# HalluMeasure: Fine-grained Hallucination Measurement Using Chain-of-Thought Reasoning

**Shayan A. Akbar***, **Md Mosharaf Hossain***, **Tess Wood***, **Si-Chi Chin, Erica Salinas, Victor Alvarez, Erwin Cornejo**
Customer Experience and Business Trends, Amazon.com
{shayaakb, hosmdmos, tesswoo, sichi, erislin, miranvi, eccornej}@amazon.com

## Abstract

Automating the measurement of hallucinations in LLM-generated responses is a challenging task as it requires careful investigation of each factual claim in a response. In this paper, we introduce HalluMeasure, a new LLM-based hallucination detection mechanism that decomposes an LLM response into atomic claims, and evaluates each atomic claim against the provided reference context. The model uses a step-by-step Chain-of-Thought reasoning process and can identify 3 major categories of hallucinations (e.g., contradiction) as well as 10 more specific subtypes (e.g., overgeneralization) which help to identify reasons behind the hallucination errors. Specifically, we explore four different configurations for HalluMeasure's classifier: with and without CoT prompting, and using a single classifier call to classify all claims versus separate calls for each claim. The best-performing configuration (with CoT and separate calls for each claim) demonstrates significant improvements in detecting hallucinations, achieving a 10-point increase in F1 score on our TechNewsSumm dataset, and a 3-point increase in AUC ROC on the SummEval dataset, compared to three baseline models (RefChecker, AlignScore, and Vectara HHEM). We further show reasonable accuracy on detecting 10 novel error subtypes of hallucinations (where even humans struggle in classification) derived from linguistic analysis of the errors made by the LLMs.

## 1 Introduction

Hallucinations in Large Language Models (LLMs) are inevitable (Xu et al., 2024) and can cause significant harm. For instance, non-existent legal cases generated by an LLM in court papers submitted by a law firm resulted in sanctions by a judge (CNBC, 2023). Wrongly claiming the Webb Space Telescope was the first to photograph an exoplanet during an LLM-based product demo led to a 7.7% drop in a company's stock value, wiping out $100 billion in market capitalization (Reuters, 2023). Moreover, misleading information about bereavement travel policy generated by an airline chatbot led to a court order for partial refund to a passenger (Technica, 2024). Consequently, detecting and measuring hallucinations in LLMs has become a crucial research area in recent years to prevent their potential harmful effects.

We measure hallucinations in LLM responses by comparing the claims made in a response against a reference context document. Since manually annotating LLM responses against such context documents is time-consuming and expensive, there is a need to develop automatic approaches to measure hallucination at scale. In recent years, several studies (Zha et al., 2023; Hu et al., 2023; Vectara, 2023; Min et al., 2023; Huang et al., 2023) have proposed solutions to automate measurement of hallucinations in LLM responses. These solutions involve classifying pairs of context and LLM response texts into hallucination vs. non-hallucination classes using machine learning models.

We highlight key differences between prior solutions and our HalluMeasure method below:

- **LLM-based classification approach:** Unlike some of the prior works that use traditional machine learning models and BERT-based classifiers (Zha et al., 2023; Vectara, 2023), we use LLMs with prompt engineering for hallucination detection and measurement, achieving better performance.
- **Claim-level classification:** Many studies (Zha et al., 2023; Vectara, 2023) propose solutions measuring hallucinations at the response level, some (Kryscinski et al., 2020; Laban et al., 2021) segment responses into sentences, and a few (Min et al., 2023; Chern et al., 2023) operate at the claim level. We employ claim-level classification, enabling fine-grained mea-

*These authors contributed equally to this work.

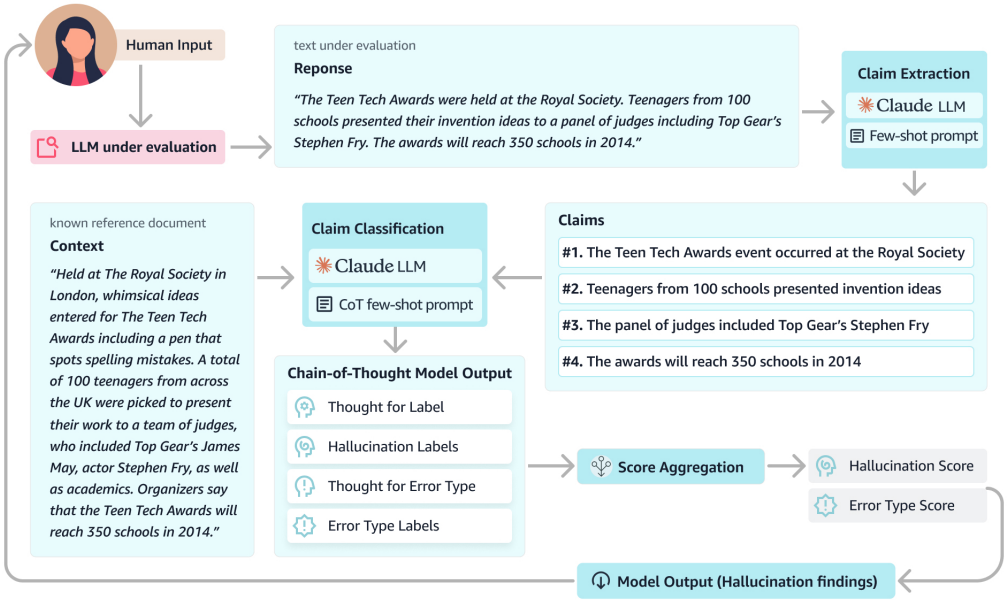


Figure 1: HalluMeasure Overview. The LLM response goes through claim extraction model to produce a list of claims which are classified using a claim classification model. Finally scores are aggregated and returned to the user.

**Chain-of-Thought Model Output**

| Claim | Thought for Label | Labels (5) | Thought for Error Type | Error Type (10) |
|---|---|---|---|---|
| 1 | Context says that whimsical ideas were presented at the The Teen Tech Awards and that the event occurred at The Royal Society in London. | Supported | None, claim is supported. | None |
| 2 | Context explicitly mentions that there were 100 students from across the UK, but not how many schools the students came from | Absent | The claim involves a reasoning error by attributing the 100 teenagers to 100 different schools. This information is not present or supported by the context. | Reasoning |
| 3 | Context mentions a judge from Top Gear named James May, and also separately mentions actor Stephen Fry | Contradicted | The claim contains an error due to combining information about two different people. Stephen Fry is described as an actor, and James May is on Top Gear. | False Concat |
| 4 | Context explicitly states that the organizers say the awards will reach 350 schools in 2014. | Partially Supported | The claim is mostly accurate, but does not correctly attribute the information. The quote from the organizers is stated as fact, without attribution | Attribution |

Figure 2: HalluMeasure assigns each claim a thought for hallucination labeling, then a label from 5 labels (supported, partially supported, absent, contradicted, unsupported, unevaluatable). Each claim also gets a thought for error type classification, then an error type label from 10 labels (see Table 2).

surement. When hallucinated information exists within lengthy LLM responses, evaluating individual extracted claims improves hallucination detection accuracy.

- **Chain-of-Thought Prompting:** We leverage few-shot Chain-of-Thought (CoT) prompting (Wei et al., 2022) to teach reasoning abilities to our claim classifier model, enabling it to accurately classify claims through exemplar demonstrations. This enhances claim classification accuracy over prior works that use simple few-shot prompting (Hu et al., 2023).

- **Single classification call for list of claims:** Previous studies processed claims individually (Hu et al., 2023). However, we leverage batch prompting (Cheng et al., 2023) and demonstrate an LLM-based classifier that classifies multiple claims extracted from the same response simultaneously with reasonable accuracy. This approach reduces LLM calls, associated latency, and costs, enabling more scalable hallucination detection while maintaining reasonable performance.

- **Fine-grained error types:** Previous works often classify results into binary or NLI (ternary) classes. Our study demonstrates the value of granular hallucination error types (Section 3.2). By providing deeper insights into the type of hallucinations produced, HalluMeasure enables more targeted solutions to enhance LLM reliability.

| LLM-in-test response | Atomic claims |
|---|---|
| Samsung's Gear Blink could have a projected keyboard that allows you to type in the air. Ralph Lauren's Polo Tech Shirt uses bio-sensing fabrics to monitor physical activity. Hush earplugs filter out unwelcome sounds while allowing phone calls and alarms to intrude. | 1. Samsung has a product called Gear Blink.<br>2. Gear Blink could have a projected keyboard.<br>3. Gear Blink's projected keyboard would allow typing in air.<br>4. Ralph Lauren has a product called Polo Tech Shirt.<br>5. Polo Tech Shirt uses bio-sensing fabrics.<br>6. Polo Tech Shirt bio-sensing fabrics monitor physical activity.<br>7. There is a product called Hush earplugs.<br>8. Hush earplugs filter out unwelcome sounds.<br>9. Hush earplugs allow phone calls to be heard.<br>10. Hush earplugs allow alarms to be heard. |

Table 1: An example summary text of a news article & extracted atomic claims by our claim extractor.

Our HalluMeasure method works by first decomposing the LLM response into a set of claims using a claim extraction model. Then, we classify the claims into 5 key classes (e.g., supported, absent, contradiction, partially supported, and unevaluatable) by comparing them against the contexts using our claim classification model. Additionally, we classify the claims into 10 novel distinct error types (e.g., entity, temporal, over-generalization, etc.) that provide a fine-grained analysis of hallucination errors. Finally, we produce an aggregated hallucination score by measuring the rate of unsupported claims (i.e., those assigned classes other than supported), and calculate the distribution of fine-grained error types. This distribution provides LLM builders with valuable insights into the nature of errors their LLM is making, facilitating targeted improvements. Figure 1 illustrates the main components and process behind HalluMeasure.

Our results demonstrate that HalluMeasure outperforms existing solutions in terms of F1 score and AUC ROC metric on two datasets: TechNews-Summ (our own curated dataset) and a popular public benchmark dataset SummEval (Fabbri et al., 2021). We attribute HalluMeasure's superior performance to (1) our improved prompting strategy that utilizes Chain-of-Thought (CoT) reasoning, and (2) our claim-level classification approach that measures hallucination based on fine-grained information present in the response text.

We attempt to answer the following 6 key research questions as part of this study:

- **RQ1:** How effectively can HalluMeasure extract claims from LLM responses?
- **RQ2:** How does HalluMeasure method compare against state-of-the-art methods?
- **RQ3:** Is a single call to classify all claims effective for hallucination classification?
- **RQ4:** Can HalluMeasure effectively detect fine-grained hallucination error types?
- **RQ5:** Does the use of CoT prompting improve hallucination measurement performance?
- **RQ6:** How effectively can HalluMeasure's method generalize to different underlying LLMs for hallucination classification?

Our key contributions are (1) a novel HalluMeasure method that automatically measures hallucinations using fine-grained analysis of LLM responses using Chain-of-Thought reasoning, (2) experimental results of our HalluMeasure method that outperforms existing solutions (RefChecker, AlignScore, and Vectara HHEM), and (3) a novel TechNews-Summ dataset containing fine-grained claim level labels for news summarization task with tech news articles taken from CNN/DailyMail dataset.

## 2 Related Work

Hallucination is a topic of growing research interest, and a range of prior studies have addressed its identification and measurement. Several survey papers provide a useful overview and analysis (Huang et al., 2023; Wang et al., 2024; Rawte et al., 2023). A number of works provide either *a hallucination measurement dataset* (Li et al., 2023; Lin et al., 2022; Tam et al., 2023), *an automatic evaluation metric* (Zha et al., 2023; Chern et al., 2023; Min et al., 2023; Manakul et al., 2023; Mündler et al., 2024; Gekhman et al., 2023; Kryściński et al., 2019) or *a meta-evaluation* (i.e., evaluation of different hallucination metric performances) (Honovich et al., 2022; Gabriel et al., 2021). Many of the use cases addressed focus on summarization (e.g., news summarization or headline generation) and use popular news datasets for testing (CNN/DailyMail news articles corpus (See

et al., 2017), or XSUM news headline dataset (Narayan et al., 2018)). Another popular use case is Wikipedia-style biography generation (e.g., as in WikiBio dataset (Lebret et al., 2016)). Previous approaches to measurement include using pretrained or finetuned models or NLI- and Question-Answer-Generation (QAG)-based metrics. More recent studies employ LLMs to classify responses. Most of these (Zha et al., 2023; Vectara, 2023) classify at the response level while a smaller number classify at the sentence (Kryscinski et al., 2020; Laban et al., 2021) or fine-grained claim level (Min et al., 2023; Chern et al., 2023). Further, they differ in whether they use a binary hallucination/non-hallucination classification, ternary NLI classes (Chern et al., 2023; Min et al., 2023), or perform fine-grained multi-class classification to divide hallucination into different error types (e.g., negation error, number swap, or entity swap, etc.) (Rawte et al., 2023). A number of different taxonomies of hallucination error types have been proposed (Tang et al., 2023; Zhu et al., 2023; Tang et al., 2024). We build on these earlier approaches by combining claim-level analysis with fine-grained error types (distinguished based on their impact and potential causes or mitigations), and by using current LLMs with few-shot learning and CoT prompting to produce strong results in measuring hallucinations at both the claim and response levels.

## 3 Method

In this section we present methodology for HalluMeasure. We use a claim-level hallucination measurement approach inspired by Chern et al. (2023), Hu et al. (2023), and Min et al. (2023). We decompose LLM responses into smaller units ('claims') for more precise measurement using our claim extraction model based on Claude 2.1. Then, we classify claims and assign high-level labels, further identifying 10 more specific types of hallucination errors for unsupported claims using CoT reasoning with Claude 3 Sonnet (See Table 9).

HalluMeasure calculates the percentage of hallucinations at both the claim and response level, and provides the distribution of hallucination types identified. Below, we discuss specific components of our hallucination measurement method that takes in context and response pairs as input and produces hallucination scores as output. Figure 1 shows the HalluMeasure process.

### 3.1 Extracting Claims from LLM Responses

As noted above, our approach decomposes an LLM response into a set of claims. An intuitive definition of claim is 'the smallest unit of information that can be evaluated against a context'; in the prototypical case, this consists of a single predicate with a subject and (optionally) an object. Several recent works on hallucination and factuality decompose sentences into claims for evaluation; however, as noted by Wanner et al. (2024), the method of decomposition affects the number of claims extracted from a given model response, and therefore impacts hallucination metrics. In general, higher atomicity of claims allows for more precise measurement and localization of hallucinations. Table 1 shows an example response with the claims extracted by our claim extractor.

We develop a claim extraction model which, given an LLM response, decomposes that response into claims to be evaluated. We prompt the claim extraction model using a small set of demo example responses with manually extracted claims, which have been judged to be both atomic and comprehensive (i.e., the claims list covers all significant information from the response text). Note that unlike some existing approaches (Min et al., 2023), we don't use sentences in responses to decompose into claims. Rather, we directly extract the list of claims from the full response text since a single claim may incorporate information from more than one sentence (e.g. entity resolution, reasoning).

We utilize the Amazon Bedrock hosted Claude 2.1 model (with *temperature=0.8* and *top_p=0.9*) (Amazon Web Services) to develop our claim extractor. The process involves developing a prompt that enables the model to learn how to extract claims from an LLM response. The prompt structure begins with an initial instruction, followed by a set of rules outlining the task requirements. It also includes a selection of example texts accompanied by their manually extracted claims. Finally, the prompt ends with the target response (i.e., LLM response under evaluation) from which the model needs to extract the relevant claims. By providing this comprehensive prompt, we aim to effectively teach (without updating weights) Claude to accurately extract claims from any given response. Once Claude returns the claims as a text string, we convert it into a Python list to store as a list of claims associated with the response. Figure 3 in the appendix provides the prompt for claim extraction.

| | |
|---|---|
| Number | A claim has a different number than the original context (e.g. 20% vs. 0.7%). Any number, including year, dimensions, ages, etc. |
| Entity | A claim includes swapped, incorrectly specified, or inserted noun phrases (e.g. one named entity used in a context where another word is expected). |
| False Concatenation | A claim incorrectly combines information about multiple entities or events. |
| Attribution Failure | A claim lacks proper attribution, either crediting the wrong source or presenting information as fact without citation. |
| Overgeneralization | A claim is based on accurate contextual information but is too broad or too general to be supported by the context. |
| Reasoning Error | A claim is based on accurate contextual information but contains a reasoning error or makes an unsupported conclusion. |
| Hyperbole | A claim is based on accurate information but exaggerated or overstated. |
| Temporal | A claim does not accurately incorporate tense, modality (e.g. might vs. will), or time reference in relation to the context. |
| Context-based meaning error | A claim includes incorrect interpretation of idiomatic language, homonyms, or words with multiple meanings, therefore failing to capture the intended meaning. |
| Other | All other types of errors are captured in this category. This includes too-far inferences, circumstantial errors, or incoherent sentences or paragraphs. |

Table 2: Specific subtypes of hallucinations.

### 3.2 Classifying Claims into Hallucination Labels

When comparing claims against a reference context, our primary distinction is between claims that are *supported* vs. *unsupported* by the context. A claim is *supported* if, under normal circumstances, a reader would believe the claim to be true given the context. We divide *unsupported* claims into three main types: **1. Contradicted claims**: the context contains information which is explicitly inconsistent with the claim ('intrinsic hallucination'). **2. Absent claims**: there is no evidence in the context to support or refute the claim ('extrinsic hallucination'), and **3. Partially Supported claims**: the claim is almost fully supported by the context but has a minor error. We note that the categories *Supported, Contradicted,* and *Absent* are similar to *Entailed, Contradicted,* and *Neutral* in NLI terminology. However, the NLI labels may imply a continuum from entailment to contradiction, in which *neutral* appears to be a lesser error - yet many of the most stereotypical and most problematic examples of hallucination are those in which a model adds completely new, unsupported content in its output. At the same time, we recognize that there are in fact degrees of severity of errors, and we therefore distinguish a class of Partially Supported claims. Distinguishing these claims which are unsupported in subtle ways allows us to both quantify the presence of hallucination errors, and, at least at a basic level, distinguish their severity. Identifying Partially Supported claims is unique to our approach. Examples include: missing/incorrect attribution, number/conjunction misinterpretation, and mild hyperbole. For example, if the context states, "*According to the company, their revenue increased*" then the claim "*The company's revenue increased*" would be Partially Supported. In addition to the above claim types, we have an Unevaluatable class label for claims that do not fall into any of the high-level types (e.g., questions).

Within unsupported claims, we identify specific subtypes of hallucinations in order to understand and compare hallucinations from specific models in more detail, and to potentially apply such information to strategies for reducing or mitigating LLM hallucinations. We currently distinguish 10 subtypes (see Table 2), though we will continue to refine our categorization based on emerging data as part of our future work. Subtypes of hallucination are exemplified in Table 9 in the appendix.

Unlike the traditional BERT-based approaches in some recent studies (Zha et al., 2023; Vectara, 2023), we leverage in-context prompting and develop the claim classifier with the Amazon Bedrock-hosted Claude 3 Sonnet model (with *temperature* $\approx 0.1$ for reproducibility). Notably, our classifier not only detects the main hallucination labels but also identifies specific subtypes (as elaborated in Table 9), and provides an explanation for the claim label. To analyze the effectiveness of our approach, we have developed four prompting

| Prompting LLM with CoT | Prompting LLM without CoT |
|---|---|
| **Step 1**: Read and fully understand the claim. It is a short, standalone sentence containing a single piece of information related to the source text. | **Step 1**: Read and fully understand the claim. It is a short, standalone sentence containing a single piece of information related to the source text. |
| **Step 2**: Thoroughly analyze how the claim relates to the information in the source text. Then, write your reasoning in 1-3 sentences to determine the most appropriate label to describe the claim's truthfulness based on the source text. | **Step 2**: Write the most appropriate label for the claim based on the source text. |
| **Step 3**: Write the label for the claim based on your reasoning in Step 2. | **Step 3**: If the label in Step 2 is 'contradicted', 'absent', or 'partially supported', then write the specific error type (i.e., sublabel) for the claim. |
| **Step 4**: If the label in Step 3 is 'contradicted', 'absent', or 'partially supported', then thoroughly analyze the specific errors (i.e., sublabels) present in the claim based on provided source text. Then, provide your reasoning in 1-3 sentences to determine the error. However, if the label in Step 3 is 'supported', simply write 'None - claim is supported', and set the sublabel to 'None'. | |
| **Step 5**: Write sublabel based on your reasoning in Step 4. | |

Table 3: Instruction steps for prompting LLM with and without CoT.

strategies to investigate two key aspects: 1) the potential benefits of incorporating Chain-of-Thought (CoT) reasoning (Wei et al., 2022), and 2) whether evaluating claims individually (requiring multiple Claude API calls) offers advantages over evaluating all claims together (requiring a single API call). We provide the four prompt setups below.

**(1) With and (2) Without CoT Reasoning.** We check if asking the model to think and analyze before deciding the labels and specific error subtypes (i.e., sublabels) is beneficial. By following Wei et al. (2022), we develop a 5 step CoT prompt, including steps to thoroughly examine each claim's faithfulness to the reference context, and writing down the reasoning behind the thoughts. Similar to other reasoning tasks where CoT is useful (Wei et al., 2022) like mathematical and commonsense reasoning, we hypothesize that these written thoughts provide insights into the model's reasoning process prior to selecting the final hallucination label for each claim. Table 3 shows the steps for with and without CoT prompting strategies. See Figures 5 and 4 for prompt templates. Figure 2 shows CoT model output with Thoughts generated for hallucination label and error type sublabels.

**(3) One-claim-eval and (4) All-claims-eval.** We check if evaluating each claim separately is better than evaluating all claims together. The first approach evaluates each claim independently against the context by making multiple API calls to Claude (depending on the number of claims in a response). The second approach includes all the claims in the same prompt and makes a single API call to Claude. We refer to the former as one-claim-eval and the latter as all-claims-eval. While all-claims-eval is better for reducing latency, one-claim-eval performs better as it allows the model to focus on only one claim at the time of evaluation. See Figures 6 and 7 for prompt templates.

### 3.3 Aggregating Scores for Hallucination Measurement

After classifying claims, each claim has a label (out of 5 labels) and an error type (out of 10 types) assigned to it. Now, we assign a score for hallucination and for each error type by aggregating across all claims in the responses in the dataset. We produce two scores: 1) Response hallucination rate, calculated by dividing the number of unsupported claims (combining different error types) by the total number of claims for the response. 2) Error type distribution, calculated by scoring subtype or class errors separately.

## 4 Experiments and Results

### 4.1 Dataset

We present the datasets used to evaluate our HalluMeasure approach and compare with existing models. We create the first dataset, **TechNews-Summ**, by sampling 30 tech news articles from the CNN/Dailymail dataset, collecting summaries (20

| | Response-level | | | Claim-level | | |
|---|---|---|---|---|---|---|
| | Precision | Recall | F1 | Precision | Recall | F1 |
| Vectara HHEM | 0.75 | 0.15 | 0.25 | - | - | - |
| AlignScore | 0.57 | 0.20 | 0.30 | - | - | - |
| RefChecker (AlignScore Checker) | 0.68 | 0.75 | 0.71 | - | - | - |
| RefChecker (NLI Checker) | 0.75 | 0.75 | 0.75 | - | - | - |
| RefChecker (Claude Checker) | 0.79 | 0.75 | 0.77 | - | - | - |
| W/o CoT + all-claims-eval (ours) | 0.73 | 0.80 | 0.76 | 0.72 | 0.59 | 0.65 |
| W/ CoT + all-claims-eval (ours) | 0.85 | **0.85** | 0.85 | 0.73 | **0.67** | 0.70 |
| W/o CoT + one-claim-eval (ours) | 0.83 | 0.75 | 0.79 | 0.86 | 0.56 | 0.68 |
| W/ CoT + one-claim-eval (ours) | **0.89** | **0.85** | **0.87** | **0.87** | 0.66 | **0.75** |

Table 4: Results on TechNewsSumm Dataset: Response- and claim-level evaluation metrics for the *unsupported* label obtained with the existing methods and the four settings of our approach. For a fair comparison, we convert our four main labels into binary labels (i.e., *supported* and *unsupported*).

| | API Call Time | #Prompt Tokens (K) | #Output Tokens (K) |
|---|---|---|---|
| W/o CoT + all-claims-eval | 9.45 | 6.27 | 0.51 |
| W/o CoT + one-claim-eval | 42.24 | 81.51 | 0.52 |
| W/ CoT + all-claims-eval | 40.21 | 10.14 | 1.27 |
| W/ CoT + one-claim-eval | 77.17 | 133.10 | 1.58 |

Table 5: Latency and Token Stats on TechNewsSumm Experiments: Claude API call duration and Input/Output tokens for HalluMeasure's Claim Classifier. Duration in seconds, token counts in thousands (K). For one-claim-eval setups, prompt and output tokens from all API calls for a single response are summed.

from the Cohere Command model and 10 human-written), extracting atomic claims (400 claims) from the summaries using our claim extractor (Section 2), and manually evaluating the claims against the reference context to identify three main types and 10 specific types (i.e., subtypes) of hallucinations (Table 9). We observe moderate agreement between the annotators in main labels (Kappa: 0.44) and subtypes annotations (Kappa: 0.45). Our second dataset is the **SummEval** dataset (Fabbri et al., 2021) with 1600 annotated samples (hallucination vs. non-hallucination) at the response level from the TRUE benchmark (Honovich et al., 2022).

### 4.2 Baselines

We compare HalluMeasure with three state-of-the-art hallucination measurement approaches: **Vectara HHEM** (Vectara, 2023) outputs a factual consistency score (0-1) using a finetuned cross-encoder model. We use 1 - factual consistency score as hallucination score and threshold at 0.5 for label assignment. **AlignScore** (Zha et al., 2023) classifies claims against contexts into aligned/not-aligned classes using a RoBERTa model trained on 7 NLP tasks. We use 1 - aligned class score as hallucination score and threshold at 0.5. **RefChecker** (Hu et al., 2023) splits responses into claim-triplets and checks them against references using a claim checker (GPT4, Claude2, NLI, or AlignScore). It produces hallucination scores based on strict (any contradicted claim means the response is labeled as hallucination), soft (ratio of contradicted and neutral claims), or majority voting criteria. We use the soft criteria for hallucination score and strict for label assignment.

### 4.3 Hallucination Measurement Experiments

We provide experimental results for these four hallucination measurement approaches (HalluMeasure, RefChecker, AlignScore, and Vectara HHEM). Tables 4 and 6 present evaluation results on binary classification and error subtype classification on our own curated TechNewsSumm dataset. Table 7 shows results on the SummEval dataset.

Through our experiments, we attempt to answer our research questions (RQs) below:

***RQ1: How effectively can HalluMeasure extract claims from LLM responses?***

We validate our claim extractor's accuracy by evaluating its performance on 25 tech news articles from the CNN/DailyMail dataset. We generate summaries using Cohere's Command model and extract claims from these summaries. Four researchers manually annotated the claims; each claim was annotated by two annotators, with disagreements adjudicated. The rate of claims with

| | #Adj. | #Pred. | Precision | Recall | F1 |
|---|---|---|---|---|---|
| Number | 1 | 0 | 0.00 | 0.00 | 0.00 |
| Entity | 3 | 9 | 0.11 | 0.50 | 0.18 |
| False Concatenation | 6 | 1 | 0.00 | 0.00 | 0.00 |
| Attribution Failure | 1 | 0 | 0.00 | 0.00 | 0.00 |
| Overgeneralization | 6 | 13 | 0.15 | 0.50 | 0.24 |
| Reasoning Error | 10 | 7 | 0.14 | 0.17 | 0.15 |
| Hyperbole | 3 | 2 | 1.00 | 1.00 | 1.00 |
| Temporal | 3 | 4 | 0.50 | 1.00 | 0.67 |
| Context-based meaning | 6 | 0 | 0.00 | 0.00 | 0.00 |
| Other | 48 | 28 | 0.93 | 0.58 | 0.71 |
| Macro-average | | | 0.32 | 0.47 | 0.33 |
| Weighted-average | | | 0.71 | 0.52 | 0.58 |

Table 6: Error Subtypes Classification Performance on TechNewsSumm Dataset: Evaluation results on the specific subtypes of hallucinations obtained with our best setup of HalluMeasure model (i.e., W/ CoT + one-claim-eval). #Adj. denotes the count of adjudicated subtypes, and #Pred. denotes the count of predicted subtypes.

disagreements is 6.5% (12/185). The adjusted Cohen's kappa (PABAK) score of 0.87 indicates strong annotator agreement. Using the adjudicated claims as ground truth, the claim extractor's precision is 0.96. Since there is no definitive gold set of claims, we compute a revised recall metric as correctly extracted claims / (correctly extracted claims + missing correct claims). The revised recall and F1-score of 0.97 indicate the claim extractor accurately extracts claims from responses.

***RQ2: How does HalluMeasure method compare against state-of-the-art methods?***

We answer this question by comparing our method with RefChecker, AlignScore, and Vectara HHEM. Out of these three methods, RefChecker's approach is similar to HalluMeasure with a key distinction that we use CoT few-shot prompting instead of simple few-shot prompting employed by RefChecker. The remaining two methods (AlignScore and Vectara HHEM) are specifically trained to measure hallucinations using BERT-based models. While RefChecker performs similarly to some of our prompting setups (e.g. without CoT), HHEM and AlignScore achieve lower performance on our TechNewsSumm dataset. Our best setup (HalluMeasure W/ CoT + one-claim-eval) outperforms existing models by at least 13% F1 score on TechNewsSumm dataset (See Table 4).

To demonstrate performance on a public benchmark dataset, we show experimental results on the popular SummEval dataset in Table 7. Note that we report AUC ROC scores on the SummEval dataset as computed using the TRUE benchmark software package (Honovich et al., 2022). Our HalluMeasure model achieves an AUC ROC value of 0.80, outperforming the baseline models by 3 to 9 points on 1600 samples of SummEval. The baseline models' AUC ROC values are: Vectara HHEM: 0.77, AlignScore: 0.71, RefChecker (Alignscore checker): 0.75, RefChecker (NLI checker): 0.75, and RefChecker (Claude 2): 0.74. Our results on SummEval show that HalluMeasure significantly outperforms several existing baseline models. So, results on two datasets show that HalluMeasure outperforms existing state-of-the-art models.

***RQ3: Is a single call to classify all claims effective for hallucination classification?***

We answer this question by comparing the results for one-claim-eval vs. all-claims-eval from Table 4. Evaluating one claim at a time is beneficial over evaluating all claims together based on results on both response and claim level evaluations (e.g., F1: 0.75 vs. 0.70). However, all-claims-eval comes with benefit of improved latency and cost compared to one-claim-eval (9.45 secs vs. 42.24 secs for W/o CoT prompt; 40.21 secs vs. 77.71 secs for W/ CoT prompt). See Table 5 for details.

***RQ4: Can HalluMeasure effectively detect fine-grained hallucination error types?***

While HalluMeasure's best setup shows excellent overall results, it struggles to accurately classify the specific error types, with macro-F1 and weighted-F1 scores of only 0.33 and 0.58, respectively (Table 6). This is likely due to having 10 different error types, with similar classification issues demonstrated by low human agreement (Kappa: 0.45) as mentioned in Section 4.1. Moreover, some error types are hard to distinguish and may not in fact be mutually exclusive, such as *reasoning error* vs. *context-based meaning*, and *false concatenation* vs. *overgeneralization*. Improving the accurate identification of these specific error types remains

| | AUC ROC |
|---|---|
| Vectara HHEM | 0.77 |
| AlignScore | 0.71 |
| RefChecker (AlignScore Checker) | 0.75 |
| RefChecker (NLI Checker) | 0.75 |
| RefChecker (Claude Checker) | 0.74 |
| HalluMeasure (ours; W/o CoT) | 0.78 |
| HalluMeasure (ours; W/ CoT) | **0.80** |

Table 7: Performance Comparison on SummEval Benchmark: AUC ROC scores obtained with existing models and HalluMeasure (W/ CoT and W/o CoT + one-claim-eval) on the SummEval dataset (Fabbri et al., 2021).

a priority for future work.

***RQ5: Does few-shot CoT prompting improve the hallucination measurement performance?***

We answer this question by comparing the performance of our claim classifier model with and without CoT prompting. Table 4 shows that in both one-claim-eval and all-claims-eval settings, CoT prompting improves model performance on our TechNewsSumm dataset at both response-level (F1: 0.85 vs. 0.76) and claim-level (0.7 vs. 0.65). Table 7 also shows that CoT prompting improves model performance on SummEval public benchmark dataset (AUC ROC: 0.78 vs. 0.80).

***RQ6: How effectively can HalluMeasure's method generalize to different LLMs for hallucination classification?***

We present additional results with Cohere's Command R+ and Mistral Large models on the TechNewsSumm dataset in Table 8. We only experiment with the *one-claim-eval* setup due to read timeout issues with the *all-claims-eval* setup; and we obtain only response-level results due to the additional annotation effort required for claim-level analysis. The results show that, unlike Claude Sonnet 3.0, Command R+ and Mistral Large exhibit similar results for both with and without CoT approaches, with F1 scores ranging from 0.77 to 0.79. However, these scores are significantly lower than the best results we achieved with Claude Sonnet (0.87). We also check HalluMeasure's performance with these LLMs on the SummEval dataset. Surprisingly, Mistral Large performs slightly better than Claude, achieving an AUC ROC score of 0.81, while Command R+ scores 0.73. These results show that performance may be sensitive to characteristics of the dataset, and proprietary LLMs do not always out-perform open source models.

| | Precision | Recall | F1 |
|---|---|---|---|
| **Cohere's Command R+** | | | |
| W/o CoT + one-claim-eval | 0.79 | 0.75 | 0.77 |
| W/ CoT + one-claim-eval | 0.76 | 0.80 | 0.78 |
| **Mistral large** | | | |
| W/o CoT + one-claim-eval | 0.68 | 0.95 | 0.79 |
| W/ CoT + one-claim-eval | 0.69 | 0.90 | 0.78 |

Table 8: Response-level results from Cohere's Command R+ and Mistral Large LLMs on TechNewsSumm.

## 5 Conclusion

We introduce HalluMeasure, a novel approach to automatically measure hallucinations in LLM responses. HalluMeasure decomposes an LLM response into set of claims using a claim extraction model based on Claude with few-shot prompting. It compares the extracted claims against a context document using a claim classification model leveraging few-shot Chain-of-Thought prompting with Claude to enhance classification performance. An aggregated response-level score is produced by measuring the rate of unsupported claims and distribution of specific error types. We demonstrate the effectiveness of HalluMeasure on: TechNewsSumm (our own curated dataset with detailed claim-level error labels) and SummEval (a popular benchmark dataset). Our results demonstrate HalluMeasure's superior performance over baseline models, with at least a 10-point F1 score improvement on TechNewsSumm and a 3-point AUC ROC increase on SummEval. For future work, we plan to employ dynamic few-shot prompting and use optimized prompts with fast-inference LLMs.

## Limitations

Our study detects 10 hallucination error types, including an Other class. We plan to further explore and refine error categorization for better detection and to support mitigation. Additionally, we have focused mainly on hallucination detection in plain text responses, and have yet to explore detecting and measuring hallucinations in other formats like tables and code. While our focus to date has been on news article benchmarks, we aim to include specialized domains like medicine, law, and finance.

## Acknowledgement

The authors would like to thank Lisa Baytler, UX designer, for her assistance in creating figures for this paper.

## References


Amazon Web Services. Amazon bedrock. https://aws.amazon.com/bedrock/?refid=36201f68-a9b0-45cc-849b-8ab260660e1c. Accessed: 05-30-2024.

Zhoujun Cheng, Jungo Kasai, and Tao Yu. 2023. Batch prompting: Efficient inference with large language model apis. *arXiv preprint arXiv:2301.08721*.

I-Chun Chern, Steffi Chern, Shiqi Chen, Weizhe Yuan, Kehua Feng, Chunting Zhou, Junxian He, Graham Neubig, and Pengfei Liu. 2023. Factool: Factuality detection in generative ai – a tool augmented framework for multi-task and multi-domain scenarios. *Preprint*, arXiv:2307.13528.

CNBC. 2023. Judge sanctions lawyers for brief written by a.i. with fake citations.

Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. Summeval: Re-evaluating summarization evaluation. *Preprint*, arXiv:2007.12626.

Saadia Gabriel, Asli Celikyilmaz, Rahul Jha, Yejin Choi, and Jianfeng Gao. 2021. Go figure: A meta evaluation of factuality in summarization. *Preprint*, arXiv:2010.12834.

Zorik Gekhman, Jonathan Herzig, Roee Aharoni, Chen Elkind, and Idan Szpektor. 2023. Trueteacher: Learning factual consistency evaluation with large language models. *Preprint*, arXiv:2305.11171.

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. *Preprint*, arXiv:2204.04991.

Xiangkun Hu, Dongyu Ru, Qipeng Guo, Lin Qiu, and Zheng Zhang. 2023. Refchecker for fine-grained hallucination detection.

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

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. 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, Online. Association for Computational Linguistics.

Wojciech Kryściński, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Evaluating the factual consistency of abstractive text summarization. *Preprint*, arXiv:1910.12840.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2021. Summac: Re-visiting nli-based models for inconsistency detection in summarization. *Preprint*, arXiv:2111.09525.

R. Lebret, D. Grangier, and M. Auli. 2016. Neural Text Generation from Structured Data with Application to the Biography Domain . In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

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

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. *Preprint*, arXiv:2109.07958.

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

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

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

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *ArXiv*, abs/1808.08745.

Vipula Rawte, Swagata Chakraborty, Agnibh Pathak, Anubhav Sarkar, S. M Towhidul Islam Tonmoy, Aman Chadha, Amit P. Sheth, and Amitava Das. 2023. The troubling emergence of hallucination in large language models – an extensive definition, quantification, and prescriptive remediations. *Preprint*, arXiv:2310.04988.

Reuters. 2023. Alphabet shares dive after google ai chatbot bard flubs answer in ad.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. 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, Vancouver, Canada. Association for Computational Linguistics.

Derek Tam, Anisha Mascarenhas, Shiyue Zhang, Sarah Kwan, Mohit Bansal, and Colin Raffel. 2023. Evaluating the factual consistency of large language

models through news summarization. *Preprint*, arXiv:2211.08412.

Liyan Tang, Tanya Goyal, Alexander R. Fabbri, Philippe Laban, Jiacheng Xu, Semih Yavuz, Wojciech Kryściński, Justin F. Rousseau, and Greg Durrett. 2023. Understanding factual errors in summarization: Errors, summarizers, datasets, error detectors. *Preprint*, arXiv:2205.12854.

Liyan Tang, Igor Shalyminov, Amy Wing mei Wong, Jon Burnsky, Jake W. Vincent, Yu'an Yang, Siffi Singh, Song Feng, Hwanjun Song, Hang Su, Lijia Sun, Yi Zhang, Saab Mansour, and Kathleen McKeown. 2024. Tofueval: Evaluating hallucinations of llms on topic-focused dialogue summarization. *Preprint*, arXiv:2402.13249.

Ars Technica. 2024. Air canada must honor refund policy invented by airline's chatbot.

Vectara. 2023. Vectara hughes hallucination evaluation model.

Yuxia Wang, Minghan Wang, Muhammad Arslan Manzoor, Fei Liu, Georgi Georgiev, Rocktim Jyoti Das, and Preslav Nakov. 2024. Factuality of large language models in the year 2024. *Preprint*, arXiv:2402.02420.

Miriam Wanner, Seth Ebner, Zhengping Jiang, Mark Dredze, and Benjamin Van Durme. 2024. A closer look at claim decomposition. *arXiv preprint arXiv:2403.11903*.

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.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2024. Hallucination is inevitable: An innate limitation of large language models. *Preprint*, arXiv:2401.11817.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. AlignScore: Evaluating factual consistency with a unified alignment function. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11328–11348, Toronto, Canada. Association for Computational Linguistics.

Rongxin Zhu, Jianzhong Qi, and Jey Han Lau. 2023. Annotating and detecting fine-grained factual errors for dialogue summarization. *Preprint*, arXiv:2305.16548.

## A Prompts

We show the following prompts: claim extraction in Figure 3; claim classification *without CoT + all-claims-eval* in Figure 4; *with CoT + all-claims-eval* prompt in Figure 5; *without CoT + one-claim-eval* prompt in Figure 6; *with CoT + one-claim-eval* prompt in Figure 7. Note that we have truncated the examples from the prompts to save space.

## B Error Type Classification Examples

See Table 9 for error type examples.

## C Qualitative Example

We show example #1 in Table 10. Note that news article is the context input document, and Cohere Command output summary is the response under evaluation. We show one claim extracted from the response for analysis. Note that the human annotated label for the claim is "Absent" since the news article does not mention the status of "Chris Hadfield". However, HalluMeasure W/o CoT model labels the claim as "Supported". This could be because the model does not reason properly when generating the class label for a claim. When HalluMeasure is executed W/ CoT prompting, the claim is correctly labeled as "Absent". In addition, we show the explanation automatically generated by our model about why the claim is labeled "Absent".

| Error type | Example context | Example claim with error |
|---|---|---|
| Number | *The company's drones will begin delivering products in Sydney in early 2014.* | The company's drone deliveries will begin in 2013. |
| Entity | *New Zealand company Martin Jetpack...* | Martin Jetpack is based in Sydney. |
| False Concatenation | *The company will be listed on the stock exchange within the next few months.... Their product will be released early next year.* | The product will be released within the next few months. |
| Attribution Failure | *According to Cooper, drone technology is currently under-regulated...* | Drone technology is under-regulated. |
| Overgeneralization | *This product is designed with teachers and students of STEM in mind...* | This product develops math skills. |
| Reasoning Error | *We depend on the Atlas V rocket, which carries many of our most important satellites and is powered by the Russian-made RD-180 rocket engine.* | We rely on Russia's Atlas V rocket |
| Hyperbole | *The technology will significantly improve driver safety.* | The technology will revolutionize driver safety. |
| Temporal | *The company will use drones to deliver packages.* | The company uses drones. |
| Context-based meaning error | *The package includes a Bluetooth system that lets users turn their Roomba into a DJ.* | DJs can instruct the robot. |
| Other | [Context contains no information about the army unveiling a robot called Atlas.] | The army previously unveiled the Atlas robot. |

Table 9: Examples of specific subtypes of hallucinations.

**Task**

```
A claim is a short sentence containing a single piece of
information.
You will extract claims from a given text inside <text></text> XML
tags.
```

**Task-rules**

```
Here are the "Task-rules" you must follow when generating the
claims.
<task-rules>
 <rule>The claim should be entirely self-contained. For instance, the
claim should be comprehended without relying on other claims.</rule>
 <rule>The claim should not contain pronouns. If there are pronouns
in the input text, replace them with their corresponding nouns when
generating the claims.</rule>
 <rule>The claim should not exceed 15 words.</rule>
 <rule>You will always output a list of the extracted claims.</rule>
 <rule>You will always change double quotes to single quotes in the
claims. For example, write 'glass' instead of "glass".</rule>
</task-rules>
```

**Example**

```
An example is given below:
<example>
...
</example>
```

**Input**

```
<text>:
 #TARGET_TEXT#
</text>
\n\nAssistant:
```

Figure 3: Claim extractor prompt.

**Task**

```
<task>
You will act as an expert annotator to evaluate a claim against a provided source text.
The source text will be given within <source>...</source> XML tags
The claims to evaluate will be provided within <claims>...</claims> XML tags.

For each claim, follow these steps:

Step 1: Read and fully understand the claim. It is a short, standalone sentence containing a single piece of information related to the source text.
Step 2: Write the most appropriate label for the claim based on the source text. The definitions of the labels are provided below.

1. supported: The claim is definitely true according to the source text.
2. contradicted: The claim is definitely false according to the source text.
3. absent: The source text provides no information to confirm or deny the claim.
4. partially supported: The claim is mostly consistent with the source text, but some words are added, omitted, or inaccurate. This is a 'near miss' label. If either '2-contradicted' or '3-absent' seems like an appropriate label, use these instead. Note that claims that should be attributed to a particular source (e.g. quotes) are '4-partially supported' if the attribution is missing.
5. unevaluatable: The claim cannot be interpreted as a statement to evaluate against the source text (e.g. it is a question or instruction).

Step 3: If the label in Step 2 is '2-contradicted', '3-absent', or '4-partially supported', then write the specific error type (i.e., sublabel) for the claim. Given below are the definitions of those specific error types.

1. number: the claim has a different number than the original context (e.g., 50m vs. 60m).
2. entity: the claim includes swapped or incorrectly selected entities (e.g., one named entity vs. another).
3. false-concat: the claim inappropriately combines information about two or more different entities or events.
4. attribution-failure: the claim does not correctly attribute the information it contains, e.g. failing to attribute quoted material to the correct person or entity named in the article.
5. overgeneralization: the claim is based on accurate contextual information but is too broad or too general to be supported by the context.
6. reasoning-error: the claim is based on accurate contextual information but includes a reasoning error which makes the claim inaccurate.
7. hyperbole: the claim is based on accurate contextual information but is inappropriately strengthened or overstated.
8. temporal: the claim does not accurately incorporate tense, modality (e.g. might vs. will) or time reference in relation to the context.
9. context-based-meaning: the claim includes incorrect interpretation of idiomatic language, homonyms, or words with multiple meanings, therefore failing to capture the intended meaning
10. other: all other types of errors are captured in this category.

Repeat these steps for each claim provided.

You will output a list of dictionaries. Each dictionary will have below format:
{
"claim": "<CLAIM TEXT>",
"label": "<CLAIM LABEL>",
"sublabel": "CLAIM SUBLABEL"
}
</task>
```

**Example**

```
Given below are the examples for you to comprehend the task:
<example>
...
</example>
```

**Source**

```
<source>:
 {{SOURCE_TEXT}}
</source>
\n
<claims>
 {{TARGET_CLAIMS}}
</claims>
```

Figure 4: Claim classifier prompt for *without CoT + all-claim-eval* setup.

**Task**

```
<task>

You will act as an expert annotator to evaluate claims against a provided source text.
The source text will be given within <source>...</source> XML tags
The claims to evaluate will be provided within <claims>...</claims> XML tags.

For each claim, follow these steps:

Step 1: Read and fully understand the claim. It is a short, standalone sentence containing a single
piece of information related to the source text.
Step 2: Thoroughly analyze how the claim relates to the information in the source text. Then, write
your reasoning in 1-3 sentences to determine the most appropriate label to describe the claim's
truthfulness based on the source text. The definitions of the labels are provided below.

1. supported: The claim is definitely true according to the source text.
2. contradicted: The claim is definitely false according to the source text.
3. absent: The source text provides no information to confirm or deny the claim.
4. partially supported: The claim is mostly consistent with the source text, but some words are
   added, omitted, or inaccurate. This is a 'near miss' label. If either '2-contradicted' or '3-
   absent' seems like an appropriate label, use these instead. Note that claims that should be
   attributed to a particular source (e.g. quotes) are '4-partially supported' if the attribution
   is missing.
5. unevaluatable: The claim cannot be interpreted as a statement to evaluate against the source
   text (e.g. it is a question or instruction).

Step 3: Write the label for the claim based on your reasoning in Step 2.
Step 4: If the label in Step 3 is '2-contradicted', '3-absent', or '4-partially supported', then
thoroughly analyze the specific error or inaccuracy present in the claim based on provided source
text. Then, provide your reasoning in 1-3 sentences to determine the error. Given below are the
definitions of those specific error types (i.e., sublabels). However, if the label in Step 3 is '1-
supported', simply write 'None - claim is supported', and set the sublabel to 'None'.

1. number: the claim has a different number than the original context (e.g., 50m vs. 60m).
2. entity: the claim includes swapped or incorrectly selected entities (e.g., one named entity vs.
   another).
3. false-concat: the claim inappropriately combines information about two or more different
   entities or events.
4. attribution-failure: the claim does not correctly attribute the information it contains, e.g.
   failing to attribute quoted material to the correct person or entity named in the article.
5. overgeneralization: the claim is based on accurate contextual information but is too broad or
   too general to be supported by the context.
6. reasoning-error: the claim is based on accurate contextual information but includes a reasoning
   error which makes the claim inaccurate.
7. hyperbole: the claim is based on accurate contextual information but is inappropriately
   strengthened or overstated.
8. temporal: the claim does not accurately incorporate tense, modality (e.g. might vs. will) or
   time reference in relation to the context.
9. context-based-meaning: the claim includes incorrect interpretation of idiomatic language,
   homonyms, or words with multiple meanings, therefore failing to capture the intended meaning
10. other: all other types of errors are captured in this category.

Step 5: Write the sublabel based on your reasoning in Step 4.

Repeat these steps for each claim provided.

You will output a list of dictionaries. Each dictionary will have below format:
{
"claim": "<CLAIM TEXT>",
"thought_for_label": "<THOUGHT FOR LABEL FOR THE CLAIM>",
"label": "<CLAIM LABEL>",
"thought_for_sublabel": "<THOUGHT FOR SUBLABEL FOR THE CLAIM>",
"sublabel": "CLAIM SUBLABEL"
}

</task>
```

**Example**

```
Given below are the examples for you to comprehend the task:
<example>
...
</example>
```

**Source**

```
<source>:
 {{SOURCE_TEXT}}
</source>
\n
<claims>
 {{TARGET_CLAIMS}}
</claims>
```

Figure 5: Claim classifier prompt for *with CoT + all-claims-eval* setup.

**Task**

<task>

You will act as an expert annotator to evaluate a claim against a provided source text.
The source text will be given within <source>...</source> XML tags
The claim to evaluate will be provided within <claim>...</claim> XML tags.

For each claim, follow these steps:

Step 1: Read and fully understand the claim. It is a short, standalone sentence containing a single piece of information related to the source text.
Step 2: Write the most appropriate label for the claim based on the source text. The definitions of the labels are provided below.

1. supported: The claim is definitely true according to the source text.
2. contradicted: The claim is definitely false according to the source text.
3. absent: The source text provides no information to confirm or deny the claim.
4. partially supported: The claim is mostly consistent with the source text, but some words are added, omitted, or inaccurate. This is a 'near miss' label. If either '2-contradicted' or '3-absent' seems like an appropriate label, use these instead. Note that claims that should be attributed to a particular source (e.g. quotes) are '4-partially supported' if the attribution is missing.
5. unevaluatable: The claim cannot be interpreted as a statement to evaluate against the source text (e.g. it is a question or instruction).

Step 3: If the label in Step 2 is '2-contradicted', '3-absent', or '4-partially supported', then write the specific error type (i.e., sublabel) for the claim. Given below are the definitions of those specific error types.

1. number: the claim has a different number than the original context (e.g., 50m vs. 60m).
2. entity: the claim includes swapped or incorrectly selected entities (e.g., one named entity vs. another).
3. false-concat: the claim inappropriately combines information about two or more different entities or events.
4. attribution-failure: the claim does not correctly attribute the information it contains, e.g. failing to attribute quoted material to the correct person or entity named in the article.
5. overgeneralization: the claim is based on accurate contextual information but is too broad or too general to be supported by the context.
6. reasoning-error: the claim is based on accurate contextual information but includes a reasoning error which makes the claim inaccurate.
7. hyperbole: the claim is based on accurate contextual information but is inappropriately strengthened or overstated.
8. temporal: the claim does not accurately incorporate tense, modality (e.g. might vs. will) or time reference in relation to the context.
9. context-based-meaning: the claim includes incorrect interpretation of idiomatic language, homonyms, or words with multiple meanings, therefore failing to capture the intended meaning
10. other: all other types of errors are captured in this category.

For the claim, you will output a dictionary with the below format:

```
{
"claim": "<CLAIM TEXT>",
"label": "<CLAIM LABEL>",
"sublabel": "CLAIM SUBLABEL"
}
```

</task>

**Example**

Given below are the examples for you to comprehend the task. Each example includes a list of claims and a corresponding list of dictionaries as responses to aid your understanding. However, the input you will receive consists of a single claim, and you will need to provide a dictionary response for that particular claim.

<example>

...

</example>

**Source**

<source>:

{{SOURCE_TEXT}}

</source>

\n

<claims>

{{TARGET_CLAIMS}}

</claims>

Figure 6: Claim classifier prompt for *without CoT + one-claim-eval* setup.

**Task**

```
<task>

You will act as an expert annotator to evaluate claims against a provided source text.
The source text will be given within <source>...</source> XML tags
The claim to evaluate will be provided within <claim>...</claim> XML tags.

For each claim, follow these steps:

Step 1: Read and fully understand the claim. It is a short, standalone sentence containing a single
piece of information related to the source text.
Step 2: Thoroughly analyze how the claim relates to the information in the source text. Then, write
your reasoning in 1-3 sentences to determine the most appropriate label to describe the claim's
truthfulness based on the source text. The definitions of the labels are provided below.

1. supported: The claim is definitely true according to the source text.
2. contradicted: The claim is definitely false according to the source text.
3. absent: The source text provides no information to confirm or deny the claim.
4. partially supported: The claim is mostly consistent with the source text, but some words are
   added, omitted, or inaccurate. This is a 'near miss' label. If either '2-contradicted' or '3-
   absent' seems like an appropriate label, use these instead. Note that claims that should be
   attributed to a particular source (e.g. quotes) are '4-partially supported' if the attribution
   is missing.
5. unevaluatable: The claim cannot be interpreted as a statement to evaluate against the source
   text (e.g. it is a question or instruction).

Step 3: Write the label for the claim based on your reasoning in Step 2.
Step 4: If the label in Step 3 is '2-contradicted', '3-absent', or '4-partially supported', then
thoroughly analyze the specific error or inaccuracy present in the claim based on provided source
text. Then, provide your reasoning in 1-3 sentences to determine the error. Given below are the
definitions of those specific error types (i.e., sublabels). However, if the label in Step 3 is '1-
supported', simply write 'None - claim is supported', and set the sublabel to 'None'.

1. number: the claim has a different number than the original context (e.g., 50m vs. 60m).
2. entity: the claim includes swapped or incorrectly selected entities (e.g., one named entity vs.
   another).
3. false-concat: the claim inappropriately combines information about two or more different
   entities or events.
4. attribution-failure: the claim does not correctly attribute the information it contains, e.g.
   failing to attribute quoted material to the correct person or entity named in the article.
5. overgeneralization: the claim is based on accurate contextual information but is too broad or
   too general to be supported by the context.
6. reasoning-error: the claim is based on accurate contextual information but includes a reasoning
   error which makes the claim inaccurate.
7. hyperbole: the claim is based on accurate contextual information but is inappropriately
   strengthened or overstated.
8. temporal: the claim does not accurately incorporate tense, modality (e.g. might vs. will) or
   time reference in relation to the context.
9. context-based-meaning: the claim includes incorrect interpretation of idiomatic language,
   homonyms, or words with multiple meanings, therefore failing to capture the intended meaning
10. other: all other types of errors are captured in this category.

Step 5: Write the sublabel based on your reasoning in Step 4.

You will output a list of dictionaries. Each dictionary will have below format:
{
"claim": "<CLAIM TEXT>",
"thought_for_label": "<THOUGHT FOR LABEL FOR THE CLAIM>",
"label": "<CLAIM LABEL>",
"thought_for_sublabel": "<THOUGHT FOR SUBLABEL FOR THE CLAIM>",
"sublabel": "CLAIM SUBLABEL"
}

</task>
```

**Example**

```
Given below are the examples for you to comprehend the task. Each example includes a list of
claims and a corresponding list of dictionaries as responses to aid your understanding. However,
the input you will receive consists of a single claim, and you will need to provide a dictionary
response for that particular claim.
<example>
...
</example>
```

**Source**

```
<source>:
 {{SOURCE_TEXT}}
</source>
\n
<claims>
 {{TARGET_CLAIMS}}
</claims>
```

Figure 7: Claim classifier prompt for *with CoT + one-claim-eval* setup.

**Context (News Article):** (CNN)Elon Musk has built a $12 billion company in an endeavor to pave the way to Mars for humanity. He insists that Mars is a ̈long-term insurance policyf̈or ẗhe light of consciousnessïn the face of climate change, extinction events, and our recklessness with technology. On the other hand, astronaut Chris Hadfield is skeptical: Ḧumanity is not going extinct,ḧe told me. He added: T̈here's no great compelling reason to go, apart from curiosity, and that's not going to be enough to sustain the immense cost necessary with the technology that exists right now.B̈ut I question our future, stuck here on Earth. Our environment is a highly balanced system and we are the destabilizing element. Pursuing g̈reenïnitiatives is no long-term solution to the wall we're hurtling towards, they're speed bumps. If this is where humankind is destined to remain, then we shall find ourselves fighting over whatever is left of it. Politically speaking, sending humans into space brings nations together – the International Space Station stood as the physical manifestation of the reunification of the USA and Russia and is now a platform for broader international cooperation. Space exploration is also inspiring: during NASA's Apollo program to the Moon, the number of graduates in mathematics, engineering and the sciences in the US doubled. Igniting the imagination of that generation helped propel the US into the dominant position it's held since the 1960s. What could a Mars program do? Wouldn't the Moon, so much nearer than Mars, be a better first step? Actually, no – it's just too different. It's better to test hardware and train people in analogs on Earth, such as the geologically similar high-altitude desert in Utah or the cold and dry Canadian Arctic desert. Why the European Space Agency has declared the Moon a stepping-stone to Mars is beyond me, as doing so increases the cost of a Mars program hugely. It takes about 50% more energy to put something on the surface of the Moon than it does on Mars. The Martian atmosphere can be used to slow down approaching spacecraft, instead of the need for extra fuel to slow the descent. It would also mean developing two different sets of landing techniques and hardware. There are reasons to go to the Moon, just not if your ultimate destination is Mars. Even colonizing the Moon is questionable: it simply hasn't the resources to sustain an advanced colony. Mars has fertile soil, an abundance of water (as ice), a carbon-dioxide rich atmosphere and a 24-and-a-half hour day. The Moon's soil is not fertile, water is as rare, it has no effective atmosphere, and a 708-hour day. It's feasible to introduce biological life to Mars, but not the Moon. With only a relatively small push, Mars could be returned to its former warm, wet, hospitable state. Raising the temperature at the south pole by a few degrees would see frozen CO2 in the soil begin to gasify. As a greenhouse gas, it would further raise the temperature, gasifying more CO2 in a self-sustained global-warming process. Eventually, water frozen into the soil would liquefy, covering half of the planet. After about a century, Mars would settle down with an atmosphere about as dense as the lowland Himalayas and a climate suitable for T-shirts. Hadfield warns that ẅe need to invent a lot of thingsb̈efore going to Mars, and that ẗhere's no great advantage to being the early explorers who die.̈ Few would disagree with that, but what are the challenges a crewed mission to Mars faces? Radiation: An astronaut would receive a lifetime allowable dose of radiation in a single 30-month round-trip, including 18 months on the surface. But this is only equivalent to increasing the lifetime cancer risk from about 20% to 23%. As the majority of this is received in transit between planets, with proper radiological protection on the ship, it would actually be (radiologically speaking) healthier for an astronaut to live on Mars with a radiation dose of 0.10 sieverts per year than to smoke on Earth at 0.16 sieverts per year. There is no single practical solution to the radiation problem. One strategy I helped develop was to optimise the internal layout of the equipment and structures in the Mars habitat module to minimise exposure – placing existing bulk in all the right places. This reduced exposure by about 20%, without adding any mass. Even taking empty sandbags, packing them with Martian soil and putting them on the roof would be a simple and effective measure on Mars. Radiation is an issue to tackle, but it's not a deal-breaker. Power: Ẅe need a compact energy source,s̈ays Hadfield. Ẅe cannot be relying on the tiny bit of solar power that happens to arrive at that location.Ẅhile the solar energy reaching the surface of Mars is about half that on Earth, this isn't a show-stopper. A quick back-of-the-envelope calculation shows that to power the equivalent of an average U.S. household on Mars, even through dust storms, one would need an array of solar panels totalling six metres square – very achievable. Reduced gravity: The effects of microgravity on astronauts' health have been studied for decades, and a range of techniques have been developed to mitigate the wasting effects on muscle and bone. With Martian gravity around a third of that on Earth, it would take astronauts a couple of days to acclimatize, and perhaps a few months to fully adapt. NASA and ESA have been developing an under-suit that compresses the body to overcome the negative effects of a reduction in pressure and gravity. However, biological adaption could be made easier if microgravity were avoided altogether. The spacecraft could be spun in-transit to generate an artificial gravity that slowly decayed, simulating a transition from Earth to Mars gravity (and vice versa) over the six-month journey. Ultimately, until humans are actually living on other planets it's unlikely we'll solve or even recognise all the subtle long-term health problems associated with reduced gravity. And who's to say what the advances in bio-engineering and technology will make the human body capable of when that time comes? Life on Mars: If there's life on Mars, even if it's microbial, should we be allowed to spread to the planet, potentially risking its extinction? I find this question strange – as Chris McKay put it: Ẅe commit microbial genocide every time we wash our hands.̈ We engineer and farm the complex life around us as systematically and as cheaply as possible. Billions of people eat the carcasses of organisms that were thinking and breathing only days before. Why, all of a sudden, should Martian microbes be given such sanctity? It should certainly be studied, but it shouldn't prevent our spreading. Back contamination: Conversely, the question of whether some Martian plague might accidentally be introduced to Earth should be taken seriously – but not blown out of proportion. There's only a remote chance that Martian life might be hazardous. The things that kill us do so because they've evolved in lock-step with us in a continual evolutionary arms race. Any Martian life will have evolved independently and is unlikely to be capable even of interacting with Earth life on a molecular level. As Robert Zubrin put it: T̈rees don't get colds and humans don't get Dutch Elm Disease.P̈sychology: Depending on relative orbits, sending a message between Earth and Mars can take between three and 22 minutes. This loss of real-time communication will leave astronauts feeling cut-off and alone. Hadfield says that it's vital to keep up crew morale and motivation: Ö̈nce you get any distance away on any sort of voyage, the epic-ness disappears, the reality becomes the foreground, and the applause is long gone.C̈ost: A crewed Mars program would cost the equivalent of a few weeks of the U.S. defense budget. The US plans on spending about 10 times more on nuclear weapons than on space exploration over the coming decade. The UK government spends about as much on gastric band surgery through the NHS as it does on its space activities. So while a Mars program certainly has challenges to overcome, the technological gap between us and Mars is far smaller than it was for the Moon program in the 1960s. And the prospects the Red Planet holds for humanity are far greater. Copyright 2015 The Conversation. Some rights reserved"

**Response (Cohere Summary):** The Conversation article presents an argument in favor of pursuing space exploration, specifically a manned mission to Mars. It explains that while Chris Hadfield, a retired astronaut, is skeptical about the necessity of such an endeavor, the author believes that it is imperative for humanity to explore space. The author points out that Earth's environment is delicately balanced and that human activity has the potential to disrupt it significantly. Therefore, humans should look to expand their presence beyond Earth in order to ensure their long-term survival. In addition, the article addresses common concerns and challenges of a Mars mission, such as the high cost, potential dangers, and lack of practical benefits. However, it argues that these challenges are not insurmountable, and that the benefits of exploring Mars are immense. Ultimately, the author concludes that while Mars may seem like a distant and impractical goal, it is a worthwhile endeavor that has the potential to propel humanity into a new era of innovation and progress.

**HalluMeasure Extracted Claim:** Chris Hadfield is a retired astronaut.

**Human Annotated Label:** Absent

**HalluMeasure W/o CoT Label**: Supported

**HalluMeasure W/ CoT Label**: Absent

**HalluMeasure W/ CoT Label Explanation**: The source text refers to Chris Hadfield as an astronaut and quotes him directly, but does not explicitly state whether he is currently active or retired.

Table 10: Sample Result: HalluMeasure W/ and W/o CoT comparison on a claim from news article context document in TechNewsSumm dataset.