Fine-Grained Detection of Context-Grounded Hallucinations Using LLMs

Yehonatan Peisakhovsky[T*] **Zorik Gekhman**[T*] **Yosi Mass**[I] **Liat Ein-Dor**[I] **Roi Reichart**[T]

[*] Equal contribution; author order was chosen randomly.

[T]Technion - Israel Institute of Technology [I]IBM Research

yonip1997@gmail.com zorikgekhman@gmail.com roiri@technion.ac.il

Abstract

Context-grounded hallucinations are cases where model outputs contain information not verifiable against the source text. We study the applicability of LLMs for *localizing* such hallucinations, as a more practical alternative to existing complex evaluation pipelines. In the absence of established benchmarks for *meta-evaluation* of hallucinations *localization*, we construct one tailored to LLMs, involving a challenging human annotation of over 1,000 examples. We complement the benchmark with an LLM-based evaluation protocol, verifying its quality in a human evaluation. Since existing *representations* of hallucinations limit the types of errors that can be expressed, we propose a new representation based on free-form textual descriptions, capturing the full range of possible errors. We conduct a comprehensive study, evaluating four large-scale LLMs, which highlights the benchmark's difficulty, as the best model achieves an F1 score of only 0.67. Through careful analysis, we offer insights into optimal prompting strategies for the task and identify the main factors that make it challenging for LLMs: (1) a tendency to incorrectly flag missing details as inconsistent, despite being instructed to check only facts in *the output*; and (2) difficulty with outputs containing factually correct information absent from the source - and thus not verifiable - due to alignment with the model's parametric knowledge.

1 Introduction

The ability to generate responses conditioned explicitly on a given input is critical for many downstream tasks, including summarization (Zhang et al., 2020), open book question-answering (Nakano et al., 2021) and retrieval-augmented generation (Lewis et al., 2020). In such context-grounded setups, a response is considered *factually consistent* if any piece of information it contains is supported by the source text (Bohnet et al., 2022).

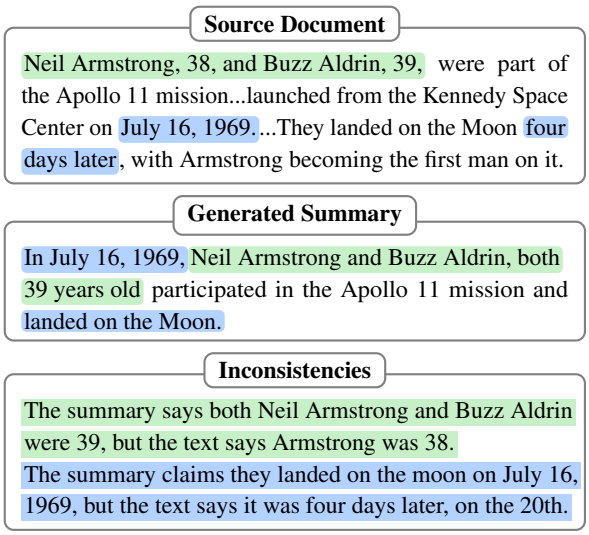

Figure 1: Since factual inconsistencies can be complex and hard to represent, we propose a representation based on free-form textual descriptions in natural language.

Early work on factual consistency *evaluation* used a *binary* setup, classifying the entire output as consistent or not (Honovich et al., 2022; Laban et al., 2022; Gekhman et al., 2023). Such binary classification not only overlooks severity by scoring outputs with many errors the same as those with only one, but also fails to *localize* hallucinations (pinpointing which parts of the output are not supported): a critical capability for analyzing model failure patterns and enabling targeted corrections. This motivated a shift to *fine-grained* evaluation, analyzing smaller output units such as entities (Cao et al., 2022), spans (Mishra et al., 2024), question-answer (QA) pairs (Honovich et al., 2021), or atomic facts (Min et al., 2023).

While the shift to fine-grained evaluation marks an important step towards more reliable evaluation, existing methods still have fundamental limitations. First **(i)** the *error-representation* in prior work using formats such as entities, spans, QA pairs, and atomic facts, constrain the error types that can be captured, making it infeasible to represent the full

Findings of the Association for Computational Linguistics: ACL 2026, pages 38242–38266

spectrum of factual errors (Table 1 and §2). Second **(ii)** most methods output a continuous score, rather than identifying specific errors (Min et al., 2023), which limits their usefulness for error localization.

The third limitation **(iii)** is that existing methods are often based on complex, multi-stage pipelines that can be difficult to train, maintain, and deploy effectively, limiting practical usability. A promising, simpler alternative is to leverage large language models (LLMs) for end-to-end localization. However, the extent to which LLMs can reliably localize factual inconsistencies remains unclear.[1] Lastly **(iv)** the lack of established frameworks for reproducible automatic *meta-evaluation* makes it challenging to compare different evaluation systems. This limitation is even more pronounced when using LLMs, given their rapid development and wide variety.

We argue that existing evaluation pipelines should be replaced by LLMs, increasing practical usability **(iii)**. To support automatic evaluation **(iv)**, we curate the **FINAL** benchmark for evaluating LLMs on the task of **F**actual **I**nconsistencies **L**ocalization. It is constructed from partially annotated examples in the DeFacto dataset (Liu et al., 2023), with the curation involving a challenging human annotation task of over 1,000 examples.

To capture the full range of errors **(i)**, we represent inconsistencies as free-form natural language *descriptions*, which provide maximal flexibility and align naturally with the strengths of LLMs. This choice also helps to localize errors **(ii)**, as the LLM generates a list of interpretable descriptions rather than a single continuous score. To enable the evaluation of such lists of errors, we design an LLM-based evaluation protocol that complements our benchmark, and whose judgment quality is validated through human evaluation.

We show that our benchmark is challenging even for strong LLMs by evaluating four large-capacity models using various prompting strategies: Llama-3-405B, GPT-4o, Gemini-Pro, and Claude-Sonnet, with the best-performing model achieving an F1 score of 0.67. Our analysis shows that (1) reasoning aids in addressing this task and (2) two-step approaches, where the model first classifies the summary as consistent or inconsistent and then identifies individual errors, tend to perform worse than end-to-end localization due to a conservative behavior in the binary step, leading to low recall.

Lastly, we conduct a comprehensive error analysis grouping localization failures into categories with interpretable meanings, which allows us to identify two key weaknesses: (1) LLMs tend to incorrectly treat missing information in the output as inconsistent, even when explicitly instructed to examine only facts present in the output, and (2) LLMs struggle when the output has correct information not present in the source text: although such information cannot be verified by the source, its alignment with the model's parametric knowledge causes the model to misclassify it as consistent. To summarize, our contributions are as follows:

- We create a benchmark for the meta-evaluation of LLMs on the task of fine-grained factual consistency evaluation.[2]
- We propose a new paradigm for error representation based on free-form textual description, which allows to represent *any* possible error.
- We design an LLM-based evaluation protocol for error localization and validate its quality in a human evaluation.
- We conduct a comprehensive evaluation of four large-scale LLMs on the task, exploring different prompting strategies and comparing the end-to-end and two-step paradigms.
- We conduct a comprehensive error analysis, group failures into meaningful categories, and diagnose the underlying drivers behind these model failures.

2 Representing Factual Inconsistencies via Descriptions in Natural Language

In this section we propose a new representation of errors.[3] We first discuss the limitations of the representations from previous work (see §7) using Table 1 as a running example.

Entities limit coverage as they cannot represent errors in verbs, adjectives, general nouns, or more nuanced errors. For example, in Table 1 the summary claims a *"record number of sales"*, while there is no evidence that a new record was set. Such error cannot be captured by highlighting an entity.[4]

[1] LLMs are used for intermediate steps in pipeline-based methods (Min et al., 2023), but not for end-to-end evaluation.

[2] Data and code can be found here: https://github.com/yonip97/The_final_benchmark

[3] For brevity, we use *error* throughout the paper to refer specifically to factual inconsistencies.

[4] In Appendix A we estimate the prevalence of errors that cannot be represented as entities.

Text	After surpassing 250,000 units sold on Amazon, Philips attracted the attention of the authorities, prompting an investigation... After investigating for a month, the police concluded that Philips could continue operating without restrictions.
Summary	Following a record number of sales, Amazon was investigated for a month and ordered by the police to cease operations.
Entities	Following a record number of sales, Amazon was investigated for a month and ordered by the police to cease operations.
Spans	Following a record number of sales, Amazon was investigated for a month and ordered by the police to cease operations.
Atomic Facts	• Amazon made a record number of sales. ✗ • Amazon was investigated. ✗ • The investigation lasted a month. ✓ • The police ordered Amazon to cease operations. ✗ → The police, Amazon, Ordered to cease operations
QA Pairs	• What did Amazon accomplish? A record number of sales ✗ • Who was investigated? Amazon ✗ • What was the police's action after the investigation? Order to cease operations ✗ • When was someone investigated? Following Amazon's record number of sales ✗ → Following, Amazon, Record number, Sales • How long did the investigation take? A month ✓ • Who investigated something? The police ✓
Descriptions (Ours)	• The summary calls the sales "a record," but the text says "surpassing 250,000 units" without mentioning a record. • The summary refers to Amazon, but the text says it was Philips being investigated. • The summary says the company was ordered to cease operations, but the text says it was authorized to continue.

Table 1: Examples of different strategies for annotating factual errors. The summary has three errors: (1) the number of sales is not said to be a record, (2) the company is Philips, not Amazon, and (3) the company was not ordered to cease operations but was allowed to continue. In both Atomic Facts and QA pairs, all facts are explicitly listed, and the consistent and inconsistent ones are marked with ✓and ✗, respectively.

Spans can be subjective (Mishra et al., 2024), as there are often many possible ways to annotate an error. For instance, in Table 1 we could highlight *"record number of sales"*, *"record number"* or *"record"*. In addition, some errors do not correspond to a contiguous text sequence. E.g., in the final highlighted error the summary says *"ordered by the police to cease operations"* while the text states that *"Philips could continue operating without restrictions"*. Highlighting this span could be interpreted to mean that the order was not issued by the police. Those issues introduce evaluation challenges, making it difficult to compare predicted inconsistencies against ground-truth annotations.[5] Consequently, previous work resorted to simplified settings: sentence-level evaluation, which lacks granularity (Mishra et al., 2024), and character-level span overlap, which is sensitive to minor shifts in span boundaries (Niu et al., 2024).

Atomic facts and *QA pairs* can be vague. For example, in Table 1 the inconsistent label for the atomic fact *"The police ordered Amazon to cease operations"*, does not indicate whether the error lies in the company name, the authority issuing the order, or the action itself. Another example is the QA pair *"When was something investigated? Following Amazon's record number of sales."*, which includes several facts: the company (Amazon), the timing (after the record sales), and the outcome itself (record number of sales), making it unclear which part is inconsistent. Moreover, generating questions from factually inconsistent summaries can cause those same factual errors to propagate into the questions Kamoi et al. (2023a).

Lastly, we aim to create a benchmark for LLMs, and existing representations can be less natural for LLMs that output text in natural language. We propose to use *descriptions*: free-form explanations in natural language describing the nature of the error. While descriptions address the limited expressivity of existing representations, they introduce an evaluation challenge in comparing the model-generated descriptions to gold references. To address this, in §4.2 we design an LLM-based evaluation protocol and verify its quality through human evaluation.

One practical consideration is that descriptions require users to manually link the explanation back to the summary text. In practice, this could be addressed with a hybrid approach: systems could use the generated descriptions to identify and highlight relevant spans when possible, while simultaneously presenting a textual explanation to resolve ambiguity. Because this constitutes a usability enhancement rather than a strict evaluation requirement, we leave it for future work.

3 The FINAL Benchmark

To create the **FINAL** benchmark for evaluating LLMs' performance on **F**actual **I**nconsistencies **L**ocalization, we need to (1) obtain a collection of source texts paired with corresponding outputs and (2) annotate them with factual errors as described in §2. We chose to build on the DeFacto dataset (Liu et al., 2023), since it contains document-summary

[5] In Appendix A we estimate the prevalence of errors that cannot be represented as spans.

Type	Text	Summary	DeFacto Explanation	Our Descriptions	Comment
Extraction	Robin Clark, 44, was shot in the leg in the car park at Shenfield station ... has since returned to his job at RP Martin in London... a man from Essex has been arrested ...	A 46-year-old man has been arrested in connection with the shooting of a security guard at a London Underground station.	No mention of his age, that the other man was a security guard, and it was not located at London Underground but in a park near Shenfield.	No mention of his age in the source text. No mention that the other man was a security guard in the source text It was not located at London Underground but in a park near Shenfield.	The individual inconsistencies are apparent in the original explanation. Only need to separate them into self contained factual inconsistency descriptions.
Decomposition	Platt, 19 and Thomson, 21, have both joined the National League outfit until the end of the season... Blackburn are currently 22nd in the second tier...	Barrow have signed Blackburn Rovers midfielders Ben Platt and Josh Thomson on loan.	Their first names are not mentioned and second it is not mentioned who signed them.	The first name of Platt is not in the text. The first name of Thomson is not in the text. It is not mentioned who signed them in the source text.	The original explanation merges two distinct inconsistencies (2 first names) into a single description.
Vague Explanation	...complaint was made that police did not fully investigated claims against the Sinn Féin president...he had "found no evidence to indicate that [police officers] thinking was influenced by who Mr Adams was"...	Police in Northern Ireland have been cleared of any wrongdoing over their handling of allegations against Gerry Adams.	The summary incorrectly adds info about Northern Ireland and Mr. Adams' first name.	The summary states it was Police in Northern Ireland, while the source text does not mention any location. The summary incorrectly adds Mr. Adams' first name.	The original explanation claims there's an issue with the information about Northern Ireland, but the actual inconsistency is that this location is not mentioned at all in the text.
Missing Explanation	The 46-year-old number one seed defeated his 26-year-old opponent 7-3 ..."Now, thanks to hard work, determination and Teesside steel, I am world champion." ...	England's Martin Durrant has won his first BDO world title with victory over Australia's Scott Noppert.	There is info in the summary not found in the source, e.g. BDO title, etc.	The summary calls it a "BDO world title," but the source doesn't name the organization. Noppert nationality is not in the source text	The original explanation is lacking. It does not cover the inconsistency in the nationality of Noppert.
Irrelevant Information	The 35-year-old victim was attacked outside Barclays Bank ... The men, aged 41 and 42, were arrested on suspicion of murder...	Two men have been arrested after a man was stabbed outside a bank.	Clearly states two men were arrested and the victim was attacked, but not necessarily stabbed.	The summary claims the man was stabbed, but the text only states he was attacked	The original explanation contains information on why the summary is correct, not why it is inconsistent.
Wrong Annotation	... 89 out of 157 school closures between the academic years 2006-07 and 2015-16 were in the nine predominantly-rural council areas...	Almost half of school closures in Wales over the past decade were in rural areas, it has been claimed.	The source text does not say that September 2006 happened 10 years ago or that less than half of the closures were in rural areas.	The source text does not say that less than half of the closures were in rural areas,but more than half.	The explanation wrongly flags "the past decade" as incorrect, though the text supports it, so it is not an inconsistency.
Fabricated	The victim was threatened with a knife and punched during the attack at Exhibition Park in the early hours. Her attacker is described as... Northumbria Police has...	A 19-year-old woman has been raped in Newcastle city centre.	It makes up the entire summary		Summary is almost entirely fabricated and unrelated to the text, making fine-grained annotation meaningless, so those samples were excluded.

Table 2: Examples of annotation operations that were applied to convert the explanations from the DeFacto dataset (Liu et al., 2023) to our error descriptions (§2).

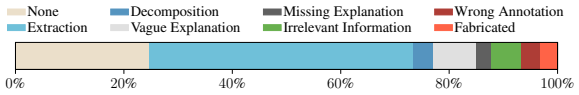

Figure 2: Estimated frequency of each annotation operation on the DeFacto explanations (see Table 2). *None* refers to cases where the explanation described a single error and could be used as-is without modification. This estimation is based on 400 samples, details in §B.2.

pairs that have already been partially annotated. The goal of DeFacto was to explore how human feedback can help revise summaries, so it is annotated with *explanations* for why summaries are factually inconsistent.

Our *descriptions* differ from DeFacto *explanations*: while both are expressed in natural language, the latter often conflate multiple errors within a single claim (see Table 2), which hinders the localization of each individual error. When considering the methodology for converting DeFacto's explanations to descriptions, we found that the conversion was often non-trivial and could not be automated. Moreover, we identified two major annotation problems: (1) many explanations contained issues such as missing or irrelevant information, vague phrasing, or incorrect annotations; and (2) a considerable number of factual inconsistencies were entirely missing from DeFacto. While partial annotations can be useful for correction tasks, where any fix can improve the output, accurate *evaluation* required a higher standard. We therefore designed a rigorous human annotation process, using DeFacto's annotations as a reference, to correct inaccurate labels and identify missing errors, ensuring our benchmark's reliability. Since the annotation task is challenging, we rely on expert annotators rather than crowdworkers: all annotations were performed by the authors of this paper. Given the scale of the dataset, we assign each example to a single annotator to balance quality with coverage.

Phase 1: DeFacto Explanations to Descriptions. Our benchmark is based on randomly selected 1,650 examples from DeFacto, consisting of 1,150 examples labeled as factually inconsistent and 500 labeled as factually consistent (Table 3, first line). For each inconsistent example, we manually converted DeFacto's explanation into a list of descriptions. This process involved several operations: extracting and decomposing spans, revising vague descriptions, adding missing information, removing irrelevant content, and correcting inaccuracies. See Table 2 for an example of each operation and Figure 2 for their frequencies. 25% of the explanations could be used without modification, 48% required relatively straightforward error extraction, while the remaining 27% involved *challenging* annotation operations, that required expert annotators. Summaries classified as "Fabricated" were filtered-out, reducing the number of inconsistent examples from 1,150 to 1,086 (Table 3, second line). The annotation guidelines can be found in §B.1.

Phase 2: Error Enrichment via Human-LLM Collaboration During the annotation we ob-

Source	Inconsistent	Consistent	Total
Original DeFacto	1,150	500	1,650
Phase 1: Explanations to Descriptions	1,086	500	1,586
Phase 2: Error's Enrichment	1,121	284	1,405

Table 3: Dataset statistics at each annotation phase.

served that DeFacto's explanations often omit errors. This undermines our benchmark's reliability: a model that fails to detect a missing error will not get penalized, whereas a model that correctly identifies it will be unfairly penalized.

Since detecting new errors is challenging, we perform LLM-assisted annotation (Wiegreffe et al., 2022; Goel et al., 2023; Lee et al., 2024), where candidate errors detected by an LLM[6] are reviewed by human annotators. To increase coverage, we explicitly prompt the LLM to favor high recall, relying on the human annotators to filter-out false positives (more details in §B.3). This process increased the total amount of annotated errors from 1627 to 2131 (+31%). Importantly, out of the 500 summaries labeled as factually *consistent* in DeFacto, 128 contained factual inconsistencies. In some cases the annotators couldn't determine whether the LLM-based suggestions were actual errors. To reduce subjectivity and maintain a reliable dataset, we filtered-out a total of 181 such examples. To validate this LLM-assisted annotation procedure, we double-annotated 150 examples and found substantial inter-annotator agreement (raw agreement $= 0.88$, Cohen's $\kappa = 0.73$). Full details in §B.3.

Final data statistics are presented in Table 3, third line. Figure 8 presents a histogram of number of errors per-example. We randomly split the data into 140 development and 1,265 test examples.

4 Experimental Setup

We use our benchmark to assess the capabilities of high-capacity LLMs on the task of fine-grained factual consistency evaluation.

4.1 Models and Baselines

We evaluate GPT-4o-2024-11-20 (Achiam et al., 2023), Claude-3.5-sonnet-20241022 (Anthropic, 2024), Gemini-1.5-pro (Google, 2024) and Llama-3.1-405B (Grattafiori et al., 2024). Each evaluated model is prompted to perform the task end-to-end (**E2E**), namely to identify all inconsistencies and generate a list of descriptions to be passed to the LLM judge. We used **Zero-shot**, **Few-shot**, and Chain-of-Thought (**CoT**) prompting (Wei et al., 2022). Full implementation details can be found in §C. We compare to the following baselines:

Pipeline. To compare to traditional evaluation pipelines, we implement a pipeline inspired by the **FactScore** approach (Min et al., 2023). The evaluated LLM first decomposes the summary into atomic facts, and then assesses the factual consistency of each fact and, if any inconsistency is found, generates a description of it. Since a single fact may contain multiple inconsistencies, the model is instructed to describe each inconsistency individually. This process can result in duplicate descriptions since multiple atomic facts may include the same erroneous information, and each fact is evaluated independently (see example in Figure 9). To address this, the LLM is prompted to merge duplicate descriptions ensuring that each final description list contains one item for each inconsistency.[7]

2-Step. LLMs have demonstrated strong performance in *binary* factual consistency evaluation (Gekhman et al., 2023), suggesting they can be used to improve *fine-grained* evaluation by filtering-out cases which are unlikely to contain errors. Motivated by this, we explore two-step baselines: (Step 1) classify whether the summary is factually consistent; (Step 2) if classified as inconsistent, prompt the evaluated LLM to identify individual errors. For Step 1, we implement three variants: (1) **Self**, where the evaluated LLM performs binary classification using CoT prompting (prompt shown at H.3); (2) **TrueTeacher**, which uses the model from Gekhman et al. (2023);[8] and (3) **Oracle**, which uses the ground-truth label, serving as an upper bound. For Step 2, we use **CoT** prompting, and also implement a **CoT&Hint** variant, modifying the instruction to indicate that inconsistencies are present. Additional technical details are in §C.

4.2 Evaluation

We report error detection precision (the fraction of predictions that are true inconsistencies), recall (the fraction of gold inconsistencies that are predicted), and F1.

[6] We used GPT-4o (Achiam et al., 2023) to generate the initial candidates.

[7] These steps are needed to evaluate error localization since the original implementation of FactScore only assesses each fact and returns a score that represent the fraction of consistent facts, which is not helpful for localizing specific errors.

[8] https://huggingface.co/google/t5_11b_trueteacher_and_anli

		GPT-4o			Claude-sonnet-3.5			Gemini-1.5-pro			Llama-3.1-405B			**Average**		
		Rec.	Prec.	F1	Rec.	Prec.	F1	Rec.	Prec.	F1	Rec.	Prec.	F1	Rec.	Prec.	F1
E2E	Zero-Shot	0.35	0.70	0.47	0.26	**0.78**	0.40	0.49	0.46	0.48	0.53	0.50	0.51	0.41	0.61	0.46
	Few-Shot	**0.56**	0.59	0.57	0.44	0.74	0.56	0.39	0.66	0.49	0.48	0.57	0.52	0.47	0.64	0.54
	CoT	0.51	0.68	**0.59**	**0.67**	0.66	**0.67**	**0.54**	0.62	0.57	**0.54**	0.59	0.56	**0.57**	0.64	**0.60**
Pipeline	FactScore	0.52	0.66	0.58	0.60	0.63	0.62	0.30	0.69	0.42	0.49	0.67	**0.57**	0.48	0.66	0.55
2-Step	Self + CoT	0.41	**0.76**	0.54	0.51	0.76	0.61	0.39	**0.76**	0.52	0.38	**0.76**	0.51	0.42	**0.76**	0.54
	Self + CoT&Hint	0.42	0.74	0.54	0.50	0.77	0.61	0.37	0.70	0.49	0.39	0.74	0.51	0.42	0.74	0.53
	TrueTeacher + CoT	0.47	0.74	0.58	0.62	0.72	**0.67**	0.49	0.70	**0.58**	0.49	0.66	0.56	0.52	0.71	0.60
	TrueTeacher + CoT&Hint	0.49	0.68	0.57	0.61	0.72	0.66	0.49	0.63	0.55	0.49	0.60	0.54	0.52	0.66	0.58
2-Step Oracle	Oracle + CoT	0.51	0.77	0.62	0.67	0.75	0.71	0.54	0.72	0.61	0.54	0.65	0.59	0.57	0.72	0.63
	Oracle + CoT&Hint	0.55	0.70	0.62	0.67	0.75	0.71	0.54	0.64	0.59	0.55	0.58	0.56	0.58	0.67	0.62

Table 4: Performance of different LLMs on the **FINAL** benchmark for **F**actual **I**nconsistencies **L**ocalization. In *E2E*, the LLM performs end-to-end localization under various prompting strategies. In *2-Step*, it localizes inconsistencies only when a preceding classifier flags the summary as inconsistent; in *2-Step Oracle*, this is a perfect, oracle classifier. In *Pipeline*, the LLM first decomposes the summary into atomic facts, which are then evaluated individually. Best non-oracle results per-column are in bold, best overall results are underlined. More details in §4.1.

To measure these metrics, we leverage LLM-as-a-judge as illustrated in Figure 3. To mitigate the potential risk of model biases such as self-preference or length bias (Ye et al., 2025; Wang et al., 2024; Zheng et al., 2023), we do not ask the model for a subjective score or classification. Instead, we frame the evaluation as a *matching task*. The judge receives the summary, the list of gold descriptions, and a list of predicted descriptions, and is prompted to align items from the gold list with those in the predicted list. Matched descriptions are counted as true positives. We use GPT-4o as the judge. To validate the judgment's quality, we conduct extensive human evaluation on our 140-sample development set, generating outputs for each prompt by each model. We assign human annotators with the same matching task as the judge model to create human-annotated matches, to serve as ground truth. We then calculate the *precision* and *recall* of the matching task, which were 0.95 and 0.92, respectively, providing evidence that our approach produces high-quality judgments. Additional implementation details on the judgment process can be found in §D. To ensure that the evaluation is stable, we test the effect of multiple model runs and perform a variance analysis for the LLM judge in Appendix F.

5 Results

Table 4 presents the main results. The cross-model average F1 (rightmost column) remains below 0.60 for all methods (except with the Oracle classifier), highlighting the benchmark's difficulty. Full results, including evaluations of smaller open-weight models, are available in Appendix E. Smaller models performed notably worse than the primary models discussed here. In the E2E setup, CoT shows

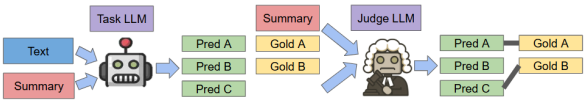

Figure 3: Illustration of our LLM-based evaluation protocol. The evaluated LLM detected three errors while the annotation listed two. The judge matches between the predicted and ground truth inconsistencies.

superior performance, suggesting that reasoning is helpful for localizing factual inconsistencies. Interestingly, CoT surpasses FactScore (average F1 of 0.60 vs. 0.55), suggesting that allowing the model to reason freely is more effective than controlling its reasoning process through predefined steps. Another notable trend is that precision consistently exceeds recall, suggesting that LLMs tend to focus on a subset of errors for which they have sufficient confidence.

We next analyze the effectiveness of the preliminary filtering step in the 2-Step setting. *Oracle+CoT* filters error-free examples, thereby improving precision and outperforming *CoT*. Conversely, *Self+CoT* underperforms *CoT*, while *TrueTeacher+CoT* only matches *CoT*'s performance. These results highlight the potential of perfect filtering to improve performance, but also suggest that with current classification quality, end-to-end approaches remain more effective.

The fact that *Self+CoT*, where the same LLM first performs a filtering step, underperforms *CoT* is rather surprising, since LLMs are expected to perform well in the binary classification task. To further understand this gap, we focus on the *binary* factual consistency evaluation task and compare each LLM's performance against a *Binarized* baseline, where the LLM is prompted to perform *fine-grained* evaluation followed by post-processing

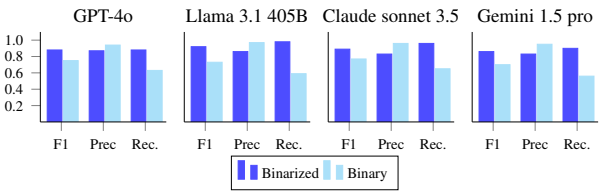

Figure 4: Performance on *binary* factual consistency evaluation. In "Binary" the LLM is prompted with the binary task, while in "Binarized" it is prompted for fine-grained evaluation and its outputs are post-processed into binary labels.

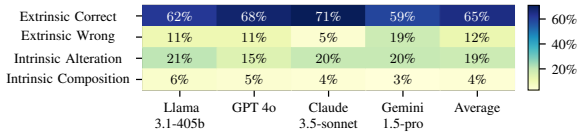

Figure 5: Distribution of false negatives (Table 5).

that labels a summary as inconsistent if at least one error is detected.[9] The results are presented in Figure 4. *Binarized* achieves higher F1, despite *Binary* assigning the model a seemingly easier task. One possible explanation is that in *Binarized*, the model must identify all inconsistencies, which may encourage a more thorough analysis of the content. Interestingly, *Binary* consistently yields higher precision but lower recall compared to *Binarized*. This suggests that the model is more conservative in the *Binary* setup, avoiding false positives but failing to detect many actual inconsistencies.

Lastly, we examine the effect of explicitly informing the model that the summary contains errors by comparing *Oracle+CoT* to *Oracle+CoT&Hint*. Since these variants share the same filtering step, they allow us to directly compare *CoT* to *CoT&Hint* on the same examples. As expected, *CoT&Hint* achieves higher recall than *CoT*, as the hint encourages the model to identify more errors. However, this comes at a significant cost to precision, suggesting the model becomes overly permissive and flags many false positives, ultimately reducing overall F1.

6 Error analysis

This section presents an error analysis to better understand the reasons models make mistakes. We divide it into 2 parts: (1) **false negatives analysis** for why models do not detect some inconsistencies, and (2) **false positives analysis** for why some of the models' predictions are incorrect.

False Negatives. We manually analyzed a random sample of 150 undetected inconsistencies from each model and categorized them into four categories.[10] We present their definitions and examples in Table 5, and their distribution in Figure 5. The most common category shared across all models is *Extrinsic Correct*. Since the errors in this category involve factually correct information, we hypothesize that a key reason for these failures is that the information aligns with the model's parametric knowledge, making it difficult for the model to recognize it as inconsistent, even when it is not supported by the source text.[11] To substantiate this hypothesis in our setup, we analyze (1) whether the information in question indeed aligns with the model's knowledge, and (2) whether this alignment is the reason the model fails to identify these inconsistencies.

To provide evidence that the *Extrinsic Correct* errors contain information that is mostly known to the model we need a method to assess LLMs' knowledge, which is not trivial (Fierro et al., 2024; Gekhman et al., 2024, 2025). We choose to use $\mathbf{P}(\mathbf{True})$ (Kadavath et al., 2022), a popular metric that quantifies the likelihood that the model assigns for the correctness of a specific answer to a question. For each inconsistency, we ask human annotators to generate a question $\mathbf{q}$ for which the answer $\mathbf{a}$ is the factually inconsistent information from the summary.[12] We then calculate $\mathbf{P}(\mathbf{True} \mid \mathbf{q}, \mathbf{a})$, as an estimate to whether the factually inconsistent information aligns with the model's knowledge. Figure 6 presents a density plot of $\mathbf{P}(\mathbf{True})$ scores. For the *Extrinsic Correct* category, the distribution is concentrated near 1, indicating that it contains facts that are largely known to the model.

After establishing that the model often possesses knowledge about the correct information added in *Extrinsic Correct* cases, we provide evidence that this may explain its failure to detect such errors. We ask human annotators to generate counterfactual versions for *Extrinsic Correct* inconsistencies

[9] We use the CoT fine-grained variant in *Binarized* and prompt the model to reason step-by-step in *Binary*.

[10] We build upon the established intrinsic/extrinsic taxonomy (Maynez et al., 2020) and expand it to four categories to capture the errors we observed.

[11] We note that prior work already discussed LLMs' tendency to prioritize their own parametric knowledge over the provided context (Ming et al., 2025; Xie et al., 2023). Our main contribution is in providing empirical evidence that this phenomena is the key driver of failures specifically in the task of error localization.

[12] Technical details are in §G.1 and an example is presented in Figure 11. We ran this analysis for GPT-4o.

Category	Definition	Summary	Text Evidence	Explanation
Extrinsic Correct	The inconsistency is additional information in the summary that is not in the source text, and is factually correct.	Japan's Hayabusa2 spacecraft landed on Ryugu, collected samples, and has returned them to Earth for solar system research.	A Japanese spacecraft successfully landed on an asteroid...	The name of the asteroid "Ryugu" is correct, but does not appear in the text
Extrinsic Wrong	The inconsistency is additional information in the summary that is not in the source text and is factually incorrect.	Japan's Hayabusa2 spacecraft landed on Bennu, collected samples, and has returned them to Earth for solar system research.	A Japanese spacecraft successfully landed on an asteroid...	The name of the asteroid "Bennu" is wrong, and it does not appear in the text
Intrinsic Alteration	The inconsistency is based on information from the source text that has been altered in the summary.	Japan's Hayabusa2 spacecraft landed on an asteroid, collected samples, and will return them to Earth for solar system research.	...and later returned the samples to Earth...	The summary claims it "will return", but according to the text it returned.
Intrinsic Composition	The inconsistency is the result of individually correct facts from the source text being combined poorly.	Japan's Hayabusa2 spacecraft landed on an asteroid, collected samples, and has returned them to Earth for groundbreaking solar system research.	...collected samples in a groundbreaking mission...to learn more about the origins of the solar system...	The summary calls the research "groundbreaking," but that's the description of the mission.

Table 5: Definitions and examples of false negatives, cases where the model failed to detect factual inconsistencies, accompanied by an explanation to why each example was classified to that category.

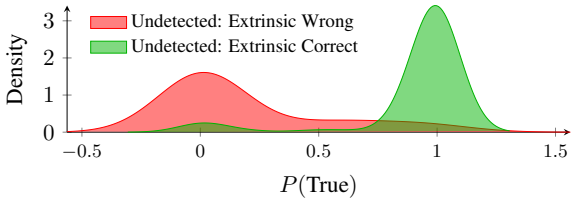

Figure 6: Density of $\mathbf{P(True)}$ scores for the *Extrinsic Correct* and *Extrinsic Wrong* false negatives (Table 5).

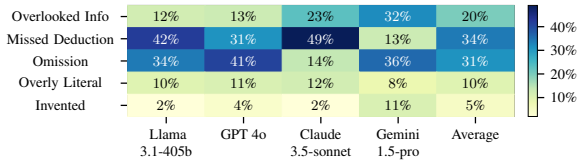

Figure 7: Distribution of False Positive categories.

by replacing the (correct) added information with semantically similar but incorrect alternatives. For example, if a summary says *"the protests were in London"*, and *"London"* is *not* in the source text but is correct, we might replace it with a different UK city. We found that 88.1% of these counterfactual errors were successfully detected. This result, together with the P(True) analysis, strongly suggests that the alignment with the model's parametric knowledge causes it to miss that the added information is unsupported by the source text.

False Positives. We manually analyzed a random sample of 100 false positives per-model: predictions that do not reflect real factual inconsistencies. We have identified 5 main categories, with an example of each presented in Table 11 in the Appendix.

- **Overlooked Info.** Failure to recognize information that is explicitly stated in the source text, leading to wrong prediction.
- **Missed Deduction.** Failure to recognize a fact that can be directly deduced from the text.
- **Omission.** Classifying information that is missing in the summary as an inconsistency.
- **Overly literal.** Classifying superficial changes in wording as an inconsistency.
- **Invented.** The information that is mentioned as inconsistent is not in the text or summary.

As shown in Figure 7, the most prevalent categories were *Missed Deduction* and *Omission*. The former is expected, as identifying information that is not explicitly stated in the text is challenging. The latter is rather surprising, as it suggests limited instruction following capabilities. The prompt clearly instructs to identify facts *in the summary* that cannot be verified, yet the model flags information that is simply omitted from the summary.

7 Related Work

Previous work on factual consistency evaluation can be categorized along two axes: (1) the setting is either *binary*, assigning a single score to the entire output, or *fine-grained*, localizing specific errors; and (2) the goal is either *evaluation*, focusing on building consistency-checking systems, or *meta-evaluation*, focusing on evaluation of such systems.

Most work in the *fine-grained* setting focuses on *evaluation* and not *meta-evaluation*. In these studies, factual inconsistencies are represented as entities (Cao et al., 2022), sentences (Laban et al., 2022), spans (Maynez et al., 2020; Cao and Wang, 2021; Niu et al., 2024; Mishra et al., 2024), atomic facts (Min et al., 2023; Chen et al., 2023; Kamoi et al., 2023b) or QA pairs (Honovich et al., 2021; Wang et al., 2020; Fabbri et al., 2022; Cattan et al., 2024). As we discuss in §2, the expressivity of these representations is limited, constraining the range of errors that can be captured, and they can often be vague, which complicates evaluation.

Perhaps owing to the evaluation challenges stemming from existing error representations, most work on *meta-evaluation* of factual consistency focus on the *binary* setting (Honovich et al., 2022;

Laban et al., 2022; Gekhman et al., 2023; Clark et al., 2023; Tang et al., 2023; Luo et al., 2023). To our knowledge, the only work that performed meta-evaluation in the fine-grained setting is Niu et al. (2024). They represent errors using spans and measure character-level overlap, a limitation we discuss in detail in §2. In addition, their evaluation focused on GPT-4-turbo (Achiam et al., 2023), GPT-3.5-turbo (OpenAI, 2022), and Llama-2-13B (Touvron et al., 2023), leaving the performance of the highest-capacity models as an open question.

Our work addresses the limitations of existing error representations by proposing a new one based on textual descriptions. Not only does it allow us to capture the full range of possible errors, but when combined with our LLM-based evaluation protocol, it also helps us overcome the evaluation challenge of comparing predicted factual inconsistencies to ground truth ones. This framework facilitates the construction of **FINAL**, a high-quality benchmark for the meta-evaluation of LLMs on the task. In addition, our study evaluates extremely high-capacity LLMs, providing a fresh perspective on their capabilities at a scale not previously studied for this task. Finally, our detailed error analysis sheds light on the reasons for their failures.

8 Conclusion

We take a step towards replacing existing fine-grained factual consistency evaluation systems with LLMs. We introduce **FINAL** - the first benchmark for **F**actual **I**nconsistencies **L**ocalization using LLMs, with 1,400 carefully annotated examples. We evaluate four strong LLMs, with a detailed analysis that offers a clear view of their strengths and weaknesses. We hope that our benchmark and insights will foster a shift towards LLM-based evaluation, which will support broader adoption of fine-grained consistency evaluation in practical, real-world applications.

9 Limitations

Exclusive Focus on LLMs. A key limitation of our work is that the proposed benchmark and evaluation protocol are specifically tailored for LLMs. We introduce an error representation based on free-form textual descriptions, which aligns naturally with the capabilities of LLMs and, when paired with our LLM-based evaluation protocol, resolves evaluation challenges caused by the vagueness of prior representations. However, this design choice makes it difficult to use our framework to directly compare LLMs with traditional, non-LLM systems that rely on different output formats, such as entities, spans, or QA pairs.

We made a considerable effort to bridge this gap with our **Pipeline** baseline (see §4.1), which emulates a traditional evaluation method by (Min et al., 2023). However, as can be seen in **Pipeline**'s implementation details, adapting Min et al. method for a direct comparison proved to be a non-trivial task, requiring substantial modifications to make its output compatible with our description-based framework, highlighting the inherent difficulty of such cross-paradigm comparisons.

In this context, we would like to highlight that the choice of error representation typically fundamentally impacts not only the system design but also the ability to annotate gold labels and create a standardized benchmark, since comparing systems with disparate output formats is a significant challenge. Thus, our work deliberately focuses on a representation suitable for LLMs to establish a high-quality benchmark for their meta-evaluation, acknowledging that this specialization limits its applicability for evaluating systems with different architectures.

Benchmark Annotation Challenges and Coverage. Creating a high-quality, comprehensive benchmark for fine-grained hallucination detection is an extremely challenging task. The difficulty of achieving complete annotation coverage is likely inherent to any benchmark for such a complex and subjective task. This was evident in the original DeFacto dataset, where we found numerous issues, including vague or incorrect annotations and a considerable number of factual inconsistencies entirely missing from the dataset. To ensure the reliability of our **FINAL** benchmark, we took several quality control steps, such as relying on expert annotators and implementing a rigorous, two-phase annotation process that included an LLM-human collaboration to enrich the data, which increased the amount of discovered inconsistencies by 31% (see §3).

Despite these extensive measures, we acknowledge that some inconsistencies may still be unannotated. While this is a common challenge for such datasets, we believe in being transparent about it.

Generalizability and Context Length. We build on the DeFacto dataset (Liu et al., 2023), which uses relatively short source documents and generated summaries compared to long form gener-

ation datasets like FActScore (Min et al., 2023) and RAGTruth (Niu et al., 2024). The task is also restricted to the summarization domain. While it remains an open question how well a model that excels on this benchmark would generalize to longer and more realistic scenarios, this design choice represents a deliberate tradeoff. We prioritized annotation quality over document length or task diversity. Creating precise and free form natural language descriptions of fine grained errors requires rigorous expert annotation. Achieving this standard on much longer documents without initial reference explanations provided by DeFacto would be prone to noise and errors. Adapting this rigorous evaluation paradigm to longer texts and diverse tasks remains an important direction for future work. Importantly, our evaluation shows that state of the art models still struggle significantly on FINAL. The best model achieves an F1 score of only 0.67 (table 4). We believe solving factual inconsistency detection in these shorter contexts is a necessary prerequisite for tackling longer and more complex scenarios.

References

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

Anthropic. 2024. Claude ai models. https://docs.anthropic.com/en/docs/welcome.

Ge Bai, Jie Liu, Xingyuan Bu, Yancheng He, Jiaheng Liu, Zhanhui Zhou, Zhuoran Lin, Wenbo Su, Tiezheng Ge, Bo Zheng, et al. 2024. Mt-bench-101: A fine-grained benchmark for evaluating large language models in multi-turn dialogues. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7421–7454.

Bernd Bohnet, Vinh Q Tran, Pat Verga, Roee Aharoni, Daniel Andor, Livio Baldini Soares, Massimiliano Ciaramita, Jacob Eisenstein, Kuzman Ganchev, Jonathan Herzig, et al. 2022. Attributed question answering: Evaluation and modeling for attributed large language models. *arXiv preprint arXiv:2212.08037*.

Meng Cao, Yue Dong, and Jackie Chi Kit Cheung. 2022. Hallucinated but factual! inspecting the factuality of hallucinations in abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3340–3354.

Shuyang Cao and Lu Wang. 2021. Cliff: Contrastive learning for improving faithfulness and factuality in abstractive summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6633–6649.

Arie Cattan, Paul Roit, Shiyue Zhang, David Wan, Roee Aharoni, Idan Szpektor, Mohit Bansal, and Ido Dagan. 2024. Localizing factual inconsistencies in attributable text generation. *arXiv preprint arXiv:2410.07473*.

Mert Cemri, Melissa Z Pan, Shuyi Yang, Lakshya A Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, et al. 2025. Why do multi-agent llm systems fail? In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Sihao Chen, Senaka Buthpitiya, Alex Fabrikant, Dan Roth, and Tal Schuster. 2023. Propsegment: A large-scale corpus for proposition-level segmentation and entailment recognition. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8874–8893.

Elizabeth Clark, Shruti Rijhwani, Sebastian Gehrmann, Joshua Maynez, Roee Aharoni, Vitaly Nikolaev, Thibault Sellam, Aditya Siddhant, Dipanjan Das, and Ankur Parikh. 2023. Seahorse: A multilingual, multifaceted dataset for summarization evaluation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9397–9413.

Alexander Richard Fabbri, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2022. Qafacteval: Improved qa-based factual consistency evaluation for summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2587–2601.

Constanza Fierro, Ruchira Dhar, Filippos Stamatiou, Nicolas Garneau, and Anders Søgaard. 2024. Defining knowledge: Bridging epistemology and large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16096–16111, Miami, Florida, USA. Association for Computational Linguistics.

Zorik Gekhman, Eyal Ben David, Hadas Orgad, Eran Ofek, Yonatan Belinkov, Idan Szpektor, Jonathan Herzig, and Roi Reichart. 2025. Inside-out: Hidden factual knowledge in llms. *arXiv e-prints*, pages arXiv–2503.

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

Zorik Gekhman, Gal Yona, Roee Aharoni, Matan Eyal, Amir Feder, Roi Reichart, and Jonathan Herzig. 2024. Does fine-tuning LLMs on new knowledge encourage hallucinations? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7765–7784, Miami, Florida, USA. Association for Computational Linguistics.

Akshay Goel, Almog Gueta, Omry Gilon, Chang Liu, Sofia Erell, Lan Huong Nguyen, Xiaohong Hao, Bolous Jaber, Shashir Reddy, Rupesh Kartha, et al. 2023. Llms accelerate annotation for medical information extraction. In *Machine Learning for Health (ML4H)*, pages 82–100. PMLR.

Google. 2024. Gemini pro. https://ai.google.dev.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Horace He and Thinking Machines Lab. 2025. Defeating nondeterminism in llm inference. *Thinking Machines Lab: Connectionism*. Https://thinkingmachines.ai/blog/defeating-nondeterminism-in-llm-inference/.

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

Or Honovich, Leshem Choshen, Roee Aharoni, Ella Neeman, Idan Szpektor, and Omri Abend. 2021. Q2: Evaluating factual consistency in knowledge-grounded dialogues via question generation and question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7856–7870.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. 2022. Language models (mostly) know what they know. *CoRR*.

Ryo Kamoi, Tanya Goyal, and Greg Durrett. 2023a. Shortcomings of question answering based factuality frameworks for error localization. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 132–146.

Ryo Kamoi, Tanya Goyal, Juan Diego Rodriguez, and Greg Durrett. 2023b. Wice: Real-world entailment for claims in wikipedia. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7561–7583.

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

Yuho Lee, Taewon Yun, Jason Cai, Hang Su, and Hwanjun Song. 2024. Unisumeval: Towards unified, fine-grained, multi-dimensional summarization evaluation for llms. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3941–3960.

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

Junyi Li, Jie Chen, Ruiyang Ren, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2024. The dawn after the dark: An empirical study on factuality hallucination in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10879–10899.

Yixin Liu, Budhaditya Deb, Milagro Teruel, Aaron Halfaker, Dragomir Radev, and Ahmed Hassan. 2023. On improving summarization factual consistency from natural language feedback. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15144–15161.

Zheheng Luo, Qianqian Xie, and Sophia Ananiadou. 2023. Chatgpt as a factual inconsistency evaluator for text summarization. *arXiv preprint arXiv:2303.15621*.

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

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. *arXiv preprint arXiv:2305.14251*.

Yifei Ming, Senthil Purushwalkam, Shrey Pandit, Zixuan Ke, Xuan-Phi Nguyen, Caiming Xiong, and Shafiq Joty. 2025. Faitheval: Can your language model stay faithful to context, even if" the moon is made of marshmallows". In *The Thirteenth International Conference on Learning Representations*.

Abhika Mishra, Akari Asai, Vidhisha Balachandran, Yizhong Wang, Graham Neubig, Yulia Tsvetkov, and Hannaneh Hajishirzi. 2024. Fine-grained hallucination detection and editing for language models. *arXiv preprint arXiv:2401.06855*.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

Cheng Niu, Yuanhao Wu, Juno Zhu, Siliang Xu, Kashun Shum, Randy Zhong, Juntong Song, and Tong Zhang. 2024. Ragtruth: A hallucination corpus for developing trustworthy retrieval-augmented language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10862–10878.

OpenAI. 2022. Chatgpt. https://openai.com/blog/chatgpt. Model: gpt-3.5-turbo.

Liyan Tang, Tanya Goyal, Alex Fabbri, Philippe Laban, Jiacheng Xu, Semih Yavuz, Wojciech Kryściński, Justin Rousseau, and Greg Durrett. 2023. Understanding factual errors in summarization: Errors, summarizers, datasets, error detectors. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. 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.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, et al. 2024. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9440–9450.

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.

Sarah Wiegreffe, Jack Hessel, Swabha Swayamdipta, Mark Riedl, and Yejin Choi. 2022. Reframing human-ai collaboration for generating free-text explanations. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 632–658.

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. 2023. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. In *The Twelfth International Conference on Learning Representations*.

Jiayi Ye, Yanbo Wang, Yue Huang, Dongping Chen, Qihui Zhang, Nuno Moniz, Tian Gao, Werner Geyer, Chao Huang, Pin-Yu Chen, et al. 2025. Justice or prejudice? quantifying biases in llm-as-a-judge. In *International Conference on Learning Representations*.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

A Representation

As discussed in Section 2, current methods of representation, such as entity or span marking, are often inadequate. To understand the magnitude of this issue, we conducted a manual analysis on the development set of our dataset. We analyzed 140 text-summary pairs, out of which 112 pairs have a total of 214 factual inconsistencies. Our objective was to evaluate whether these inconsistencies can be effectively marked using spans or entities.

Statistic	Inconsistencies		Summaries	
	C	P (%)	C	P (%)
Entity Marking Validity				
Inconsistency not an entity	114	53.27	62	55.36
Span Marking Validity				
Unsuited for span	35	16.36	32	28.57
Subjective span	30	14.02	28	25.00
Impossible to mark	5	2.34	5	4.46

Table 6: Distribution of inconsistencies by representation validity. **C** denotes the Count (out of a total of 112 text-summary pairs and 214 inconsistencies), and **P** denotes the Percentage.

We performed two analyses on the data:

- **Span Marking Validity** - We assessed how well inconsistencies can be marked with spans, classifying them as: (a) *Impossible to mark* (the inconsistency cannot be represented by a span), (b) *Subjective marking* (multiple spans can represent the same inconsistency, or one marked span can be interpreted as various mistakes), and (c) *Possible to mark*.

- **Entity Marking Validity** - We classified whether the inconsistency occurs within an entity in the summary or not.

As shown in Table 6, most inconsistencies are not contained within entities and instead involve elements like verbs or adjectives. Regarding span marking, while only a small number of inconsistencies are entirely impossible to represent, the primary challenge is the high rate of ambiguity.

B Dataset

B.1 DeFacto Curation

As noted in §3, our benchmark builds on the DeFacto Benchmark (Liu et al., 2023), which studied how human feedback improves models at correcting factual inconsistencies. DeFacto paired texts with summaries and, when inconsistencies were found, provided feedback to correct the summary. This feedback included three main components: (1) an **explanation** of the inconsistencies in the summary, (2) **instructions** of how to correct the summary and (3) a **revised summary** fixed by a human. Ideally, we could simply use the explanations provided in the dataset, but a considerable portion of the original explanations were lacking. Beyond the fact that numerous inconsistencies were missing (more on that in §B.3), the provided explanations were often not usable as is: some inconsistencies were merged together, some explanations were vague or incomplete, and others contained irrelevant information. Therefore, we extracted individual inconsistencies descriptions from DeFacto through manual annotation. Participants were instructed to review the annotations provided in the original DeFacto dataset and extract descriptions of factual inconsistencies. Then they were asked to rephrase them as individual natural language statements explicitly identifying the inconsistency in the summary, with minimal changes to the original dataset.

Below are the annotation instructions:

Task Overview

You will be provided with the following for each sample:

- Text: The source document.
- Summary: A factually inconsistent summary of the text.
- Raw Human Annotation, which includes:
  - A written explanation of the inconsistencies in the summary.
  - Instructions for fixing the summary.
  - An edited summary that is factually consistent.

Your task is to extract individual **factual inconsistency descriptions**—short, self-contained statements that identify exactly what information in the summary is factually inconsistent with the source text.

Instructions

- **Extract Individual Inconsistencies**
  - For each factual inconsistency described in the explanation, extract and formulate it as a complete, standalone sentence that clearly identifies the inconsistency on its own, without referring to other inconsistencies.
  - Each description should explicitly state what is incorrect in the summary.
- **Minimize Changes**
  - Make minimal edits to the original wording of the explanation.
  - Modify the explanation only when necessary:
    - To phrase it as a standalone sentence.
    - If the inconsistency is vague or unclear → rephrase it to be precise and unambiguous.
    - The explanation includes irrelevant or redundant information → remove any parts that do not describe factual inconsistencies (e.g., mentions of consistent facts, mentions of information that is omitted from the summary, context and so on).
- **Add Missing Inconsistencies**
  If an inconsistency is evident in the edited summary or instructions but is not mentioned in the explanation, write a new description for it.
- **Remove Invalid Annotations**
  If you notice that a described inconsistency is not actually inconsistent—for example, the information is not present in the summary, or the summary is consistent with the text—discard it.
- **Remove Low-Quality Samples**
  If the entire summary is factually inconsistent (e.g., instructions call for a complete rewrite), mark the sample as "discarded: full rewrite required" and do not extract any descriptions.

B.2 Curation Analysis

After converting the original DeFacto dataset explanations into our description based format, as outlined in §3 and §B.1, in order to understand the extent of changes to the original dataset, we performed the following analysis: We sampled 400 text-summary pairs from the DeFacto dataset, all originally labeled as factually inconsistent. These samples contained a total of 655 identified inconsistencies. We manually analyzed the required operations to adapt the original DeFacto explanations

into our format. We categorized the operations into two main types, further divided into eight subcategories:

- **Simple Changes** - These included cases where the explanation required no modification at all, or where inconsistencies were directly extracted from the explanation.
- **Challenging Changes** - These involved more significant intervention, including decomposition of merged inconsistencies, removal of irrelevant information, clarification of vague explanations, inferring missing inconsistencies not originally annotated, identifying incorrect annotations, and discarding summaries that were almost entirely unrelated to the source text.

Table 2 provides a detailed illustration of the various operations, accompanied by examples. Out of the 655 inconsistencies, the distribution of required operations was as follows:

- **Simple Changes: 480 (73.3%)**
  - No change needed: 161 (24.6%)
  - Extraction: 319 (48.7%)
- **Challenging Changes: 175 (26.7%)**
  - Decomposition: 24 (3.6%)
  - Clarifying vague explanation: 53 (8.1%)
  - Imputing missing explanation: 17 (2.6%)
  - Removal of irrelevant information: 37 (5.6%)
  - Removal of incorrect annotation: 21 (3.2%)
  - Removal of unrelated summary: 23 (3.5%)

B.3 Human-LLM Collaboration

During the process of converting DeFacto explanations into descriptions (as detailed in §3 and §B.1), and through initial experiments using an LLM as an annotator, it became clear that many inconsistencies were not present in the original dataset. At the same time, initial experiments revealed that a significant number of inconsistencies surfaced by LLMs were valid, even though they did not appear in the original dataset. Since the DeFacto dataset was annotated by humans, we hypothesized that it was less likely that further human annotation would uncover many new inconsistencies, and using LLMs blindly would have introduced a lot of false inconsistencies into our dataset. Therefore, we decided to use a collaborative approach, leveraging the LLM's ability to surface new, unannotated inconsistencies with human judgment. The approach was based on high recall prompting and human filtering.

High Recall Prompting and Human Filtering. An LLM is prompted to identify as many potential inconsistencies as possible, explicitly prioritizing high recall over precision. We used GPT-4o for this purpose and provide the full prompt in §H.1. Human annotators then reviewed the LLM's outputs to filter out false positives. This process yielded additional true inconsistencies that were originally missed by human annotators. Using this method, we managed to increase the amount of annotated inconsistencies in the data from a total of 1627 to 2131 (+31%) and also discovered inconsistencies in 128 previously thought to be consistent summaries (full details can be seen in §3).

Inter Annotation Agreement. To validate the quality of the human annotation in our Human-LLM collaboration, we randomly sampled 150 instances and had two annotators independently review the LLM's outputs generated with the high-recall prompt. Each inconsistency predicted by the LLM was to with one of three categories: (1) an inconsistency present in the original data; (2) an inconsistency absent from the original data; or (3) not an inconsistency. To compute inter annotator agreement, we mapped categories (1) and (2) to True and category (3) to False, then computed inter-annotator agreement. We observed a raw agreement = 0.877 and Cohen's κ = 0.731, indicating substantial agreement and supporting the validity of our augmentation.

Filtering samples. As stated in §3, during the Human-LLM collaboration process the annotators encountered LLM predictions that they could not definitively confirm as correct or incorrect. To maintain dataset quality, we filtered out these ambiguous cases entirely. These instances involved cases like ambiguity, subjectivity, etc. that made it impossible to confidently verify the model's predictions.

For example: the LLM flagged that a summary claimed Chris Hadfield *"left"* the ISS while the source text only mentioned *"the time came for his departure"*. In this case the phrasing in the source text is ambiguous since it could mean that the departure time had arrived but he had not left yet, or that the time had already passed and he had departed. In the first case, the inconsistency flagged by the model is correct, but in the second it is not.

This prevents us from definitively classifying the model's prediction. Another example is when the model identified an inconsistency in the summary's description of an accident as *"serious"* versus the text stating the injured *"required hospitalization"* the subjective definition of "serious" prevented a clear determination.

B.4 Dataset Statistics

As mentioned in §3, the final dataset comprises 1405 text-summary pairs with 2131 annotated factual inconsistencies. Figure 8 presents the distribution of the number of inconsistencies per summary. Notably, roughly 45% of summaries contain more than one inconsistency, highlighting the need for fine-grained annotation rather than a binary consistent/inconsistent label.

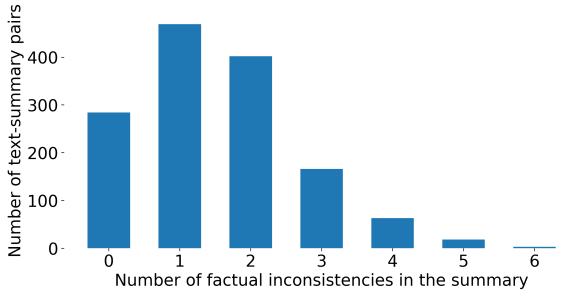

Figure 8: Number of factual inconsistencies in each summary in the final dataset.

C Baselines

In §4.1, we introduced the three experimental setups used to evaluate LLMs for detecting factual inconsistencies. These were: (1) **E2E** - a single stage, end to end approach; (2) **Pipeline** - a multi-stage method in which we adapt FactScore (Min et al., 2023) to our setup and (3) **2-Step** - a two stage setup, in which the summaries are first filtered by a classifier, and then the model is prompted to detect individual errors. For all setups, we set the temperature to 0 for the evaluated models to minimize variance in outputs. Below we provide additional technical details about each setup.

C.1 E2E

In this setup we explored three prompting styles: **Zero-shot, Few-shot** and Chain-of-thought (**CoT**). Since prompt phrasing can significantly impact LLM performance, we created two variants for each prompting style. For each model, we then selected the variant that achieved the highest F1 score on the development set, and used these selected prompts for test-time evaluation. All prompts variants are available in §H.2.1 (**Zero Shot**), §H.2.2 (**Few Shot**) and §H.2.3 (**CoT**).

C.2 Pipeline

We adapted our Pipeline setup from FactScore (Min et al., 2023). FactScore consists of two stages: (1) **decomposition** - breaking down each summary into atomic facts; and (2) **classification** - assigning each atomic fact a label of consistent or inconsistent. In our pipeline, the decomposition stage remained unchanged, using exactly the same prompt as in the original work. We deviate from the original paper solely in the classification stage. The original method produces a single continuous score, representing the aggregate fraction of facts classified as correct. However, our objective is not system evaluation but meta-evaluation: comparing the FactScore framework against other evaluation methodologies. This necessitated an output that allows for a direct matching between identified errors and our ground-truth annotations. Building on the original classification prompt from FactScore, the task was changed so that instead of a single label (consistent/inconsistent), the model outputs a short description of each inconsistency it finds. As mentioned in §2, an atomic fact may contain multiple inconsistencies, therefore we require the model to output a list of all the inconsistencies found in the atomic fact. However, as illustrated in Figure 9, information expressed in atomic facts is not mutually exclusive, and the same content may appear in multiple facts. To remove those duplications, an additional **deduplication** stage was introduced. This stage consolidates the inconsistencies extracted from all atomic facts into a single unified list, ensuring that each unique inconsistency is represented only once. Full prompts are available at §H.2.5

C.3 2-Step

In the 2-Step setup, we used the CoT variant selected for each model in the E2E experiments. We evaluated three classifiers: (1) **self**, where the model itself served as a binary classifier with a chain-of-thought prompt (full prompt at §H.3) (2) **TrueTeacher**, a classifier trained for binary factual consistency detection (Gekhman et al., 2023); and (3) **Oracle**, where ground truth was used to filter summaries. The threshold for TrueTeacher was the one that maximized the F1 score on the binary summary classification task using the devel-

Summary
The government presented new measures to stabilize prices.

Decomposition

1. The government made a presentation.
2. The government introduced new measures.
3. The measures aim to stabilize prices.

Inconsistency
The central bank, not the government, introduced the measures.

Issue
The inconsistency appears in (1) and (2), and causes duplicate detection.

Figure 9: Example of decomposition to atomic facts causing duplication: when each fact in a summary is checked separately, the same inconsistency may be flagged multiple times, so de-duplication is needed.

opment set. At this threshold, the classifier yields a precision of 0.92, a recall of 0.84, and an F1 score of 0.88. As seen in §4.1, an additional variant was implemented named **CoT&Hint**, in which the model is explicitly instructed that the summary contains inconsistencies. For each model, this variant was created by taking its selected CoT prompt and modify it by adding a statement at the beginning indicating that the summary is inconsistent, and removing the instruction to return "None" in case no inconsistencies were found. Full **CoT&Hint** prompts are available at §H.2.4.

D Evaluation

As explained in §4.2 and illustrated in Figure 3, we evaluate models by matching their predicted inconsistencies to ground-truth ones. This section provides additional technical details on how evaluation was conducted and explains how the evaluation pipeline was validated to ensure reliable judgments. To minimize variance in the process, we set the judge model's temperature to 0 across all evaluation runs.

Automatic evaluation using matching. The judge model receives the summary, the predicted inconsistencies and the ground truth inconsistencies, and outputs a matching between each predicted inconsistency to a ground truth one in the form of a dictionary, as seen in Figure 10. A match occurs when both refer to the same inconsistent information in the summary, regardless of wording or reasoning. Each predicted inconsistency can either be matched to one ground truth inconsistency, or not matched and receive *"None"*. Full matching prompt is available in §H.4. Ideally, each prediction matches a different ground-truth inconsistency. However, models may repeat the same issue, either unintentionally or even to boost precision scores by repeating high-confidence predictions. This can result in multiple predictions matching the same ground-truth item. Ideally, we would use de-duplication to remove doubles, but we found it to be less reliable for free-text outputs. Thus, instead of counting the number of matches, we count the number of ground truth inconsistencies which were matched. This means that if a model repeats the same inconsistency multiple times, all those predictions are matched to the same ground truth and are only counted as one correct detection. This ensures that the model is rewarded correctly for detecting the inconsistency, but repeated predictions of the same inconsistency only inflate the total number of predictions without increasing the number of True Positives. In practice it means that we penalize repeated inconsistencies when we calculate the precision.

Evaluation of automatic vs ground truth matching. To validate our judge model, we compared its performance against that of a human judge. Human annotators were given the full set of predictions generated by the model by a specific prompt on the development set, and tasked them with performing the same matching task as the judge model, using identical instructions. The annotators completed this task across the outputs of four models (Gemini 1.5 Pro, Claude Sonnet 3.5, Llama 3.1 405B, and GPT-4o) and across all prompt types (E2E Zero-shot, E2E Few-shot, E2E CoT, and FactScore). We then compared the annotators' matches to those of the judge model and calculated recall and precision for the task. Averaging across all configurations, we obtained an average precision of 0.95 and an average recall of 0.92.

E Additional models results

In section 5, we saw the results of high capacity llms. To provide a more comprehensive analysis, we extend our evaluation to smaller, open-source LLMs.

As shown in Table 7, the smaller models perform notably worse than the original high capacity models. Furthermore, CoT prompting proves less

		GPT-4o			Claude-sonnet-3.5			Gemini-1.5-pro			Llama-3.1-405B			Ministral-3-14B			Gemma-3-12B		
		Rec.	Prec.	F1	Rec.	Prec.	F1	Rec.	Prec.	F1	Rec.	Prec.	F1	Rec.	Prec.	F1	Rec.	Prec.	F1
E2E	Zero-Shot	0.35	0.70	0.47	0.26	0.78	0.40	0.49	0.46	0.48	0.53	0.50	0.51	0.36	0.54	0.43	0.42	0.45	0.44
	Few-Shot	0.56	0.59	0.57	0.44	0.74	0.56	0.39	0.66	0.49	0.48	0.57	0.52	0.49	0.37	0.42	0.48	0.50	0.49
	CoT	0.51	0.68	0.59	0.67	0.66	0.67	0.54	0.62	0.57	0.54	0.59	0.56	0.43	0.50	0.46	0.41	0.54	0.46
Pipeline	FactScore	0.52	0.66	0.58	0.60	0.63	0.62	0.30	0.69	0.42	0.49	0.67	0.57	0.68	0.30	0.41	0.32	0.69	0.43
2-Step	Self + CoT	0.41	0.76	0.54	0.51	0.76	0.61	0.39	0.76	0.52	0.38	0.76	0.51	0.30	0.69	0.41	0.21	0.70	0.32
	Self + CoT&Hint	0.42	0.74	0.54	0.50	0.77	0.61	0.37	0.70	0.49	0.39	0.74	0.51	0.29	0.67	0.41	0.19	0.60	0.29
	TrueTeacher + CoT	0.47	0.74	0.58	0.62	0.72	0.67	0.49	0.70	0.58	0.49	0.66	0.56	0.39	0.59	0.47	0.37	0.62	0.46
	TrueTeacher + CoT&Hint	0.49	0.68	0.57	0.61	0.72	0.66	0.49	0.63	0.55	0.49	0.60	0.54	0.40	0.52	0.45	0.35	0.45	0.40
Oracle	Oracle + CoT	0.51	0.77	0.62	0.67	0.75	0.71	0.54	0.72	0.61	0.54	0.65	0.59	0.43	0.60	0.50	0.41	0.63	0.49
	Oracle + CoT&Hint	0.55	0.70	0.62	0.67	0.75	0.71	0.54	0.64	0.59	0.55	0.58	0.56	0.44	0.51	0.47	0.39	0.45	0.42

Table 7: Extended results from Table 4 including smaller open-weight models

Matching input:

Summary:
Police in Peru have clashed with squatters who have been occupying a gold mine in the north of the country.

Predicted inconsistencies:

A. The summary says the mine is in the north of the country, but the text does not mention a location.

B. The text reports one officer killed and 10 injured, but the summary leaves this out.

C. The summary calls it a gold mine, while the text never specifies the mineral.

Ground truth inconsistencies:

A. The summary is not correct because it adds the location being in Peru.

B. The summary is not correct because it adds the mine being a gold mine.

C. The summary is not correct because it adds it taking place in the north of the country.

Matching output:

{"A" : "C", "B" : None , "C" : "B"}

Figure 10: Example of a matching output. Each predicted inconsistency is either matched to one of the ground truth inconsistencies, or not matched at all.

effective in these smaller architectures. This is consistent with (Wei et al., 2022), who identified CoT as an emergent ability that typically manifests in models with at least 100B parameters. In smaller models, CoT often fails to improve, and can even degrade performance.

This trend persists in the performance of the "Self" variant - the initial step of our 2-Step setup where the evaluated LLM performs binary classification using CoT. As shown in Table 8, smaller models underperform in this task when using CoT, with Gemma-3-12B showing a particularly sharp decline. This further indicates that these models do not benefit from, and may be hindered by, CoT prompting.

Model Name	Precision	Recall	F1
GPT-4o	0.94	0.63	0.75
Claude-sonnet-3.5	0.96	0.65	0.77
Gemini-1.5-pro	0.95	0.56	0.70
Llama-3.1-405B	0.97	0.59	0.73
Ministral-3-14B	0.96	0.51	0.66
Gemma-3-12B	0.97	0.36	0.53

Table 8: Performance on the binary factual consistency evaluation task, where the model classifies the entire summary as either factually consistent or inconsistent based on whether it contains at least one factual error, like in Section 4.1.

F Multi run and variance analysis

The primary results reported in Section 5 are based on a single execution run. While we utilized greedy decoding (temperature $\tau = 0$) to prioritize consistency, two sources of variance remain: inherent non determinism during API based LLM inference (He and Lab, 2025) and the fact that the LLM-as-a-judge can still make occasional labeling errors (Zheng et al., 2023). Since running all models multiple times to establish exact error margins is extremely computationally expensive, we perform two targeted analyses to estimate these variance bounds.

Variance Across Multiple Runs To establish the robustness of our results, it is valuable to evaluate

variance across multiple independent runs. For this analysis, we focused on one of our primary models, GPT-4o. We re-executed the complete experimental pipeline (both model inference and the LLM-as-a-judge evaluation) for all setups four additional times, yielding a total of five independent runs. In Table 9, we report the mean and 95% confidence intervals calculated using the Student's t-distribution, which demonstrate that the variance across runs is indeed small.

Method	Recall	Precision	F1
Zero-Shot	0.36 ± 0.0057	0.72 ± 0.0123	0.48 ± 0.0076
Few-Shot	0.56 ± 0.0020	0.60 ± 0.0067	0.58 ± 0.0037
CoT	0.51 ± 0.0085	0.67 ± 0.0103	0.58 ± 0.0092
FactScore	0.52 ± 0.0052	0.67 ± 0.0039	0.58 ± 0.0027
Self+CoT	0.40 ± 0.0071	0.76 ± 0.0087	0.53 ± 0.0076
Self+CoT&Hint	0.42 ± 0.0045	0.75 ± 0.0054	0.54 ± 0.0043
TrueTeacher+CoT	0.47 ± 0.0085	0.73 ± 0.0122	0.57 ± 0.0100
TrueTeacher+CoT&Hint	0.50 ± 0.0039	0.69 ± 0.0049	0.58 ± 0.0040
Oracle+CoT	0.51 ± 0.0085	0.76 ± 0.0109	0.61 ± 0.0094
Oracle+CoT&Hint	0.55 ± 0.0053	0.71 ± 0.0033	0.62 ± 0.0044

Table 9: Mean Recall, Precision, and F1 with 95% confidence intervals over five independent runs of GPT-4o

Variance from LLM Judge Accuracy While LLM-as-a-judge is used extensively, it still introduces an inherent error rate. In order to compare our evaluation to prior work, we computed the accuracy of our judge's classifications, which is 93.4%. This aligns with other similar works reporting judge accuracy between 87% to 95% (Cemri et al., 2025; Li et al., 2024; Bai et al., 2024). To account for the impact of this noise on our findings, we conducted a Monte Carlo simulation over our test set. Based on manual evaluation, our judge yielded a Sensitivity of 0.95 and a Specificity of 0.90. In this context, Sensitivity represents the probability of the judge correctly matching a prediction when a corresponding entry exists in the ground truth. Conversely, Specificity represents the probability of the judge correctly not matching a prediction when there is no corresponding entry in the ground truth. Using the single-run baseline predictions, we ran 1,000 simulated evaluation iterations where we modeled judge noise by allowing labels to flip according to these manually derived rates. As shown in Table 10, the confidence intervals are tight, and the mean is almost always within 0.01 of the reported results, indicating that our findings are robust and that the inherent noise of the LLM judge does not significantly alter the conclusions of our evaluation.

G Error Analysis

As mentioned in §6, the error analysis was meant to better understand why models make mistakes. In each analysis and for each model, we sampled a random set of inconsistencies where the model was judged as incorrect and manually validated the judgement for those to exclude the possibility of judge failures.

G.1 False Negatives Analysis

In the False Negatives analysis in §6, we examine why the model misses certain inconsistencies. Based on the categorization presented in that section and Table 5, the most common category of undetected inconsistency is *Extrinsic-Correct* - claims that are factually correct but absent from the source text. This led us to the hypothesis that such claims go undetected because they align with the model's parametric knowledge. To test this, the $\mathbf{P(True)}$ metric was introduced, which quantifies the probability the model assigns to the correctness of a specific answer to a given question. To construct question–answer pairs for computing $\mathbf{P(True)}$, we reused the 150-sample set used for categorizing the inconsistencies of GPT-4o in §6, and selected those labeled as Extrinsic–Correct (102 samples) and Extrinsic–Wrong (17 samples). For each inconsistency, human annotators were asked to generate a question based on the source text, such that the answer would be the inconsistent information found in the summary, as can be seen in Figure 11. Questions were derived exclusively from the source text, so the model has the same information it had when it originally failed to flag the inconsistency.

Text
Neil Armstrong was an American astronaut and the first person to walk on the Moon during NASA's Apollo 11 mission. He famously said, "That's one small step for man, one giant leap for mankind." Armstrong was also a test pilot and aerospace engineer.

Summary
Neil Armstrong was the first person to walk on the Moon on 20, July 1969.

Question
On what date did Neil Armstrong walk the moon for the first time?

Answer
July 20th 1969.

Figure 11: Example of question generation for P(True)

To evaluate $P(\text{true})$ on GPT-4o, we followed the

		GPT-4o F1	Claude-sonnet-3.5 F1	Gemini-1.5-Pro F1	Llama-3.1-405B F1	Average F1
E2E	Zero-Shot	0.47 ± 0.0098	0.39 ± 0.0092	0.52 ± 0.0119	0.54 ± 0.0118	0.48 ± 0.0054
	Few-Shot	0.59 ± 0.0113	0.55 ± 0.0103	0.49 ± 0.0108	0.54 ± 0.0104	0.54 ± 0.0053
	CoT	0.59 ± 0.0108	0.67 ± 0.0111	0.58 ± 0.0110	0.58 ± 0.0116	0.60 ± 0.0056
Pipeline	FactScore	0.59 ± 0.0110	0.63 ± 0.0114	0.43 ± 0.0098	0.57 ± 0.0107	0.56 ± 0.0056
2-Step	Self + CoT	0.53 ± 0.0104	0.61 ± 0.0107	0.51 ± 0.0101	0.50 ± 0.0102	0.54 ± 0.0051
	Self + CoT&Hint	0.53 ± 0.0103	0.60 ± 0.0104	0.49 ± 0.0103	0.51 ± 0.0102	0.53 ± 0.0051
	TrueTeacher + CoT	0.57 ± 0.0104	0.66 ± 0.0112	0.58 ± 0.0107	0.56 ± 0.0108	0.59 ± 0.0054
	TrueTeacher + CoT&Hint	0.58 ± 0.0106	0.66 ± 0.0109	0.56 ± 0.0113	0.56 ± 0.0114	0.59 ± 0.0056
Oracle	Oracle + CoT	0.61 ± 0.0105	0.70 ± 0.0105	0.61 ± 0.0109	0.60 ± 0.0112	0.63 ± 0.0053
	Oracle + CoT&Hint	0.62 ± 0.0108	0.70 ± 0.0106	0.59 ± 0.0110	0.58 ± 0.0115	0.62 ± 0.0055

Table 10: Model performance (F1) with 95% confidence intervals derived from Monte Carlo simulations.

following procedure:

1. **Bias Elimination** We ran the prompt in two configurations:
   - Exactly as shown below.
   - With the answer options swapped (option A marked incorrect and option B correct).
2. **Repeated sampling** Because the model's responses vary slightly between runs, we executed each prompt configuration 10 times and averaged the resulting probabilities.

All sampling was done with a temperature of 1. The prompt used to estimate $P(\text{true})$ was adapted from (Gekhman et al., 2025), and is shown below.

System Prompt:

Your job is to evaluate if a proposed answer to a question is correct.

Prompt:

Here is the question and the proposed answer.

Question: <question>
Answer: <answer>

Is the proposed answer:
A: CORRECT
B: INCORRECT
Just return the letters "A" or "B", with no text around it.

G.2 False Positives Analysis

In the False Positives analysis discussed in §6, we examine cases where the models predicted that certain information was an inconsistency, even though this was not true. While the categories of false positives are specified in that section, here we provide a more detailed illustration of each category, as shown in Table 11.

H Prompts

H.1 High Recall Prompt

This is the High recall prompt mentioned in §3 and §B.3, used to maximize the amount of inconsistencies the model detects, to later be filtered by humans to uncover additional inconsistencies.

I will give you a text and a summary. The summary is factually inconsistent with respect to the text. Your task is to identify and explain all factual inconsistencies in the summary. A factual inconsistency is any information in the summary that cannot be verified by the original text.

In your explanations focus only on factual inconsistencies, and no other types of mistakes.

List all the facts you found to be inconsistent with the text. Each inconsistent fact should appear separately. List each inconsistent fact as briefly as possible, along with a short description. The fact should be the minimal span of words which is not supported by the text, and represent the mistake. The description should be brief and concise, and describe exactly what is the mistake, with no ambiguity.

The following are examples of texts, summaries and the corresponding lists of inconsistent facts.

Text:
A 27-year-old man and a woman, 32, were detained after the 60-year-old victim's body was found at the Forest Gate house, early on Christmas Day. Four people escaped from the house on Field Road before firefighters arrived just before 04:45 GMT. A post-mortem test showed the victim had died from burns and the inhalation of fumes, the Met Police said. Fire crews found his body on the ground-floor of the two-storey house. Police believe "the fire was started deliberately" and say they believe they know who the victim was, but formal identification has not yet been made. Twenty one firefighters and four engines tackled the blaze, which was brought under control after about two-and-a-half hours. Det Ch Insp Steve McCabe said: "I need to hear from anyone who saw anything suspicious in Field Road and the surrounding area in the early hours of Christmas Day.

Text:
A Japanese spacecraft successfully landed on an asteroid and collected samples in a groundbreaking mission. The Hayabusa2 probe touched down, fired a projectile to gather material, and later returned the samples to Earth. Scientists aimed to study the asteroid's composition to learn more about the origins of the solar system. The mission was hailed as a major achievement in space exploration.

Summary:
Japan's Hayabusa2 spacecraft landed on an asteroid, collected samples, and has returned them to Earth for solar system research.

Category	Definition	Model Output	Evidence	Explanation
Overlooked Info	The model fails to recognize information that is explicitly stated in the source text, resulting in an incorrect detection of an inconsistency.	The text never claimed the samples have returned to Earth.	...The Hayabusa2 probe touched down, fired a projectile to gather material, and later returned the samples to Earth...	The model simply misses this part of the text.
Missed Deduction	The model fails to recognize a fact that can be directly deduced from the text, and classifies the information as wrong.	The text does not state that Hayabusa2 is from Japan. It states an unnamed Japanese spacecraft landed on an asteroid and later discusses the Hayabusa2 landing.	A Japanese spacecraft successfully landed...The Hayabusa2 probe touched down	The model failed to recognize that both references describe the same spacecraft.
Omission	The model considers information missing from the summary as an inconsistency, even if the summary is still factually consistent.	The summary fails to mention it was a major achievement for space exploration, which is a key detail in the text.	...The mission was hailed as a major achievement in space exploration.	The model views a lack of a specific detail (it being a major achievement) as an inconsistency, even if the summary is factually correct.
Overly Literal	The model identifies superficial changes in wording as inconsistencies.	The summary claims the Hayabusa2 spacecraft landed, but the text explicitly calls it "The Hayabusa2 probe".	A Japanese spacecraft ... The Hayabusa2 probe touched down...	The model views the use of slightly different (but still supported) terminology of "spacecraft" vs "probe" as an inconsistency.
Invented	The model claims there is inconsistency, but the information it mentions as inconsistent is not in the text or summary.	The summary implies the landing is recent ("has returned"), but the text does not specify when it happened.	..and later returned the samples to Earth	The model treats 'has returned' as recent, though present perfect can describe distant events.

Table 11: Categorization of False Positives. Above the table is a text, and a factually consistent summary of it. The table presents the categories of false positives, each demonstrated with an example based on this text-summary pair. Each row shows a case where the model flagged a factual inconsistency, even though it does not exist. For each example, the table also provides an explanation of why it was assigned to that particular category.

Summary:
Two people have been arrested on suspicion of murder after a man died in a house fire in east London.

A.
Fact: arrested on suspicion of murder
Description: The summary states the people have been arrested on suspicion of murder, but the source text does not state the charges against them.
B.
Fact: east London
Description: The summary states that the fire took place in east London, but this information does not appear in the text.

Text:
The Woodland Trust wants to buy the land at Llennyrch woodland. Natural Resources Wales (NRW) has given £50,000 but a further £750,000 is needed and a campaign will be launched on Tuesday at the National Eisteddfod in Meifod, Powys. The charity said the area has been called a "Celtic rainforest" and it wants to improve wildlife on the site. NRW chief executive Dr Emyr Roberts said: "This is a fantastic opportunity to bring the whole woodland area under conservation management." The total cost of the project is £1.5m and the rest of the cost will be met by money left to the Woodland Trust.

Summary:
A campaign has been launched to raise £1m to buy 1,000 acres of woodland in Carmarthenshire.

A.
Fact: 1000 acres
Description: The summary states they want to buy 1000 acres of woodland, but acreage is not mentioned in the source text.
B.
Fact: £1m
Description: The summary states the campaign wants to raise £1m, but the source text says the campaign want to raise an additional £750,000.
C.
Fact: Carmarthenshire
Description: The summary states the woodland is in Carmarthenshire, but the source text says it's in Llennyrch.
D.
Fact: has been
Description: The summary states the campaign has been launched, but the source text says it will be launched.

Here is a new example:

Text: <Text>
Summary: <Summary>
Output the list of inconsistent facts and explanations in the same format as in the examples above.

H.2 Detection Prompts

Those are some of the prompts used to perform the detection of factual inconsistencies in §4. We have 2 variants for each **E2E** prompt, but we decided to show only one of each for brevity.

H.2.1 Zero Shot

Prompt 1:

I will give you a text and a summary. Your task is to identify and explain all factual inconsistencies in the summary. A factual inconsistency is any information in the summary that cannot be verified by the original text.

Focus only on factual inconsistencies, and no other types of mistakes, such as omission.

List all the facts you found to be inconsistent with the text. Mark each inconsistent fact with letters A, B, C, etc., in sequential order. Each inconsistent fact should appear separately. List each inconsistent fact as a short description of the inconsistency. The description should be brief and concise.

If there are no factual inconsistencies in the summary, output None.

Text: <Text>
Summary:<Summary>

Prompt 2:

I will give you a text and a summary. Your task is to identify all factual inconsistencies and briefly describe each one. If there are no factual inconsistencies, return “None.”

A “factual inconsistency” is any detail in the summary that contradicts or cannot be verified by the text. Focus only on these inconsistencies; do not address omissions or any other errors.

List each inconsistency separately, labeling them with letters (A, B, C, etc.). For each labeled item, provide a short description of what the inconsistency is. Keep your descriptions concise and only address factual inconsistencies.

Text: <Text>
Summary: <Summary>

H.2.2 Few Shot

Prompt 1:

I will give you a text and a summary. Your task is to identify and explain all factual inconsistencies in the summary. A factual inconsistency is any information in the summary that cannot be verified by the original text.

Focus only on factual inconsistencies, and no other types of mistakes, such as omission.

List all the facts you found to be inconsistent with the text. Each inconsistent fact should appear separately. List each inconsistent fact as a short description of the inconsistency. The description should be brief and concise.

If there are no factual inconsistencies in the summary, return None.

The following are examples of texts, summaries and the corresponding lists of such facts.

Text:
A 27-year-old man and a woman, 32, were detained after the 60-year-old victim’s body was found at the Forest Gate house, early on Christmas Day. Four people escaped from the house on Field Road before firefighters arrived just before 04:45 GMT. A post-mortem test showed the victim had died from burns and the inhalation of fumes, the Met Police said. Fire crews found his body on the ground-floor of the two-storey house. Police believe "the fire was started deliberately" and say they believe they know who the victim was, but formal identification has not yet been made. Twenty one firefighters and four engines tackled the blaze, which was brought under control after about two-and-a-half hours. Det Ch Insp Steve McCabe said: "I need to hear from anyone who saw anything suspicious in Field Road and the surrounding area in the early hours of Christmas Day.

Summary:
Two people have been arrested on suspicion of murder after a man died in a house fire in east London.

A.
Description: The summary states the people have been arrested on suspicion of murder, but the source text does not state the charges against them.
B.
Description: The summary states that the fire took place in east London, but this information does not appear in the text.

Text:
The Woodland Trust wants to buy the land at Llennyrch woodland. Natural Resources Wales (NRW) has given £50,000 but a further £750,000 is needed and a campaign will be launched on Tuesday at the National Eisteddfod in Meifod, Powys. The charity said the area has been called a "Celtic rainforest" and it wants to improve wildlife on the site. NRW chief executive Dr Emyr Roberts said: "This is a fantastic opportunity to bring the whole woodland area under conservation management." The total cost of the project is £1.5m and the rest of the

cost will be met by money left to the Woodland Trust.

Summary:
A campaign has been launched to raise £1m to buy 1,000 acres of woodland in Carmarthenshire.

A.
Description: The summary states they want to buy 1000 acres of woodland, but acreage is not mentioned in the source text.
B.
Description: The summary states the campaign wants to raise £1m, but the source text says the campaign wants to raise an additional £750,000.
C.
Description: The summary states the woodland is in Carmarthenshire, but the source text says it's in Llennyrch.
D.
Description: The summary states the campaign has been launched, but the source text says it will be launched.

Text:
The 14-month old tabby and white called Pumbaa was found bleeding in a Peterborough alleyway on Saturday. The stab wound was so deep the vet was unable to operate before Pumbaa died. A second cat - Mischief - was shot by an air rifle in an area near to where Pumbaa was stabbed, according to the RSPCA. It is unclear whether the two incidents are linked. RSPCA inspector Justin Stubbs said: "These were two shocking and completely senseless attacks." Pumbaa's owner, Kirsty Cracknell, 29, of Croyland Road, said: "I am utterly devastated about Pumbaa - he was such a soppy little mummy's boy. I just keep expecting him to jump through the window. "What particularly breaks my heart is that I think he must have been on his way home to me, considering where he was found."

Summary:
A cat has been stabbed to death in what the RSPCA described as a "senseless attack".

None

Here is a new example.

Text: <Text>
Summary:<Summary>

Output the list in the same format as in the examples above.

Prompt 2:

Task: Identify Inconsistencies in Summary Texts

Objective: The given summary that may contain factual inconsistencies. Your task is to critically analyze and document these inconsistencies by comparing the summary to the original text. If there are no factual inconsistencies in the summary, return None.

Definition of Factual Inconsistency:

- Any statement in the summary that cannot be directly verified or supported by the original text.
- Discrepancies in names, locations, numbers, events, or specific claims.

Evaluation Criteria:

1. Identify each distinct factual inconsistency.
2. Describe each inconsistency briefly and precisely.
3. Focus solely on factual discrepancies, not stylistic or structural differences.
4. Do not comment on omissions or missing information.

The following are examples of texts, summaries and the corresponding lists of such facts.

Text:
A 27-year-old man and a woman, 32, were detained after the 60-year-old victim's body was found at the Forest Gate house, early on Christmas Day. Four people escaped from the house on Field Road before firefighters arrived just before 04:45 GMT. A post-mortem test showed the victim had died from burns and the inhalation of fumes, the Met Police said. Fire crews found his body on the ground-floor of the two-storey house. Police believe "the fire was started deliberately" and say they believe they know who the victim was, but formal identification has not yet been made. Twenty one firefighters and four engines tackled the blaze, which was brought under control after about two-and-a-half hours. Det Ch Insp Steve McCabe said: "I need to hear from anyone who saw anything suspicious in Field Road and the surrounding area in the early hours of Christmas Day.

Summary:
Two people have been arrested on suspicion of murder after a man died in a house fire in east London.

Inconsistencies:
A.
Description: The summary states the people have been arrested on suspicion of murder, but the source text does not state the charges against them.
B.
Description: The summary states that the fire took place in east London, but this information does not appear in the text.

Text:
The Woodland Trust wants to buy the land at Llennyrch woodland. Natural Resources Wales (NRW) has given £50,000 but a further £750,000 is needed and a campaign will be launched on Tuesday at the National Eisteddfod in Meifod, Powys. The charity said the area has been called a "Celtic rainforest" and it wants to improve wildlife on the site. NRW chief executive Dr Emyr Roberts said: "This is

a fantastic opportunity to bring the whole woodland area under conservation management." The total cost of the project is £1.5m and the rest of the cost will be met by money left to the Woodland Trust.

Summary:
A campaign has been launched to raise £1m to buy 1,000 acres of woodland in Carmarthenshire.

Inconsistencies:
A.
Description: The summary states they want to buy 1000 acres of woodland, but acreage is not mentioned in the source text.
B.
Description: The summary states the campaign wants to raise £1m, but the source text says the campaign wants to raise an additional £750,000.
C.
Description: The summary states the woodland is in Carmarthenshire, but the source text says it's in Llennyrch.
D.
Description: The summary states the campaign has been launched, but the source text says it will be launched.

Text:
The 14-month old tabby and white called Pumbaa was found bleeding in a Peterborough alleyway on Saturday. The stab wound was so deep the vet was unable to operate before Pumbaa died. A second cat - Mischief - was shot by an air rifle in an area near to where Pumbaa was stabbed, according to the RSPCA. It is unclear whether the two incidents are linked. RSPCA inspector Justin Stubbs said: "These were two shocking and completely senseless attacks." Pumbaa's owner, Kirsty Cracknell, 29, of Croyland Road, said: "I am utterly devastated about Pumbaa - he was such a soppy little mummy's boy. I just keep expecting him to jump through the window. "What particularly breaks my heart is that I think he must have been on his way home to me, considering where he was found."

Summary:
A cat has been stabbed to death in what the RSPCA described as a "senseless attack".

Inconsistencies:
None

Here is a new example:
Text: <Text>
Summary: <Summary>
Output the list in the same format as in the examples above.

H.2.3 Chain of Thought

Prompt 1:

I will give you a text and a summary. Your task is to identify all the facts in the summary that cannot be verified using the text and clearly describe each factual inconsistency. Think step by step.

Text: <Text>
Summary:<Summary>

At the end of your response, under Final Output:, output each unverifiable fact with a clear description of the factual inconsistency. Mark each unverifiable fact with letters A, B, C, etc., in sequential order.

Prompt 2:

I will give you a text and a summary. Your task is to identify all factual inconsistencies by following these steps:
Examine the summary carefully and break it down into distinct factual statements.
For each statement, check whether it is:
Contradicted by the text (directly conflicts with the text).
Not verifiable from the text (lacks necessary support in the text).
For each inconsistency, describe exactly what is incorrect in the summary.

Text: <Text>
Summary: <Summary>

Think step by step, and at the end of your response, under "Final Output:", provide a clear description of each mistake.
Mark each inconsistency with letters A, B, C, etc.
Focus only on what is wrong in the summary.
If no factual inconsistencies were identified, under "Final Output:" return None.

H.2.4 CoT&Hint

These prompts are based on the CoT prompts in §H.2.3, but include an explicit statement that the summary contains inconsistencies. This addition is highlighted in the prompts below. Furthermore, in case there was an instruction in the prompt to return "None" in case of no inconsistencies, it was removed.

Prompt 1:

I will give you a text and a summary. **The summary is factually inconsistent with respect to the text.** Your task is to identify all the facts in the summary that cannot be verified using the text and clearly describe each factual inconsistency. Think step by step.

Text: <Text>
Summary:<Summary>

At the end of your response, under Final Output:, output each unverifiable fact with a clear description of the factual inconsistency. Mark each unverifiable fact with letters A, B, C, etc., in sequential order.

Prompt 2:

I will give you a text and a summary. **The summary contains factual inconsistencies with respect to the text.** Your task is to identify all factual inconsistencies by following these steps:
Examine the summary carefully and break it down into distinct factual statements.
For each statement, check whether it is:
Contradicted by the text (directly conflicts with the text).
Not verifiable from the text (lacks necessary support in the text).
For each inconsistency, describe exactly what is incorrect in the summary.

Text: <Text>
Summary: <Summary>

Think step by step, and at the end of your response, under "Final Output:", provide a clear description of each mistake.
Mark each inconsistency with letters A, B, C, etc.
Focus only on what is wrong in the summary.

H.2.5 FactScore

Decomposition prompt:

We used the original prompt given in (Min et al., 2023).

Detection prompt:

Is the following atomic fact factually consistent with the text? A factual inconsistency is any information in the summary that cannot be verified by the original text. Respond with "yes" or "no".
If and only if the answer is "no", list all the inconsistencies you identified.
Each inconsistency must appear separately and begin with a bullet mark (-). Do not group multiple issues in one bullet.
Each description must clearly identify what the atomic fact claims and explain what specific detail is missing, wrong, or not supported in the text.
Avoid vague statements like "this information can not be verified from the text."
Each description should be brief and concise.

Deduplication prompt:

You will be given a summary and a list of factual inconsistency descriptions referring to that summary.
Your task is to create a new list that removes duplicate descriptions of the same factual inconsistency.
Follow these instructions carefully:
Identify which descriptions refer to the same incorrect piece of information in the summary, even if the wording or reasoning differs.
From each group of descriptions referring to the same inconsistency, keep only the one that most clearly specifies what the exact inconsistency is. Discard the rest.
Do not drop descriptions for any reason beside merging.
Do not rewrite or modify any description. Simply select the best one among duplicates and preserve it as-is.
If no duplicates are found, do not make any changes to the input list.
Enumerate the final list using letters: A., B., C., etc.

H.3 Binary classification prompt

This is the prompt referenced in §4 for the 2-Step setup, labeled as **Self**, where the model first acts as a binary classifier before identifying individual inconsistencies.

Decide if the following summary is factually consistent with the corresponding text. Note that consistency means all information in the summary is supported by the text.

Text: <Text>
Summary: <Summary>

Explain your reasoning step by step and end your response with final answer: yes or no only

H.4 Matching prompt

This is the judgment prompt, referenced in §4 and §D, used to evaluate a model's predictions by comparing them with the ground truth.

You are tasked with analyzing two lists of descriptions: one labeled as the Predicted Output and the other as the Gold Label. The goal is to determine whether each description in the Predicted Output is specific and matches a description in the Gold Label.

Input Format
Summary: A text containing the factual inconsistencies to be evaluated.
Predicted Output: A set of descriptions labeled with identifiers (e.g., A, B, C).
Gold Label: A set of descriptions labeled with identifiers (e.g., A, B, C).

Detailed Instructions:

For each description in the Predicted Output:

Compare the exact description of the inconsistency to items in the Gold Label.
A match occurs if:
Both descriptions identify the same fact from the summary as being wrong, regardless of why that fact is considered incorrect.
Ensure the match is based on the specific information addressed in both descriptions.
The description in the Predicted Output is not ambiguous - it clearly refers to one specific fact from the summary. Vague or overly broad descriptions should not be matched.
Each description in the Predicted Output can match at most one description in the Gold Label.

If the Predicted Output is empty (no descrip-

tions were given, or any other input is given), return an empty JSON object ().
If the Gold Label is empty, return a JSON object with the keys of the Predicted Output and null value for each.

Summary: <Summary>
Gold Label: <Gold Label>
Predicted Output: <Predicted Output>

Think about the matching step by step and output a JSON object:

For each description in the Predicted Output, provide:
A key corresponding to the predicted description's identifier (e.g., A, B, C, etc.).
If a match is found, set the value to the identifier from the Gold Label that matches it.
If no match is found, set the value to null.