as the absolute error between expected and predicted scores. The scores range from "1-highly unfaithful" to "5-highly faithful". The y-axis denotes the absolute value of the errors with a shorter bar indicating better model performance at a corresponding task. Tasks are arranged in the order of (a) Gold full length; (b) Gold sentence-level; (c) Intrinsic sentence-level; (d) Extrinsic sentence-level; and (e) All tasks. The expected score for gold data (a) and (b) should be 5 whereas for hallucination data (c) and (d) should be 1. Overall, GPT-4 is the most capable of scoring both gold and hallucination data. LLMs with less than 10B parameters seem to struggle with different aspects of the scoring. Tuning on HHEM with additional synthetic examples also improved overall performance.

and 4) style alignment which scans whether the language and tone in the hypothesis match with those from the reference.

We defined the grading rubric to follow a discrete range from 1 to 5, with 5 being "highly faithful" that all information in the hypothesis can be verified in the reference, and 1 being "highly unfaithful" which most or all of the content in the hypothesis cannot be validated from the reference. Additionally, as prior research has shown that self-reflection by including an explanation (Renze and Guven, 2024) can effectively improve LLM performance, we also instructed the model to provide detailed reasoning along with the numeric score to further improve the grading precision. More details of the prompt can be found in Appendix A.1

3.4 Baseline

We integrated a composite score from ROUGE-L (Lin, 2004), BLEU (Papineni et al., 2002), and BERTScore (Zhang et al., 2019) as our baseline.

4 Result & Discussion

In this section, we discuss faithfulness scoring on the reference and hypothesis pairs for two sets of experiments. The score ranges from "1-very unfaithful" to "5-very faithful", with discrete intervals "2-very faithful", "3-slightly unfaithful", and "4-very unfaithful". It should be noted that HHEM models return a continuous probabilistic value in the range [0, 1]. It was later scaled to [1, 5] for ease of interpretation.

Our first experiment aims to evaluate the models' scoring quality on gold and hallucinating contents. The experiments (Figure 2) outline model comparison across all datasets over the gold hypothesis, gold hypothesis on a sentence level, intrinsic hypothesis on a sentence level, and extrinsic hypothesis on a sentence level. On the other hand, the second set of experiments aims to study the score progression when the percentage of hallucinating content varies in the hypothesis. The comparison (Figure 3) involves scoring on both intrinsic and extrinsic hypotheses with 0% to 100% hallucinating contents in 20% steps. Along with the score progression, we also tested the significance of "reasoning" to LLM's scoring capability. Models tested are as follows: for rubrics-based scoring with LLMs: GPT-4 (Achiam et al., 2023), GPT-4o-mini (OpenAI, 2024a), Llama3.1-8B (Meta, 2024),

	Model	Task	Reasoning	Scored
1	GPT-4	gold sent	The summary, however, simply states 'No further assistance was requested, which **does not capture the essence** of the conversation...	1-highly unfaithful
2	Llama3.1 8B	gold full	... as it stated **'all-inclusive atmosphere'** instead of **'all inclusive'**, and **'plenty of onsite activities'** instead of '**plenty to do right on site'** ...	1-highly unfaithful
3	Llama3.1 8B	gold full	... it **lacks specific details about the kitchen**, such as the stove, ..., which are mentioned in the source ...	1-highly unfaithful
4	Claude3 Haiku	gold sent	... from the source document **that no changes were made to the reservation**. This is directly stated in the source ...	5-highly faithful
		intrinsic	.. accurately states **that several changes were made to the reservation**, which is consistent with the source document	5-highly faithful

Table 3: Example reasoning provided by models on failed cases.

Llama3.1-70B (Meta, 2024), and Claude3 Haiku (Anthropic, 2024a); for entailment-based approach HHEM (Bao et al., 2023), HHEM-intrinsic, and HHEM-extrinsic.

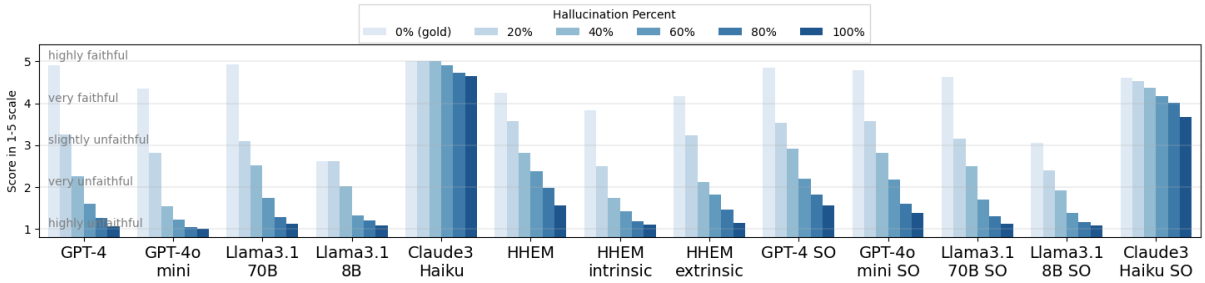
4.1 Scoring Quality

Figures 2a - 2e elaborates on the residuals of the scores. Overall, GPT-4 with zero-shot rubrics performed the best among all models, and Llama3.1-70B followed closely after. The tuned HHEM-extrinsic model also shows high accuracy with a slight flaw in scoring sentence-level JsonTG dataset. A slight performance drop can be observed transitioning from the gold hypothesis (2a) to its sentence level (2b). GPT-4 scored 6.8% out of 1393 gold sentences to "1-highly unfaithful", which was mainly due to incomplete information in individual sentences. As a result, several unfaithful occurrences were assigned reason 1 in Table 3. Furthermore, JsonTG reference seems to be consistently challenging to all models across all tasks. After inspecting the LLM provided reasoning, we found that this is because our adjective verification rubrics to capture "adjectives without any basis" contradicts the JsonTG generation task which allows LLMs to come up with expressive adjectives. For posterity, it is best to adapt the rubrics to a per-use case basis for better accuracy.

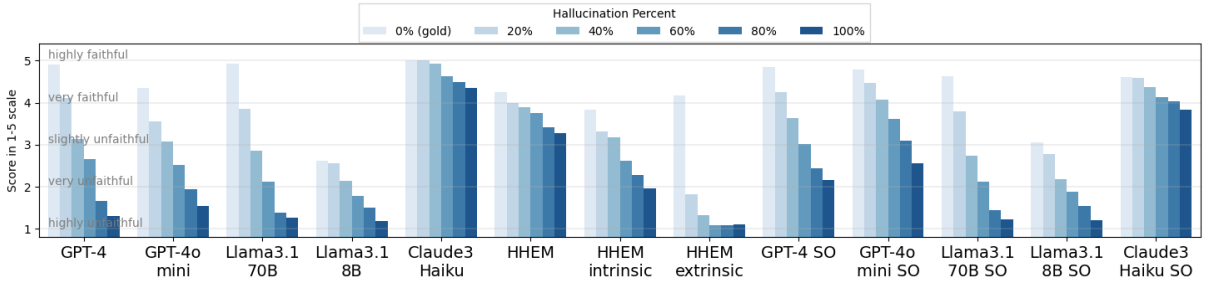
LLMs with fewer parameters such as Llama3.1-8B and Claude3-Haiku seem to struggle to understand the rubrics. Llama3.1-8B tends to be overly strict in that it scored all tasks to "3-slightly unfaithful" and below, indicating the hypothesis is unfaithful. This seems to be a result of Llama3.1-8B's following the instructions too literally. In addition, Llama3.1-8B seems to have confused "factual consistency" with "granularity". The example LLM reasoning can be found on row 2 and 3 in Table 3. In contrast, Haiku tends to be extremely lenient. It failed consistently on both hallucination tasks while scoring almost perfectly on the gold hypothesis. It appears that Haiku was able to capture the key information in the hypothesis, but could not make the association to the contradictory part in the reference. Row 4 in Table 3 illustrated an example where Haiku incorrectly scored an intrinsic hypothesis to "5-highly faithful".

It is interesting that the above observation is not reflected in the overall residual plot in Figure 2e as the scores got smoothed out by half of the tasks with good performance. However, it should be noted that although the models' strictness and leniency are consistent across different tasks, in reality, it would be difficult to integrate them into the production pipeline due to the unpredictability brought by their scoring bias.

Unlike LLMs, HHEM models do not experience significant performance degradation from full length to sentence-level hypothesis. This is likely because the base Deberta model was originally trained for NLI, which uses a slightly fuzzy definition for "textual entailment" that a premise entails a hypothesis given a very probable inference can be made from the premise (Dagan et al., 2005). Thus, a piece of a sentence from the hypothesis would receive a higher score during the prediction as long as the encoded information can be inferred from the source. It should be noted that HHEM models are naturally prone to input token size, evaluating on sentence level could improve efficiency



(a) Intrinsic Hallucination



(b) Extrinsic Hallucination

Figure 3: Scores on a hypothesis with different hallucination percentage. (top) Intrinsic hallucination. (bottom) Extrinsic hallucination. The scores range from "1-highly unfaithful" to "5-highly faithful". The percentage is simulated by gradually increasing the amount of hallucinating sentences in the 5-sentence gold hypothesis (51 samples). Models with an "SO" suffix indicate "score only", in other words, the LLMs were not asked to provide reasoning. Overall, models perform better towards intrinsic hallucination. Including reasoning in response can increase the sensitivity of the model in picking up hallucinations.

by inducing fewer segments on the reference.

HHEM models are also less sensitive toward extrinsic hallucinations as compared to intrinsic hallucinations. Fine-tuning on extrinsic synthetic JSON data alone helped HHEM-extrinsic model gain better performance on almost all tasks. However, during fine-tuning, we noticed that it was rather easy to over-tune the HHEM model and made it less tolerate to any type of creative contents. We first tried tuning with 400 each gold and hallucination examples for 5 epochs, which resulted in overly strict scores. By reducing the number of epochs to 3, we were able to achieve the performance as shown in Figure 2. We believe if the model is trained on more diverse synthetic examples with a carefully crafted dataset, the results would be more stable than what we have shown here.

4.2 Scoring Progression

Figures 3 illustrates the score progression trends with respect to the percentage of the hallucinating information. To recap, the percentage-based hallucination dataset was created by randomly replacing the N-numbers of gold sentences with hallucinating ones. Each hallucinating sentence contains a subtle change as demonstrated in Table 2. Consequently, the hallucination percentage is in fact the percentage of sentences containing unfaithful information. In this iteration, we also tested LLM's scoring capability when the "reasoning" component was removed from its response. The score-only model's performance is suffixed with "SO".

As shown in Figures 3a - 3b for both intrinsic and extrinsic hallucination, the faithfulness score decreases accordingly as the percentage of hallucination increases from 0% to 100%. This trend is consistent even for lower-performing LLMs such as Llama3-8B and Claude3-Haiku. As mentioned earlier, all models tend to perform better on intrinsic hallucination over extrinsic ones. We can potentially explain this observation further by comparing the changes in score going from lower to higher hallucination percentage. Taking GPT-4 as an example, in Figure 3a, when the intrinsic hallucination percentage increases from 0% -> 20% -> 40%, the score decreases from 4.90 -> 3.23 -> 2.26; whereas for extrinsic hallucination, the score decreases in a more gradual manner, namely 4.90 -> 4.13 -> 3.15, with every step taken. GPT-4's sensitivity to identify the initial 20% of intrinsic hallucination is roughly mapped to detecting 40% of extrinsic hallucination. Tracing back to our rubrics, although "avoiding speculative or creative content generation" was included in the guidelines,

	GPT-4	GPT-4o mini	Llama3.1 70B	Llama3.1 8B	Claude3 Haiku	HHEM models
reasoning	6.82s	1.08s	1.85s	2.02s	1.94s	N/A
score only	1.54s	0.41s	1.28s	1.99s	2.15s	0.007s
out tokens	117.08	109.18	24.69	121.67	95.90	N/A
price($)	0.0452	0.0003	0.0013	0.0003	0.0004	N/A

Table 4: Utilization per reference & hypothesis pair in terms of latency in seconds and price. The number of output tokens containing both reasoning and score was estimated by a *cl_100k* tokenizer for consistency. Score-only evaluations output 6 tokens consistently. The latency of HHEM models was estimated with batch size 32 on an AWS G5 instance. The LLMs are hosted through the same proxy. It should be noted that LLM's latency can be affected by not only the number of generation tokens but also the overall traffic load.

the score assignment was lenient towards extrinsic hallucination. We designed the "3-slightly unfaithful" to allow unverifiable information to be present without negatively impacting the reader's experience. Tightening the rubrics could help increase the sensitivity in detecting extrinsic hallucinations. However, one should proceed with caution and support the rubric's development by using similar experiments to ensure that the LLMs maintain a good tolerance between unfaithful and creative content.

We also quantified the scores without reasoning by simply removing the instruction to justify the reason for grading. Overall, in Figures 3a - 3b, all LLMs demonstrated various levels of performance degradation. For both intrinsic and extrinsic hallucinations, scoring without reasoning made the LLMs more lenient towards the unfaithful content. As the percentage of hallucination increases, the decreases in score converge much slower. As a result, none of the LLMs scored a 100% hallucinating content to "1-highly unfaithful". Another interesting observation is that the strictness of Llama3.1-8B and the leniency of Claude3-Haiku have both relaxed slightly without reasoning, which caused a performance "improvement". We argue that this is because the reasoning is the key to keeping the model "sharp", the "improvement" is really just a form of unresponsiveness when the "thinking" step is removed from the process.

4.3 Latency & Usage

Table 4 illustrates the latency and price for each model. All LLMs were tested through HTTP endpoints hosted through the same proxy service. The average time in seconds, number of output tokens, and prices were calculated in a sequential API call setting. By nature, smaller encoder models are inherently faster than decoder-based LLMs, thus we only computed a batch inference latency for HHEM models ran on an AWS G5 instance. The pricing referred to OpenAI [2] and AWS Bedrock [3] standards. While GPT-4 yields the best accuracy, it is also the most costly model on both latency and price. On the other hand, although Llama3.1-70B seems to have balanced out the accuracy vs. latency, Llama3.1-70B provided reasoning can be barely useful to a human auditor due to little granularity. It should be noted that one could customize the prompt instruction to adjust the level of granularity to achieve the desired balance. For real-world usage, we recommend combining a fast model with acceptable performance as a first pass evaluation; then apply more powerful LLMs with detailed explanation to assist human audits.

5 Conclusion

We proposed a rubrics-based template to use LLMs to score faithfulness in NLG. We evaluated 5 popular LLMs on 4 travel-domain indutry dataset. The results indicate that in a zero-shot setting LLMs, e.g. GPT-4, outperforms NLI models, e.g. HHEM, on capturing both intrinsic and extrinsic hallucinations. HHEM model with fine-tuning on synthetic data can outperform LLMs in domain specific evaluation given carefully crafted training data. Lastly, we introduced a heuristic to quantify faithfulness score by hallucination percentage, which helps visualize the sensitivity when developing faithfulness evaluation models. Our future work include exploring the score association between different LLMs from the same family and tuning the rubrics to adjust the scoring sensitivity.

[2] https://openai.com/api/pricing/

[3] https://aws.amazon.com/bedrock/pricing/

6 Limitations

1. Due to resource limitations, we focused on testing only travel-domain dataset, but did not include popular open source faithfulness dataset. We justify that real-world dataset are often more challenging than research domain dataset, as there exist more noise in real-world data. We believe our approach can be generalizable to other domains.

2. Due to legal and privacy constraints, we do not plan to release our dataset as some of it contains sensitive customer information.

3. Due to security concerns, the proxy service used for LLM testing contains enterprise guardrails which moderate incoming and outgoing data. Thus, the latency reported in Table 4 would be higher than directly calling OpenAI and Bedrock APIs.

4. Due to resource limitations, we were not able to evaluate a larger model from the Anthropics family, such as the Claude3.5-Sonnet (Anthropic, 2024b). It was not our intention to make an unfair comparison using only a smaller Claude model.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Anthropic. 2024a. Claude 3 haiku: our fastest model yet.

Anthropic. 2024b. Claude 3.5 sonnet.

Forrest Bao, Miaoran Li, Rogger Luo, and Ofer Mendelevitch. 2023. HHEM-1.0-Open.

Samuel 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.

Tyler A Chang, Katrin Tomanek, Jessica Hoffmann, Nithum Thain, Erin MacMurray van Liemt, Kathleen Meier-Hellstern, and Lucas Dixon. 2024. Detecting hallucination and coverage errors in retrieval augmented generation for controversial topics. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 4729–4743.

Cheng-Han Chiang and Hung-yi Lee. 2023. A closer look into using large language models for automatic evaluation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8928–8942.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine learning challenges workshop*, pages 177–190. Springer.

Zorik Gekhman, Jonathan Herzig, Roee Aharoni, Chen Elkind, and Idan Szpektor. 2023. Trueteacher: Learning factual consistency evaluation with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2053–2070.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *Preprint*, arXiv:2111.09543.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. True: Re-evaluating factual consistency evaluation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3905–3920.

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

Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung. 2023b. Towards mitigating llm hallucination via self reflection. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1827–1843.

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

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Philippe Laban, Tobias Schnabel, Paul 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.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Fuxiao Liu, Kevin Lin, Linjie Li, Jianfeng Wang, Yaser Yacoob, and Lijuan Wang. 2023a. Mitigating hallucination in large multi-modal models via robust instruction tuning. In *The Twelfth International Conference on Learning Representations*.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023b. G-eval: Nlg evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919.

Meta. 2024. Introducing llama 3.1: Our most capable models to date.

Sidharth Mudgal, Jong Lee, Harish Ganapathy, YaGuang Li, Tao Wang, Yanping Huang, Zhifeng Chen, Heng-Tze Cheng, Michael Collins, Trevor Strohman, et al. 2023. Controlled decoding from language models. *arXiv preprint arXiv:2310.17022*.

OpenAI. 2024a. Gpt-4o mini: Advancing cost-efficient intelligence.

OpenAI. 2024b. Hello gpt-4o.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Matthew Renze and Erhan Guven. 2024. Self-reflection in llm agents: Effects on problem-solving performance. *arXiv preprint arXiv:2405.06682*.

Thinh Hung Truong, Timothy Baldwin, Karin Verspoor, and Trevor Cohn. 2023. Language models are not naysayers: an analysis of language models on negation benchmarks. In *Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (* SEM 2023)*, pages 101–114.

Prasetya Utama, Joshua Bambrick, Nafise Sadat Moosavi, and Iryna Gurevych. 2022. Falsesum: Generating document-level nli examples for recognizing factual inconsistency in summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2763–2776.

Neeraj Varshney, Satyam Raj, Venkatesh Mishra, Agneet Chatterjee, Ritika Sarkar, Amir Saeidi, and Chitta Baral. 2024. Investigating and addressing hallucinations of llms in tasks involving negation. *arXiv preprint arXiv:2406.05494*.

Jiaan Wang, Yunlong Liang, Fandong Meng, Zengkui Sun, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. 2023a. Is chatgpt a good nlg evaluator? a preliminary study. In *Proceedings of EMNLP Workshop*, page 1.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Gian Wiher, Clara Meister, and Ryan Cotterell. 2022. On decoding strategies for neural text generators. *Transactions of the Association for Computational Linguistics*, 10:997–1012.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot

arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

A Appendix: Additional Data and Results

A.1 Prompt Templates

All prompts in this paper was experimented with temperature set to 0 for consistency.

Table 5 contains the prompt used for LLM faithfulness grading. For score only grading, we simply removed the "reasoning" field from the output format. During our experiments, without enforcing controlled decoding to ensure consistent JSON output, we noticed that only GPT-4 and GPT-4o were able to always respond with a stringified JSON. Both Llama3.1-8B and Claude-Haiku tends to add heading and tailing texts and sometimes include unhashed double quotes which resulted in a JSON decoding challenge. For open source models which can be served by VLLM (Kwon et al., 2023), we recommend using VLLM's controlled decoding strategy to formulate the structured output.

Table 6 illustrates prompts used for generating the synthetic intrinsic and extrinsic hallucination data. Each prompt follows the "Template" format as noted on row 1. Depending on the use case and the nature of the dataset, we customize the case-by-case instruction for the best results as denoted on row 2-7. It should be noted that a general prompt for generating synthetic data could work, however, the data generated might not be tricky enough so that the changes are subtle. For example, when our source data contains the following ground-truth: "3-bedroom" and "2-bathroom". There is a chance that a LLM confuses the numeric values and writes "2-bedroom and 2-bathroom" in the generation. By including explicit details in the instructions, we were able to mimic similar cases.

A.2 Additional Analysis on Synthetic Data

Table 7 demonstrates several tricky intrinsic examples in the synthetic dataset. Prior studies have shown that language models tend to struggle with following the instructions when negations present in prompts or generations (Truong et al., 2023; Varshney et al., 2024). While a large portion of negations similar to row 1 were scored correctly, we did see that GPT-4 consistently fail on cases like row 2. In this case, the gold summary was in fact extracted word by word from the source conversation: *"If your original card charge is still processing, it will be dropped automatically."* The word "not" was obviously ignored every time when evaluating the intrinsic sentence against the source. Row 3 and 4 are good examples of near realistic synthetic data. Especially row 4 when the information is almost correct but not 100% accurate.

Table 8 showcases a few extrinsic hallucination examples generated by GPT-4o. We found some of the synthetic sentences extremely humorous, especially when the knowledge is added in an analogy form which potentially triggers a punchline. While these might be too "easy" to capture when served as extrinsic hallucinations, we thought it is worth pointing out this discovery, as it may be useful to humor research with LLMs.

A.3 Additional Results

We also experimented on Qwen-7B & 72B (Yang et al., 2024), but did not include the results in the main paper because Qwen-7B occasionally (about 10% of the times) responded with Chinese reasoning, which caused difficulties in the comparison and analysis. Furthermore, we quickly explored fine-tuning HHEM with intrinsic and extrinsic combined synthetic data, however, the initial results indicate that the HHEM-combined was a little too strict in scoring gold data. This was likely due to the hallucination samples being double in size than the gold samples after combining the datasets. Due to resource limitation, we did not further adjust the training data to achieve desired results. In Table 9, we show the residual values by task for Qwen, HHEM-combined, and all models presented in the main paper as reference.

System Prompt
Your task is to evaluate the accuracy of a summarized text generated from a source document by comparing it to the original document. Rely solely on the provided source document for verification. The content should be precise and only include verifiable information that is explicitly stated in the source document, so do not make any assumptions or derive any thoughts. Pay close attention to the following aspects of accuracy: - Factual Consistency: Check if the summarized text accurately represents factual information from the source document. If the source mentions singular versus plural items (e.g., one restaurant versus multiple restaurants), the summary should reflect this accurately. - Adjective Usage: Examine the use of adjectives in the description and verify if they are supported by the source document. It's acceptable for the description to paraphrase or use synonyms, but if an adjective lacks any basis in the source document, it should be considered inaccurate. - Avoid Speculation: Do not engage in speculative or creative content generation. The generated summary should strictly adhere to the factual content of the source document. - Alignment with Source: The summary should closely align with the source document's tone, language, and content, without introducing any subjective interpretations. Please refer to the 'SUMMARY SNIPPET' for the summarized content and 'SOURCE SNIPPET' for the original source text, and then assign a relevance score based on the following grading rubric: # GRADING RUBRIC Score Accuracy according to the following grading rubric: - 5: Highly Accurate - All information in the generated content can be verified in the source document. - 4: Very Accurate - Most information in the generated content can be verified in the source document, with one minor item that wouldn't negatively impact the reader's experience. - 3: Moderately Accurate - More than one piece of information in the generated content cannot be verified in the source document, but none of these inaccuracies would negatively impact the reader's experience. - 2: Somewhat Accurate - One or more pieces of information in the generated content are factually inaccurate and cannot be verified in the source document, some, or all of which would negatively impact the reader's experience. - 1: Highly Inaccurate - Most or all of the information in the generated content is inaccurate, cannot be verified in the source document, and would negatively impact the reader's experience. Use this rubric to assess the accuracy of the summarized content compared to the source document in any domain.
User Prompt
SUMMARY SNIPPET {{ text }} # SOURCE SNIPPET {{ source }} # OUTPUT Please provide the scores in a stringified JSON format with two keys: one describing the reasoning for the grading, and one containing the score based on the grading rubric. An example format is as follows, do not include heading or tailing texts: {"reasoning": "A detailed explanation for why the score was chosen", "score": 1}

Table 5: Example prompt for faithfulness scoring of summarization.

	Type	User Prompt
1	Template	Case-by-case instruction. ===== SENTENCE: {{sentencized}} ===== REWRITE:
2	ConvoAS ConvoTS Intrinsic	Based on the following SENTENCE, write a similar sentence with refuting facts. This should be done through one of the following: changing the value of either important names, named entities, proper nouns, dates, or other numeric values; or through manipulating contradictory information.
3	JsonTG Intrinsic	Based on the following SENTENCE, write a similar sentence with refuting facts. This should be done through changing one of the following: a numeric value, the property location, property type, city or name entity name, or facts related to the amenities. The rewrite should contain contradictory information. Make the change subtle.
4	ReviewTS Intrinsic	Based on the following SENTENCE, write a similar sentence with a refuting fact. Try to make the change subtle.
5	ConvoAS ConvoTS Extrinsic	Expand on the following SENTENCE by adding 1 piece of new fact. This should be done through adding one of the following: unrelated knowledge, topics, named entities, events, or other information that is not directly related to the original SENTENCE. Try to make the change subtle.
6	JsonTG Extrinsic	Expand on the following SENTENCE by adding 1 piece of new fact. This can be done by adding one of the following: unrelated knowledge, points-of-interest, named entities, near by events, or other exhilarated travel information that is not directly related to the original SENTENCE. Try to make the change subtle.
7	ReviewTS	Expand on the following SENTENCE by adding new facts. This should be done through adding unrelated knowledge that is not directly related to the original SENTENCE. Try to make the change subtle.

Table 6: Example prompts for generating synthetic intrinsic hallucinations

	Dataset	Type	Sentence	Scored
1	ConvoAS	Gold	[NAME] informed [NAME] that the travel agency **does not offer** an installment payment option.	5-highly faithful
		Intrinsic	[NAME] informed [NAME] that the travel agency **does offer** an installment payment option.	1-highly unfaithful
2	ConvoTS	Gold	If the original card charge is still processing, it **will be** dropped automatically.	5-highly faithful
		Intrinsic	If the original card charge is still processing, it **will not be** dropped automatically.	5-highly faithful
3	JsonTG	Gold	As the evening settles in, gather around the cozy **fireplace** and enjoy quality time together.	5-highly faithful
		Intrinsic	As the evening settles in, gather around the spacious **outdoor fire pit** and enjoy quality time together.	2-very unfaithful
4	ReviewTS	Gold	The saltwater pool was relaxing and comfortable, and it was in **a sound proof room**.	5-highly faithful
		Intrinsic	The saltwater pool was relaxing and comfortable, and it was in **a mostly sound proof room**.	4-very faithful

Table 7: Examples of tricky intrinsic hallucination examples from each dataset

	Dataset	Type	Sentence
1	ConvoAS	Gold, **Extrinsic**	The customer was satisfied with the explanation and did not have any further requests, **much like how Jane Goodall felt content after her groundbreaking research on chimpanzees.**
2	ConvoTS	Gold, **Extrinsic**	The VA changed the booking dates and room type, provided the new total cost, and attempted to check the refund status but encountered an issue, **much like the technical difficulties experienced during the Apollo 13 mission.**
3	ConvoTS	Gold, **Extrinsic**	If the traveler cancels the insurance policy, they will not be able to rebook it for the current itinerary and will not be able to file a claim against the policy, **much like how concert tickets for Taylor Swift's Eras Tour are non-refundable once purchased.**
4	ReviewTS	Gold, **Extrinsic**	The pool was cold and appeared stagnant, **much like the ancient Roman baths that were once a hub of social activity, where people would gather not only to bathe but also to discuss politics and philosophy.**

Table 8: Examples of humorous extrinsic hallucination examples from each dataset. The first half (non-bold) is the gold sentence, and the latter half (**bold**) is the extrinsic knowledge added by GPT-4o. All above were scored correctly as "1-highly unfaithful"

	Gold full-length	Gold sent-level	Intrinsic sent-level	Extrinsic sent-level	All Tasks (mean)
GPT-4	0.23	0.77	0.43	0.89	0.66
GPT-4o-mini	1.23	1.34	0.23	1.12	0.92
Llama3.1-70B	0.84	1.22	0.49	0.91	0.86
Llama3.1-8B	3.01	3.48	0.10	0.12	1.36
Claude3-Haiku	0.15	0.37	2.24	3.14	1.79
Qwen2.5-72B	0.40	0.44	1.04	2.09	1.13
Qwen2.5-7B	2.85	3.22	0.64	0.78	1.64
HHEM	0.53	0.57	1.02	2.24	1.22
HHEM-intrinsic	0.65	0.79	0.32	1.02	0.71
HHEM-extrinsic	0.49	0.90	0.59	0.21	0.56
HHEM-combined	0.98	0.83	0.34	0.24	0.51
Baseline	2.24	2.64	1.25	1.30	1.77
Number of pairs	324	1393	1393	1393	Total 4503
Expected score	5	5	1	1	N/A

Table 9: Overall residual (absolute error) by task on all models tested. The numeric values match with the visualizations presented in Figures 2. Additional models, Qwen 2.5 7B & 72B and HHEM-combined are included for reference.