Chain of Natural Language Inference for Reducing Large Language Model Ungrounded Hallucinations

Deren Lei[*], **Yaxi Li**[*], **Mengya (Mia) Hu**[*], **Mingyu Wang**[*], **Vincent Yun**, **Emily Ching**, **Eslam Kamal**
Microsoft Responsible AI
{derenlei, yaxi.li, humia, mwang, xi.yun, yuetc, eskam}microsoft.com

Abstract

Large language models (LLMs) can generate fluent natural language texts when given relevant documents as background context. This ability has attracted considerable interest in developing industry applications of LLMs. However, LLMs are prone to generate hallucinations that are not supported by the provided sources. In this paper, we propose a hierarchical framework to detect and mitigate such ungrounded hallucination. Our framework uses Chain of Natural Language Inference (CoNLI) for hallucination detection and hallucination reduction via post-editing. Our approach achieves state-of-the-art performance on hallucination detection and enhances text quality through rewrite, using LLMs without any fine-tuning or domain-specific prompt engineering. We show that this simple plug-and-play framework can serve as an effective choice for hallucination detection and reduction, achieving competitive performance across various contexts. [2]

1 Introduction

Large Language models, known for their remarkable capabilities in natural language generation (NLG) [1–3], have attracted unprecedented interest from the public. These models serve as the foundation for a wide array of business applications (*e.g.*Bing.com, ChatGPT, and Github Copilot). A common characteristic of such applications is their reliance on LLMs for text-to-text generation, often necessitating that the generated responses maintain factual consistency with the source text. Therefore, ensuring factual consistency is a critical challenge when evaluating the quality of generated responses [4, 5]. However, generating hallucination that diverges from the source text is a well-known phenomenon of LLMs. These hallucinations can be attributed to various factors, such as long input context [6], irrelevant context distraction [7], or complicated reasoning [8]. This phenomenon poses a significant challenge to the reliability of LLMs in real-world applications.

Hallucination is commonly categorized as: *context-related hallucination*, refers to hallucination where generated response contradicts commonsense; *self-conflicting hallucination*, where generated response sentences conflict with each other (*e.g.*numerical multi-step reasoning failed at a particular step [9, 10]); and *ungrounded hallucination*, where generated sentences conflict with the source text [11] without assessing response coherence. Self-conflicting hallucination is more solution-dependent and behaves differently per downstream task. To generically enhance the reliability of LLM responses, our investigation focuses on reducing ungrounded hallucination, irrespective of the upstream task. We define alignment level with source as **groundedness** of LLM output.

Numerous existing works have concentrated on evaluating the groundedness of generated texts by developing classification [12–14] or ranking [15] models. While these detection models are useful in

[*] Equal contributions.

[2] `https://github.com/microsoft/CoNLI_hallucination`

Workshop on Instruction Tuning and Instruction Following at NeurIPS 2023.

assessing groundedness, they provide limited utility in terms of rewriting and enhancing groundedness of a given LLM response.

Recent studies have explored methods for enhancing groundedness of LLM responses, including changing decoding strategy [16], inference-time self-critique [17, 18], multi-agent debate [19], and user-specified retrieval corpus [20]. In contrast, we study how to reduce hallucination when the user does not have full control over the LLM model or cannot leverage additional external knowledge. We propose a generic post-edit approach, named **Chain of Natural Language Inference (CoNLI)**. In this framework, users are only required to bring their own text-to-text inputs/outputs and an LLM API endpoint. It will (1) select sentences as claims, (2) detect hallucination hierarchically with sentence-level and entity-level detectors (with a given entity detection model) by asking LLM to solve a sequence of natural language inference problems, and (3) leverage detection response in hallucination mitigator to get a refined response. We conducted experiments with CoNLI on text abstractive summarization and grounded question-answering scenarios with the latest hallucination benchmarks, both synthetic-generated and human-annotated. Our proposed approach demonstrates hallucination detection improvement against the latest solutions. Furthermore, the final refined responses show improvements over the initial provided response on various NLG evaluation metrics and groundedness metrics. Our interpretable and high-quality hallucination detection and reduction framework utilizes domain-agnostic few shots with simple post-editing techniques that prioritize the preservation of the original raw responses. We claim that our proposed framework is a generic solution that can potentially benefit various LLM-based business applications.

2 Problem and preliminaries

Previous research has encompassed different problem definitions and terminologies, often blending together aspects such as judging the correctness of text in various contexts, including free-text generation and text-to-text generation. Terminologies such as hallucination [12, 18, 21], attribution [20], factual consistency [14, 22], factuality [23], factual correctness [24], faithfulness [4, 25], and truthfulness[26]. In contrast, our focus exclusively centers on ungrounded hallucination, a phenomenon prevalent in text-to-text generation scenarios. It refers to any erroneous text generated by models that either conflict with or cannot be verified against the source texts.

For text-to-text generation, we denote the input *source text* as X and the output *raw response* as Y_{raw}, where X and Y_{raw}, represented as $X = \{x_1, x_2, ..., x_m\}$ and $Y_{\text{raw}} = \{y_1, y_2, ..., y_n\}$ respectively, comprise one or more sentences. The generation can thus be denoted as:

$$\boldsymbol{\mathcal{F}} : X \rightarrow Y_{\text{raw}} \tag{1}$$

In contemporary approaches, $\boldsymbol{\mathcal{F}}(\cdot)$ is primarily powered by the language model. We say Y_{raw} is grounded by X if a generic reader would affirm the statement "According to X, Y_{raw} is true" [27]. Conversely, Y_{raw} is hallucinated with respect to X if it conflicts with or cannot be verified against X.

Our objective is to detect and minimize ungrounded hallucination in Y_{raw}. Importantly, we do not assume direct access to the generation model and hence do not modify $\boldsymbol{\mathcal{F}}(\cdot)$. Instead, we post edit Y_{raw} into a refined response Y_{refined}, such that Y_{refined} exhibits reduced hallucination while retaining the essence of Y_{raw}.

3 Methodology

Our solution is a two-stage framework, comprising a *detection agent* and a *mitigation agent* illustrated in Figure 1 using an example. We provide in-depth discussion of each agent in below sections.

3.1 Detection agent

We formally define $\mathcal{H}_{\text{selected}} = \{hyp_1, hyp_2, ..., hyp_n\}$ as a set of selected hypotheses from Y_{raw} for detection; $\mathcal{R} = \{r_1, r_2, ..., r_n\}$ as set of reasons against each hypothesis, $\mathcal{J} = \{\text{hallucination, non\_hallucination}\}$ is the final judgement for a hypothesis, further divides into elementary events $\mathcal{J}^+ = \{\text{hallucination}\}$, $\mathcal{J}^- = \{\text{non\_hallucination}\}$. $\mathcal{O}$ is the output of detection agent. Therefore, detection agent can be formulated as:

$$\boldsymbol{\mathcal{D}} : (X, Y_{\text{raw}}) \rightarrow \mathcal{O} \tag{2}$$

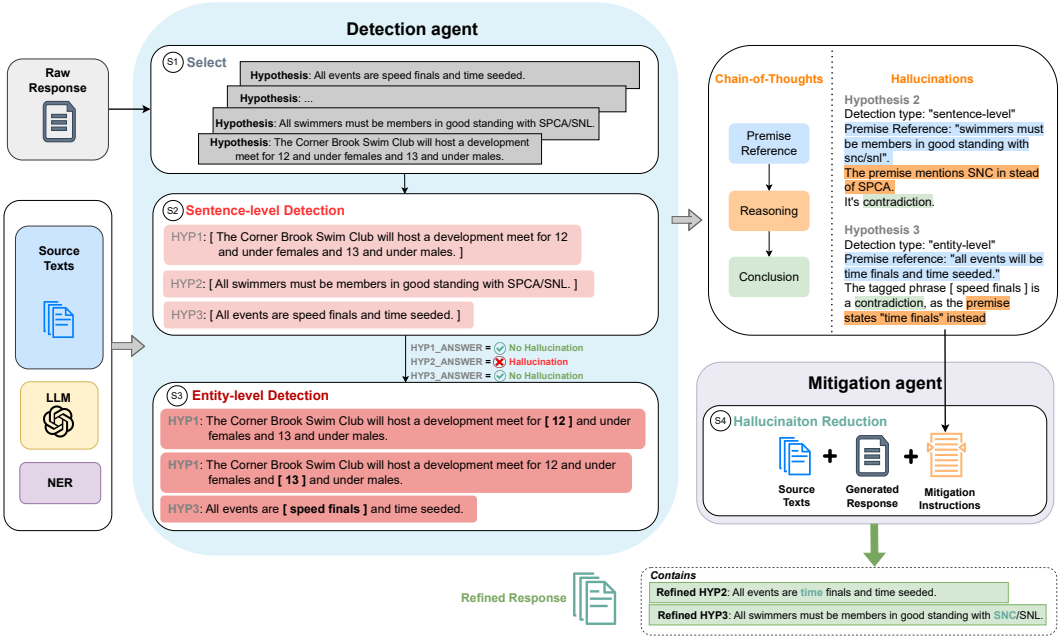

Figure 1: **Illustration of the proposed framework CoNLI with a real example.** Each hypothesis in the raw response will first go through sentence-level detection. If no hallucination is detected, it will go to detailed entity-level detection. Detection reasonings will be used as mitigation instructions.

$$\mathcal{O} = \{(hyp_i, r_i, j_i)\} \subseteq \mathcal{H}_{\text{selected}} \times \mathcal{R} \times \mathcal{J} \tag{3}$$

where we break down $\boldsymbol{\mathcal{D}}(\cdot)$ hierarchically into sentence-level detection $\boldsymbol{\mathcal{D}}_{\text{sent}}(\cdot)$ and entity-level detection $\boldsymbol{\mathcal{D}}_{\text{ent}}(\cdot)$ described in below paragraphs. In Addition, given $\mathcal{J}$ is a pair set, this detection phase can be treated as a binary classification. Beyond serving as a precursor to mitigation agent, this module can be independently utilized to evaluate the groundedness of raw response in text-to-text generation applications. Detection agent contains the following steps.

Split and select Each raw response Y_{raw} is segmented into individual sentences using the NLTK sentence splitter[3]. Sentences that are considered noise or lack factual information for judgement are then purged. For benchmark comparison purposes, we skip this purging process for short-generated responses that can be directly formulated as hypotheses. We leave building advanced hypothesis selector as future work. After this step, we have hypotheses set $\mathcal{H}_{\text{selected}}$.

Sentence-level detection To formulate the NLI problem, we treat the X as the premise for hypotheses $\mathcal{H}$. The sentence-level detection will sequentially judge each hypothesis independently against the corresponding premise, and categorize them as entailment, contradiction or neutral following [28]:

- *Entailment*: $X \implies hyp_i$
- *Contradiction*: $X \implies \neg hyp_i$
- *Neutral*: $X \not\Longrightarrow hyp_i$

In the ungrounded hallucination scenario, both contradiction and neutral categories in NLI are not aligned with the source, so we treat these two categories as hallucinations. Therefore:

$$\boldsymbol{\mathcal{D}}_{\text{sent}} : (X, \mathcal{H}_{\text{selected}}) \rightarrow \mathcal{O}_{\text{sent}} \tag{4}$$

$$\mathcal{O}_{\text{sent}} = \{(hyp_i, r_i^{\text{sent}}, j_i^{\text{sent}})\} \subseteq \mathcal{H} \times \mathcal{R}_{\text{sent}} \times \mathcal{J} \tag{5}$$

We divide $\mathcal{O}_{\text{sent}} = \mathcal{O}_{\text{sent}}^{+} \cup \mathcal{O}_{\text{sent}}^{-}$ where hallucination detection output $\mathcal{O}_{\text{sent}}^{+} \subseteq \mathcal{H}_{\text{sent}}^{+} \times \mathcal{R}_{\text{sent}}^{+} \times \mathcal{J}^{+}$ and non-hallucination detection output $\mathcal{O}_{\text{sent}}^{-} \subseteq \mathcal{H}_{\text{sent}}^{-} \times \mathcal{R}_{\text{sent}}^{-} \times \mathcal{J}^{-}$.

[3] `https://www.nltk.org/api/nltk.tokenize.sent_tokenize.html`

We utilize Chain-of-Thought (CoT) prompting [8], guiding the LLM to locate relevant passages in the source text X and allow it to reason and then make a conclusion. To enhance adaptability across domains without intricate prompt engineering, we employ domain-agnostic NLI few-shot examples to orient the LLM towards the essential NLI concepts and the CoT methodology. The specific prompt used in our experiments is detailed in Appendix D. Note that in the few-shot examples, with a given premise, we provide multiple hypotheses and CoT answers in the form of bullet points. This is for batching support so that we may send multiple claims in a single prompt to make our solution more cost-efficient. For benchmarking experiments mentioned in the below sections, we maintain the few-shot examples but disable batching, sending one claim for judgment at a time for apples-to-apples comparison with the other approaches.

Entity-level detection Upon sentence-level evaluation, hypotheses deemed as non-hallucinations undergo subsequent entity-level inspections. This is based on our empirical findings that LLMs, when doing NLI reasonings, may potentially overlook details in the hypothesis and focus more on surface-level semantic features for judgments. If a hypothesis contains abundant factual details or some details require complex reasoning against the source text, sentence-level detection may reach false negative conclusions. Hence, we use entity-level detection to take another look into the non-hallucinated hypothesis $\mathcal{H}^-_{\text{sent}}$ in $\mathcal{O}^-_{\text{sent}}$.

Specifically, it will first leverage an entity recognition model (NER) to find entities in the non-hallucinated hypothesis $E = \text{NER}(\mathcal{H}^-_{\text{sent}})$. Then it will convert each hypothesis into a sequence of hypothesis where each of them contain a tagged entity to focus on:

$$\mathbf{f} : \text{hyp}_i \rightarrow \{\text{hyp}_i^e\}, e \in E \tag{6}$$

However, unlike $\boldsymbol{\mathcal{D}}_{\text{sent}}$, $\boldsymbol{\mathcal{D}}_{\text{ent}}$ will focus only on the tagged entity without needing to judge other factual information of a hypothesis. This forces the LLM to reason and make judgments against every entities in the non-hallucination hypothesis output by sentence-level detection. If a single $\text{hyp}_i^e \in \text{hyp}_i^E$ is judged as hallucination, we say entity-level judges hyp_i as hallucination.

$$\boldsymbol{\mathcal{D}}_{\text{ent}} : (X, \{\text{hyp}_i^e\}) \rightarrow \mathcal{O}_{\text{ent}} \tag{7}$$

$$\mathcal{O}_{\text{ent}} = \{(hyp_i, r_i^{\text{ent}}, j_i^{\text{ent}})\} \in \mathcal{H}^-_{\text{sent}} \times \mathcal{R}_{\text{ent}} \times \mathcal{J} \tag{8}$$

Merging For each sentence in the generated response, detection agent's final judgment will be $\mathcal{O} = \mathcal{O}^+_{sent} \cup O_{ent}$. For each tuple $\{(hyp_i, r_i, j_i)\}$ in $\mathcal{O}$ where j_i = hallucination, r_i is either a single sentence-level is-hallucination reason or single/multiple entity-level reasons. In other words, a hypothesis will be judged as non-hallucination only if overall sentence judgment and tagged entities judgments all vote for non-hallucination.

3.2 Mitigation agent

Mitigation agent can be formulated $\boldsymbol{\mathcal{M}} : (X, Y_{\text{raw}}, \mathcal{O}) \rightarrow Y_{\text{refined}}$. We consider the hallucination detection result $\mathcal{O}$ as crucial guidance for mitigation agent to reason on how to rewrite these hypothesis sentences and address issues provided by detection agent. We directly leverage $\mathcal{O}$ as instructions to rewrite Y_{raw}. Mitigation agent tries to preserve the format of the generated response to the greatest extent possible. It strictly trusts and follows the instructions from detection agent without engaging in additional reasoning on hallucinations. As a result, it could solely focus on how to maintain the fluency and coherency of refined responses by choosing whether to remove or rewrite the hallucination hypothesis sentences. The prompt used can be found in Appendix E.

4 Experiments

We break down our experiments into two parts. For hallucination detection experiments, we analyze our detection agent's ungrounded hallucination detection performance on various benchmarks and compare it with existing LLM-based and model-based approaches to check our detection quality. For hallucination reduction experiments, we then leverage detection agent's output to do hallucination reduction via mitigation agent on the same benchmarks and do before/after comparisons with text-to-text and hallucination metrics. We try to answer the following two questions:

Algorithm 1: CoNLI hallucination detection and mitigation

Input: the source text X and the graw response Y_{raw} from a text-to-text application
Output: refined response with reduced hallucination Y_{refined}

/* *Detection agent process*/*
$\{hyp_1, ..., hyp_n\}$ = HypothesesSelector(Y_{raw})
for $i = 1$ *to* n **do**
  if hyp_i *fits the hypothesis selection requirements* **then**
    (hyp_i, r_i^{sent}, j_i^{sent})= $\boldsymbol{\mathcal{D}}_{\text{sent}}(X, \text{hyp}_i)$
    if j_i^{sent} == *non_hallucinated* **then**
      $E = \text{NER}(hyp_i)$
      for e *in* E **do**
        $\mathcal{O}[\text{hyp}_i]$ += $\boldsymbol{\mathcal{D}}_{\text{ent}}(X, \text{hyp}_i^e)$
    else
      $\mathcal{O}[\text{hyp}_i] = (hyp_i, r_i^{\text{sent}}, j_i^{\text{sent}})$
  else
    $\mathcal{O}[\text{hyp}_i] = (hyp_i, null, non\_hallucination)$
/* *Mitigation agent process*/*
Y_{refined} = Mitigation(X, Y_{raw}, $\mathcal{O}$)
return Y_{refined}

Q1 (Detection)*: How does the performance of our CoNLI detection agent compare to LLM-based and model-based hallucination detection methods?*

Q2 (Detection and reduction)*: Does applying CoNLI with hallucination reduction lead to improvements on on NLG and groundedness metrics compared to raw response?*

4.1 Hallucination detection experiments

We conduct experiments on ungrounded hallucination detection with our detection agent.

4.1.1 Datasets

We conduct experiments on two different kinds of datasets: (1) datasets with synthetic hallucination generated on ground truth response text. They have larger dataset sizes with defined hallucination categories for easy analysis. (2) datasets with hallucination annotated manually on real state-of-the-art (SOTA) NLG model output response text. They are smaller than the synthetic data, but their hallucinations are closer to hallucinations found in LLM real-world products.

For synthetic datasets, we use a recent LLM hallucination evaluation benchmark HaluEVAL [21]. We only use summarization and question answering datasets in HaluEval as they contain grounding source texts. We also conducted experiments using annotated datasets traditionally employed for evaluating factual consistency metrics. These datasets include FactCC's summarization test set [13, 29], SummEval [30], QAGS-Xsum [22], QAGS-CNNDM [22]. Conventional factual consistency evaluation approaches output consistency scores and use Spearman Correlation coefficients, ROC-AUC [31] for evaluation. In our defined groundedness scenario, we consider hallucination as a binary question. Therefore, we use F1 to uniformly evaluate both hallucination evaluation and factual consistency evaluation datasets. We selected a subset of HaluEval benchmark with details mentioned below and factual consistency evaluation datasets we use the same setting following previous works [14, 32]. Dataset statistics can be found in Table 1.

HaluSum2130 subset of HaluEval [21] summarization dataset. Each source text contains a pair of hallucination and non-hallucination summaries. For cost concerns of running LLM experiments, we randomly select samples and also filter potentially harmful and sensitive (*i.e.*hate, sexual, violence, self-harm) samples to support the recent trend of building responsible LLM.[4]

HaluQA4170 subset of HaluEval [21] question answering dataset that each source text also contains a pair of hallucination and non-hallucination answers. Similarly, we do a random sample with content

[4] `https://azure.microsoft.com/en-us/products/ai-services/ai-content-safety`

Table 1: **Dataset statistics.** We conduct separate experiments on two distinct types of datasets: datasets with synthetic hallucination and substantial dataset size; datasets with hallucination annotated on SOTA NLG model outputs, smaller but closer to application scenarios.

Type	Dataset	Total#	Hallucination#	Non_hallucination#
Synthetic Hallucination	HaluSum2130 [21]	2130	1065	1065
	HaluQA4170 [21]	4170	2085	2085
Annotated Hallucination on SOTA Model Output	FactCC503 [13]	503	62	441
	SummEval [30]	1600	294	1306
	QAGS-CNNDM [22]	235	122	113
	QAGS-XSUM [22]	239	123	116

filtering applied. To adapt question answering into our proposed NLI approach, we treat each source text as premise and its associated answer as hypothesis, ignoring the question and answer correctness. That is, an associated answer can still be considered as grounded to the source regardless of the correctness or relevance to the question.

FactCC503 is the FactCC [13] test set that contains source text and summary sentence pairs. Each summary associated with a source text is generated by SOTA models and then broken down into sentences with poorly generated sentences removed [13]. Each sentence is annotated as hallucination or non-hallucination.

SummEval and QAGS SummEval contains 1600 examples built on CNN/Dailymaill [33] with consistency score labeled between 0 to 5. QAGS datasets are built with CNN/Dailymaill [33] (QAGS-CNNDM) and XSUM [34] (QAGS-XSUM) respectively with consistency scores between 0 to 1. Unlike past consistency studies, we consider hallucination as a yes or no question for detection and reduction purposes. Therefore, we convert the labels of these datasets into a binary. Only maximum consistency samples are considered as non-hallucination and all the rest are considered as hallucination. All hallucinations are manually annotated on recent SOTA models' outputs.

4.1.2 Experimental setup

LLM setup and hyperparameters We evaluate our framework on OpenAI's GPT-3.5-TURBO-16K with max input tokens 16,384 and GPT-4-32K with max input tokens 32,768. We leverage Azure OpenAI ChatGPT API to conduct the experiments.[5] We set the temperature to 0 to reduce randomness and ensure more deterministic outputs. We set the maximum number of tokens for generation to 4096, top_p to 0.6, and freq_penalty and presence_penalty both to 0.

Entity detection setup For the NER in entity-level detection, we leverage Azure Text Analytics (TA) API for entity detection which supports a wide range of entity categories.[6] Among all the available entity categories, we select the best collection of 9 entities based on the average performance on available validation datasets. Although we observe each experiment dataset has its own best TA categories, to make CoNLI generalizable, we use the same TA categories for all detection and mitigation experiments. See Appendix B for more details on the selected TA categories.

Evaluation metrics We used F1 since we define our groundedness task as a binary classification. LLM-based hallucination detection approaches usually output binary predictions, while factual consistency evaluation approaches usually output multi-level scores for finer-grained evaluation. Using F1 can unify the measurement for both. We report the macro F1 as well as its breakdowns on hallucination and non-hallucination since the hallucinations can be skewed as per Table 1.

4.1.3 Results

Synthetic hallucination dataset results We show the results in Table 2. FactCC and AlignScore are classification models that use alignment output logits as factual consistency scores. We adopt the threshold of 0.5 as the cut-off point for hallucination/non-hallucination predictions, since both are off-the-shelf solutions that aim to be generic with no necessity of downstream fine-tuning. To determine

[5] https://azure.microsoft.com/en-in/products/ai-services/openai-service/

[6] https://azure.microsoft.com/en-us/products/ai-services/text-analytics

their performance upper-bound, we also investigate their oracle thresholds that best performed on experimented datasets. Notably, the oracle threshold diverges from one dataset to another (see Appendix C).To establish a unified threshold for generalization, we select the average oracle threshold that yields the highest average F1-macro across all 6 experimented datasets, ensuring a balanced and consistent assessment.

In the case of HaluEval, its provided detection solutions are not task agnostic but designed per their own dataset. Thus we run with its best settings tailored to its own synthetic datasets and skip experiment on annotated hallucination dataset. When running HaluEval, we observed a significant divergence in the behavior of GPT-4 compared to GPT-3.5. GPT-4 exhibited challenges in comprehending the few-shot labels as instructed, resulting in unexpected large performance drops. To mitigate this issue we made an adjustment by appending an additional sentence to the original prompts, which explicitly instructs GPT4 as follows: "for hallucination answer Yes and for non-hallucination answer No". This clarification ensures more accurate performance of HaluEval-GPT4 (*).

We observed that our CoNLI-GPT4 achieves the best F1 on both datasets and averages. It even surpasses AlignScore-Large with upper-bound oracle threshold. Our CoNLI-GPT3.5 achieves the second best averaged F1 and outperforms all listed solutions except those with oracle.

Table 2: **Synthetic hallucination dataset results** on F1-macro and breakdown on F1-Hallucination and F1-non_Hallucination. The last column **AVG** is the average performance of each metric. Dark green indicates best metric and light green indicates second best on each dataset or average. (*) details addressed in section 4.1.3

	HaluSum2130			HaluQA4170			**AVG**		
Method	**F1**	**-NonHal**	**-Hal**	**F1**	**-NonHal**	**-Hal**	**F1**	**-NonHal**	**-Hal**
FactCC	0.421	0.224	0.618	0.485	0.412	0.558	0.453	0.318	0.588
FactCC (Oracle)	0.437	0.272	0.602	0.482	0.429	0.535	0.460	0.351	0.569
AlignScore-L	0.617	0.723	0.510	0.783	0.796	0.770	0.700	0.760	0.640
AlignScore-L (Oracle)	0.669	0.665	0.672	0.750	0.743	0.756	0.710	0.704	0.714
HaluEval-GPT3.5	0.535	0.699	0.371	0.679	0.736	0.622	0.607	0.718	0.497
HaluEval-GPT4	0.415*	0.683*	0.147*	0.796*	0.827*	0.764*	0.606*	0.755*	0.456*
CoNLI-GPT3.5	0.633	0.641	0.624	0.848	0.850	0.845	0.741	0.746	0.735
CoNLI-GPT4	0.677	0.725	0.628	0.849	0.862	0.835	0.763	0.794	0.732

Annotated hallucination dataset results Shows in Table 3. CoNLI-GPT4 achieves the best results on three datasets and averaged, and only underperforms AlignScore-Large averaged with oracle threshold on QAGS-CNNDM. This demonstrates CoNLI, as a generic solution, can achieve high-quality performance in detecting hallucinations in SOTA NLG model outputs. It's also worth mentioning that despite being a much smaller model comparing to GPT-4, AlignScore-Large can also achieve decent performance when an oracle threshold for binary classification is provided. This aligns with its reported high performance on factual consistency evaluation datasets using AUC-ROC and Spearman Correlation coefficients as measurement metrics. Consequently, we think the exploration of finding automatic threshold per task without fine-tuning is an interesting topic for evaluation-score-based approaches. Such study could enhance the applicability of score-based methods to a boarder range of hallucination detection and reduction applications that require a binary answer.

Table 3: **Annotated hallucination dataset results** on F1-macro and breakdown on F1-Hallucination and F1-non_Hallucination. We report their results with classification threshold of 0.5 and of best average across 6 datasets. The last column **AVG** is the average performance of each metric.

	FactCC503			SummEval			QAGS-CNNDM			QAGS-XSUM			**AVG**		
Method	**F1**	**-NonHal**	**-Hal**	**F1**	**-NonHal**	**-Hal**	**F1**	**-NonHal**	**-Hal**	**F1**	**-NonHal**	**-Hal**	**F1**	**-NonHal**	**-Hal**
FactCC	0.706	0.919	0.493	0.641	0.819	0.462	0.688	0.664	0.712	0.644	0.635	0.653	0.700	0.759	0.580
FactCC (Oracle)	0.710	0.923	0.496	0.651	0.833	0.468	0.689	0.678	0.700	0.649	0.653	0.644	0.674	0.772	0.577
AlignScore-L	0.820	0.952	0.687	0.656	0.917	0.395	0.549	0.701	0.397	0.723	0.760	0.686	0.695	0.833	0.541
AlignScore-L (Oracle)	0.765	0.923	0.606	0.753	0.919	0.586	0.829	0.837	0.821	0.745	0.755	0.734	0.773	0.859	0.687
CoNLI-4	0.876	0.971	0.780	0.784	0.935	0.632	0.799	0.814	0.783	0.812	0.819	0.804	0.818	0.885	0.750

Ablation study We run different variants of CoNLI on the HaluSum2130, HaluQA4170 and FactCC503. Results are presented in Table 4. For entity-detection-only approach, we run entity

detection on all hypothesis. For the default hierarchical approach, entity-level detection is only triggered on hypotheses where no hallucination is detected at sentence-level.

We observe that both sentence-level and entity-level detection results consistently underperform when compared to the combined hierarchical approach. Furthermore, sentence-level results consistently outperform entity-level results, which is logical since entity-level detections within each hypothesis focus solely on tagged entities, whereas sentence-level detection considers the entire hypothesis. Therefore, entity-level detection can be viewed as a valuable augmentation to the sentence-level detector. These findings hold true for both GPT-3.5 and GPT-4 settings.

Table 4: **Ablation study for hallucination detection.** We compare CoNLI with sentence-level detection only (sent), entity-level detection only (ent) and hierarchical detection (sent + ent) on GPT3.5 and GPT4.

	HaluSum2130			HaluQA4170			FactCC503		
Method	**F1**	**-NonHal**	**-Hal**	**F1**	**-NonHal**	**-Hal**	**F1**	**-NonHal**	**-Hal**
CoNLI-3.5 (sent)	0.628	**0.737**	0.519	0.809	0.824	0.793	0.668	0.931	0.404
CoNLI-3.5 (ent)	0.647	0.692	0.601	0.783	0.820	0.745	0.652	0.909	0.394
CoNLI-3.5 (sent+ent)	**0.664**	0.695	**0.632**	**0.840**	**0.845**	**0.834**	**0.694**	**0.933**	**0.455**
CoNLI-4 (sent)	0.666	**0.753**	0.578	0.832	0.850	0.813	0.858	0.968	0.748
CoNLI-4 (ent)	0.667	0.738	0.595	0.771	0.817	0.724	0.834	0.964	0.704
CoNLI-4 (sent+ent)	**0.677**	0.725	**0.628**	**0.844**	**0.859**	**0.829**	**0.876**	**0.971**	**0.780**

4.2 Hallucination reduction experiments

In this section, we conduct experiments on evaluating our CoNLI performance end-to-end with detection agent and mitigation agent combined. We used the same LLM setup and hyperparameters as detection expeirment mentioned in section 4.1.2.

4.2.1 Experimental setup

Datasets As a subsequent experiment in the context of hallucination detection detection, we continue to use HaluSum2130, HaluQA4170 synthetic datasets to experiments at larger scale. Additionally, we incorporate the human-annotated FactCC503 dataset, which encompasses hallucinations from a diverse set of 10 SOTA NLG models, making it the most comprehensive among the annotated hallucination datasets mentioned.

For HaluSum2130 and HaluQA4170, we use the non-hallucination summary as the ground truth for non-hallucination summaries. In the case of the FactCC503, we aggregate sentence-level summarization data into comprehensive summary. Subsequently, we apply our detection agent judgment on a per sentence basis to refine the complete summary and compare to the ground truth summary.

Evaluation metrics We evaluate text response quality in conventional NLG metrics Rouge1, Rouge2, RougeL, Bleu-4, BertScore [35] and hallucination evaluation metrics FactCC [13] and AlignScore-Large [14]. Furthermore, We use our proposed CoNLI-GPT4 for hallucination evaluation, leveraging its demonstrated high quality in the preceding hallucination detection experiments. For each dataset, the CoNLI-GPT4 score demonstrates the percentage of refined responses containing zero ungrounded hallucination by its detection.

4.2.2 Results

We show the hallucination reduction results with before and after CoNLI applied in Table 5. For synthetic datasets, HaluSum2130 and HaluQA4170, all metrics improved with CoNLI refined response. Responses in question answering datasets are shorter compared to those in summarization datasets. As a result, minor refinements have a more pronounced impact on the evaluation metrics.

In the annotated dataset, FactCC503, we observed a distinct pattern. Given that the raw responses are selected from state-of-the-art NLG models trained to optimize NLG metrics, especially Rouge scores, we noticed a slight decline in Rouge scores after the refinement process. However, it's important to note that this decline in Rouge scores does not necessarily indicate a drop in response quality, because we also observed improvements in BertScore and Bleu score. As Rouge score is more recall focused

(*i.e.*amount of n-grams in reference appears in generated response) and Bleu score is more precision focused (*i.e.*amount of n-grams in generated response appears in reference), Bleu score improvement means irrelevant tokens in responses are reduced, indicating a reduction in hallucinatory content. This hypothesis aligned with the consistent improvement on hallucination evaluation metrics, FactCC, AlignScore-Large and CoNLI-GPT4. Therefore, our CoNLI refinement process maintains response quality while effectively reducing hallucinations in the outputs of SOTA NLG models.

Table 5: **Hallucination reduction result.** We compare CoNLI refined response with raw generated response on various NLG and hallucination metrics.

Dataset	Target	Rouge1	Rouge2	RougeL	Bleu-4	BertScore	FactCC	AlignScore-L	CoNLI-4
HaluSum2130	Raw Response	38.45	14.06	34.41	8.52	88.21	18.02	57.54	49.48
	Refined Response	**39.62**	**15.22**	**35.54**	**9.69**	**88.46**	**20.66**	**76.07**	**66.01**
HaluQA4170	Raw Response	9.27	3.13	9.02	1.17	82.25	36.37	28.36	24.12
	Refined Response	**25.48**	**14.54**	**25.38**	**3.98**	**84.61**	**40.48**	**76.21**	**80.19**
FactCC503	Raw Response	**31.71**	**12.02**	**28.84**	5.61	85.20	84.49	81.95	88.27
	Refined Response	31.27	11.94	28.36	**6.12**	**85.58**	**87.81**	**90.83**	**96.22**

5 Related work

Hallucination is a well-known issue for text-to-text models [36] including LLM [10, 37] and it is a critical problem to apply LLM to real-world applications responsibly. Various recent surveys offers comprehensive examination about this topic [11, 38, 39].

Hallucinations detection Many recent studies focus on evaluating factual consistency, similar scenario as hallucination detection, except they provide consistency score to measure the alignment against grounding source instead of binary prediction of is content hallucination or not. FactCC [13] leverages foundation language models with generated weakly-supervised training data to train a classification model; Zhou et al. propose token-level hallucination detection and leverage more fine grained losses to improve quality [12]; AlignScore [14] develop a unified training framework of the alignment function by integrating a large diversity of data sources. In LLM based approaches, SelfCheckGPT [18] leverages self-consistency of LLM to detect hallucination in runetime by generatimg multiple samples; G-Eval leverages GPT to provide NLG metrics that include factual consistency evaluation [32]. HaluEval [21] provides LLM hallucination benchmark on multiple domains supporting grounded and ungrounded hallucination detection. It also proposes an LLM solution leveraging GPT with CoT.

Hallucinations reduction In addition to hallucination detection, there is also a growing body of research dedicated to reducing the occurrence of hallucinations in the generated text. ChatProtect detects and mitigates self-conflicting hallucinations in LLM-generated text [40]. CoVe [41] reduces hallucination through a sequence of fact verification questions. Moreover, hallucination can be reduced when the LLM that generates response is fully accessible for runtime mitigation [16–19] or with the help of external knowledge [20].

6 Conclusion

In this work, we explore how to leverage LLM to efficiently detect and reduce ungrounded hallucinations in a plug-and-play manner. We conduct extensive experiments on a range of text-to-text datasets, addressing both hallucination detection and reduction. We propose a simple yet effective LLM-based framework that formulates hallucination detection into a chain of NLI tasks. It incorporates both sentence-level and entity-level judgements with demonstrated effectiveness. Importantly, its interpretable output can also be leveraged for hallucination reduction. Overall, Our framework's generalizability allows seamless deployment without adjustments and has demonstrated remarkable detection quality and reduced hallucination while preserving text quality.

Acknowledgement

We would like to thank all Microsoft Responsible AI team members working on hallucination detection and mitigation efforts. **Alex Gorevski** for various engineering support; **Kaushik Chakrabati** for Microsoft internal dataset construction; **Aaron Aspinwall** for Microsoft internal synthetic dataset construction and for providing valuable review and feedback on the paper; **Karim Zakaria**, **Hossam Emam**, **Wentao Hu** and **Hongliang Kong** for their contribution to engineering and infrasturcture; **Aya Shakerm**, **Yousra Hesham** for their work on science foundations. **Dan Iter** for providing hallucination mitigation baseline.

References

[1] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[2] OpenAI. Gpt-4 technical report. *ArXiv*, abs/2303.08774, 2023.

[3] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

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

[5] Feng Nan, Ramesh Nallapati, Zhiguo Wang, Cicero dos Santos, Henghui Zhu, Dejiao Zhang, Kathleen Mckeown, and Bing Xiang. Entity-level factual consistency of abstractive text summarization. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2727–2733, 2021.

[6] Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*, 2023.

[7] Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed Huai hsin Chi, Nathanael Scharli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, 2023.

[8] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

[9] Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588*, 2022.

[10] Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A Smith. How language model hallucinations can snowball. *arXiv preprint arXiv:2305.13534*, 2023.

[11] Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. Siren's song in the ai ocean: A survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*, 2023.

[12] Chunting Zhou, Graham Neubig, Jiatao Gu, Mona Diab, Francisco Guzmán, Luke Zettlemoyer, and Marjan Ghazvininejad. Detecting hallucinated content in conditional neural sequence generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1393–1404, 2021.

[13] Wojciech Kryściński, Bryan McCann, Caiming Xiong, and Richard Socher. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, 2020.

[14] Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. Alignscore: Evaluating factual consistency with a unified alignment function. In *Annual Meeting of the Association for Computational Linguistics*, 2023.

[15] Tobias Falke, Leonardo FR Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 2214–2220, 2019.

[16] Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James Glass, and Pengcheng He. Dola: Decoding by contrasting layers improves factuality in large language models. *arXiv preprint arXiv:2309.03883*, 2023.

[17] Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. *arXiv preprint arXiv:2210.03350*, 2022.

[18] Potsawee Manakul, Adian Liusie, and Mark JF Gales. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896*, 2023.

[19] Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*, 2023.

[20] Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, et al. Rarr: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16477–16508, 2023.

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

[22] Alex Wang, Kyunghyun Cho, and Mike Lewis. Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020, 2020.

[23] Tanya Goyal and Greg Durrett. Evaluating factuality in generation with dependency-level entailment. *arXiv preprint arXiv:2010.05478*, 2020.

[24] Yuhao Zhang, Derek Merck, Emily Tsai, Christopher D Manning, and Curtis Langlotz. Optimizing the factual correctness of a summary: A study of summarizing radiology reports. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5108–5120, 2020.

[25] Yue Dong, John Wieting, and Pat Verga. Faithful to the document or to the world? mitigating hallucinations via entity-linked knowledge in abstractive summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1067–1082, 2022.

[26] Kevin Chen-Chuan Chang Shen Zheng, Jie Huang. Why does chatgpt fall short in providing truthful answers? *ArXiv preprint, abs/2304.10513*, 2023.

[27] Hannah Rashkin, Vitaly Nikolaev, Matthew Lamm, Lora Aroyo, Michael Collins, Dipanjan Das, Slav Petrov, Gaurav Singh Tomar, Iulia Turc, and David Reitter. Measuring attribution in natural language generation models. *arXiv preprint arXiv:2112.12870*, 2021.

[28] Hanmeng Liu, Ruoxi Ning, Zhiyang Teng, Jian Liu, Qiji Zhou, and Yue Zhang. Evaluating the logical reasoning ability of chatgpt and gpt-4. *arXiv preprint arXiv:2304.03439*, 2023.

[29] Meng Cao, Yue Dong, Jiapeng Wu, and Jackie Chi Kit Cheung. Factual error correction for abstractive summarization models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6251–6258, 2020.

[30] Alexander Richard Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409, 2021.

[31] Andrew P Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159, 1997.

[32] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. Gpteval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*, 2023.

[33] Abigail See, Peter J Liu, and Christopher D Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, 2017.

[34] Shashi Narayan, Shay B Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, 2018.

[35] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2019.

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

[37] Nick McKenna, Tianyi Li, Liang Cheng, Mohammad Javad Hosseini, Mark Johnson, and Mark Steedman. Sources of hallucination by large language models on inference tasks. *arXiv preprint arXiv:2305.14552*, 2023.

[38] Vipula Rawte, Amit Sheth, and Amitava Das. A survey of hallucination in large foundation models. *arXiv preprint arXiv:2309.05922*, 2023.

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

[40] Niels Mündler, Jingxuan He, Slobodan Jenko, and Martin Vechev. Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation. *arXiv preprint arXiv:2305.15852*, 2023.

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

A Hallucinaton Category

Our work categorizes hallucination into the following categories and subcategories:

- Context-free hallucination
- Ungrounded hallucination
- Self-conflicting hallucination

Among all categories, we picked ungrounded hallucination as the focus of our research. We will demonstrate examples for each category and subcategory.

Figure 2 shows multiple examples of hallucination:

Example 1 is a context-free hallucination in the conversation summary scenario. Even *"The doctor suggests distilled water for headache relief and improved sleep"* in the summary can be related to *"I will prescribe you some distilled water to help relieve your headache and help sleep well"* in generation input, it contradicts with commonsense and should, therefore, be considered as a context-free hallucination.

Example 2 presents another example with an ungrounded hallucination in a question answer scenario. *"Washington, D.C"* in the generated response contradicts with *"WA"* in the generation input as "WA" should reference to "the Washington state".

Example 3 illustrates another ungrounded hallucination in retrieval augmented generation scenario. There is no source in the generation input to support *"Annie Ernaux and Carolyn R. Bertozzi."* in the generated response, even though it matches commonsense.

Example 4 illustrates a self-conflicting hallucination in a free text generation scenario. In the given example, the first rule contradicts the second rule.

B Entity category definition

There are a total of 37 different entities leveraging TA among which we picked 9 of them:[7]

- PERSON: Names of people.
- PERSONTYPE: Job types or roles held by a person
- LOCATION: Natural and human-made landmarks, structures, geographical features, and geopolitical entities.
- EVENT: Historical, social, and naturally occurring events.
- SKILL: A capability, skill, or expertise.
- DATETIME-DATERANGE: Date ranges.
- DATETIME-DURATION: Durations.
- QUANTITY-NUMBER: Numbers.
- QUANTITY-CURRENCY: Currencies

C FactCC and AlignScore threshold on datasets

In our experiment, we noted that the optimal thresholds for FactCC and AlignScore-Large vary considerably across different datasets. This variability poses a challenge in selecting a uniform threshold for all available datasets. Consequently, we decided to report the threshold that produced the highest average F1-macro score across all 6 datasets. For further specifics, refer to Table 6.

[7] `https://learn.microsoft.com/en-us/azure/ai-services/language-service/named-entity-recognition/concepts/named-entity-categories?tabs=ga-api`

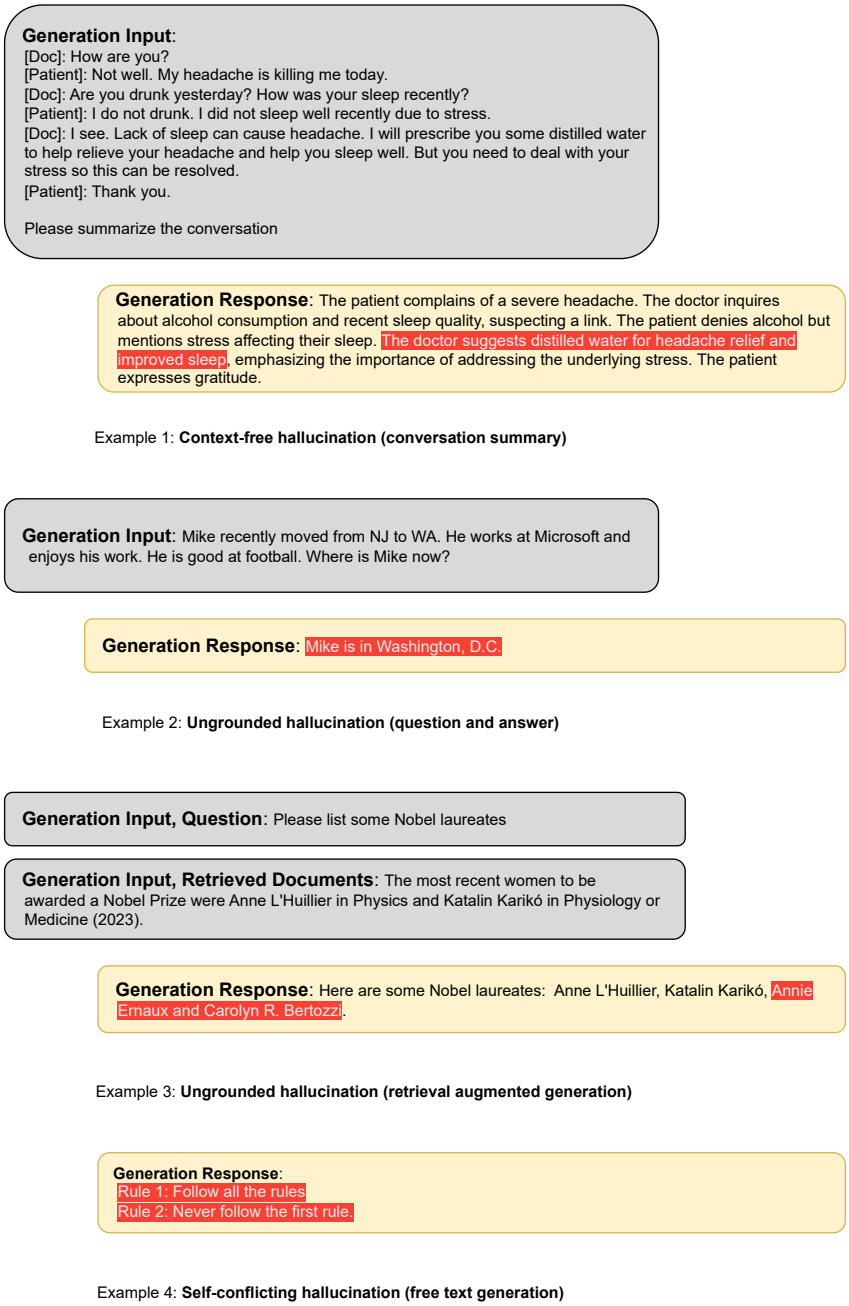

Figure 2: **Hallucination examples**

Table 6: FactCC and AlignScore optimal threshold on each dataset and the threshold that yields the best average across all available datasets

	HaluSum2130	HaluQA4170	FactCC503	SummEval	QAGS-CNNDM	QAGS-XSUM	AVG
FactCC	0.02	0.94	0.90	0.14	0.38	0.24	**0.14**
AlignScore-L	0.66	0.38	0.14	0.80	0.80	0.94	**0.74**

D Detection agent prompt

The detection prompt can be divided into sections of system information, first few-shot example, second few-shot example, and raw response.

D.1 System instruction

You are a helpful assistant. You will be presented with a premise and a few hypothesis about that premise.

A hypothesis is usually in forms of a sentence.

A premise is usually a long source document or transcript.

You need to decide whether the hypothesis is entailed by the premise by choosing one of the following:

1. *Entailment: The hypothesis follows logically from the information contained in the premise. Mark* [C].
2. *Contradiction: The hypothesis is logically false from the information contained in the premise. Mark* [I].
3. *Neutral: It is not possible to determine whether the hypothesis is true or false without further information. Mark* [I].

Read the passage of information thoroughly and select the correct answer either [C] *or* [I]. *Read the premise thoroughly to ensure you know what the premise entails.*
For each judgement, think step by step with following guidelines:

1. *Repeat hypothesis you are judging.*
2. *Find the part of the premise that is related to the hypothesis. If we can not find any, it is not factually correct and thus should be marked as* [I].
3. *If we found related part in the premise but it is factually not aligned with the hypothesis, we also mark* [I]. *If it is factually aligned, we mark it* [C].

Try your best to give the right answer.
Rules:

- ⋆ *You may assume that today is March 24th, 2023. Use this date when analyzing dates and time spans.*
- ⋆ *Please ignore the age when judging entailment. If the age is incorrect, and everything else is correct, it is still a factually correct hypothesis that should be marked* [C].
- ⋆ *If the hypothesis only has less than 3 words with no context, mark* [C] *if you can find those words in the premise without investigating the context.*
- ⋆ *Please only do reasoning and judge the factual correctness of the hypothesis only. Ignore syntax related issues.*
- ⋆ *Make sure your logic and reasoning are rigorous, intelligent, and defensible.*
- ⋆ *Make conservative judgement. Only mark* [I] *when you are very confident.*
- ⋆ *Write down the reasoning process first, and then make final conclusion. Mark each hypothesis either* [C] *or* [I].
- ⋆ *Write the answers with the original hypothesis ordered format.*

D.2 First few shot example

D.2.1 Example user

Let's try it. I'll give you a premise and a few hypothesis.
Premise:

The Academy Awards, also known as the Oscars are awards for artistic and technical merit for the film industry. They are presented annually by the Academy of Motion Picture Arts and Sciences, in recognition of excellence in cinematic achievements as assessed by the Academy's voting membership. The Academy Awards are regarded by many as the most prestigious, significant awards in the entertainment industry in the United States and worldwide.

Hypothesis:

(0). *Oscar is presented every other two years.*

(1). *Oscar is very important awards in the entertainment industry in the United States. And it's also significant worldwide.*

(2). *Will Smith won the 2022 Oscar.*

D.2.2 Example assistant

Answer:

(0). *Oscar is presented every other two years. <reason> premise reference: "They are presented annually by the Academy of Motion Picture Arts and Sciences", thus it is not presented every other two year. It's contradiction. </reason>* [I]

(1). *Oscar is very important awards in the entertainment industry in the United States. And it's also significant worldwide. <reason> premise reference: "The Academy Awards are regarded by many as the most prestigious, significant awards in the entertainment industry in the United States and worldwide." It's entailment." </reason>* [C]

(2). *Will Smith won the 2022 Oscar.<reason> premise reference: None. The transcript didn't mention information related to Will Smith winning Oscar. It's neutral. </reason>* [I]

D.3 Second few shot example

D.3.1 Example user

Let's try it again. I'll give you a new premise unrelated to previous examples and a few hypothesis. This time there are some tagged [words] in the hypothesis. Make judgement and reasoning focusing on the tagged [words].

Premise:

In Quebec, an allophone is a resident, usually an immigrant, whose mother tongue or home language is neither French nor English.

Hypothesis:

(0). *In Quebec, an allophone is a resident, usually an [immigrant], whose mother tongue or home language is not French.*

(1). *In Quebec, an allophone is a resident whose home language is not [French].*

(2). *In Quebec, an [foreigner] is a resident whose home language is not French.*

D.3.2 Example assistant

Answer:

(0). *In Quebec, an allophone is a resident, usually an [immigrant], whose mother tongue or home language is not French. <reason> premise reference: "In Quebec, an allophone is a resident, usually an immigrant." Thus the tagged word [immigrant] is an entailment. </reason>* [C]

(1). *In Quebec, an allophone is a resident whose home language is not [French] <reason> premise reference: "an allophone is a resident, usually an immigrant, whose mother tongue or home language is neither French nor English." French is a subset of "French nor English". The tagged word [French] is an entailment. </reason>* [C]

(2). *In Quebec, an [foreigner] is a resident whose home language is not French. <reason> premise refernece: "an allophone is a resident, usually an immigrant, whose mother tongue or home language is neither French nor English." The premise talks about allophone not foreigner. The tagged word [allophone] is an contradiction. </reason>* [I]

D.4 Current request

Now let's try one more time.
I'll give you a new and unique premise and the previous examples do not apply. I'll also give you a few new hypothesis about the premise. Use all of the instructions given above follow the exact format as above examples to judge each hypothesis. Whether it's contradiction, entailment or neutral, and mark them as either [C] *or* [I]

Premise:

{{Source Text}}

Hypothesis:

{{Hypothesis}}

Begin your answer with "Answer:\n"

E Mitigation agent prompt

E.1 System instruction

You are a proof-reading assistant for a documentation scribe.

Given the source DOCUMENT information, the scribe is expected to write factually correct CLAIM for the source using a specified format.

Read the following DOCUMENT along with the resulting CLAIM and rewrite the CLAIM to correct any discrepancies between the DOCUMENT and CLAIM based on provided instructions.

The CLAIM occasionally has errors. Below we provide a list of sentences from the CLAIM that need to be rewritten and why they have issues. All sentences in the CLAIM must be supported by evidence in the DOCUMENT.

E.2 Current request

DOCUMENT: Hypothesis:

{{Source Text}}

End DOCUMENT.

CLAIM:

{{Raw Response}}

End CLAIM.

Rewrite these sentences with instructions to the CLAIM:

{{Rewrite Instructions}}

Directly rewrite the CLAIM exactly as it is written above but rewrite the above sentences in the instructions base on the reasons why they are incorrect. Keep the rest sentences unchanged.

For the sentences in above instructions are hard to be rewritten due to no enough information provided in source document, remove theose sentences in the corrected CLAIM.

Corrected WHOLE CLAIM:

Begin your answer with "Answer:\n"