HalluTree: Explainable Multi-Hop Hallucination Detection for Abstractive Summarization

Oskar Oomen*, **Daniel Orshansky***, **Naaisha Agrawal, Ryan Lagasse**
Algoverse
ryan@algoverseairesearch.com

Abstract

Black-box verifiers for abstractive summaries often struggle with complex claims that require multi-hop reasoning, and they typically provide a single verdict without an interpretable rationale. As a result, it becomes difficult to understand or audit their failures. We address this with HalluTree, a framework that models verification as an interpretable claim tree. HalluTree first decomposes summaries into subclaims, classifying each into two types – extractive (directly verifiable against evidence) or inferential (requiring reasoning) – which follow distinct verification paths. Extractive claims are robustly verified against evidence using an ensemble of lightweight NLI models. Crucially, inferential claims trigger a process that generates a natural program – an explicit reasoning chain that integrates supporting evidence and logical steps – which is then executed to determine the claim's validity. Evaluation on the LLM-AggreFact benchmark demonstrates HalluTree's effectiveness: it achieves performance competitive with top-tier black-box models, including Bespoke-MiniCheck, while providing transparent and auditable reasoning programs for every inferential judgment. This combination of competitive accuracy and high interpretability offers a significant advance over opaque, single-classification verifiers.

1 Introduction

Large language models (LLMs) frequently hallucinate, producing content that is factually unsupported or incorrect (Dmonte et al., 2025; Huang et al., 2023, 2025). Even when grounded in source documents, LLM-generated summaries may contain contradictions or unverifiable statements, which can mislead readers and contribute to the spread of misinformation (Huang et al., 2025; Scirè et al., 2024). Ensuring the factual consistency of such outputs is therefore critical, particularly in domains where accuracy is paramount. In addition to raw accuracy, the explainability of these classifiers is increasingly important for transparency and human validation (Wang and Shu, 2023; Dammu et al., 2024). Without clear rationales, even correct predictions may be difficult to trust, and incorrect ones may be difficult to diagnose.

Existing work on grounded factuality verification spans a variety of strategies, including entailment-based classification, question-answering formulations, and more recent LLM-driven verification pipelines (Dmonte et al., 2025; Huang et al., 2023). While these approaches have achieved strong results in certain settings, they often provide limited transparency into the inferences behind complex judgments, and struggle with claims that require multi-hop reasoning across dispersed evidence (Belém et al., 2025). Additionally, despite the lack of fine-grained classification and weak interpretability, frontier LLMs with few-shot prompting can achieve top-tier performance even compared to the strongest specialized baselines, but still have room for improvement in complex reasoning tasks (Seo et al., 2025). This highlights the need for strong-performing and explainable verification methods that are robust to challenging multi-hop reasoning.

We propose a dual-path verification framework that decomposes a generated summary into subclaims and organizes verification results in an interpretable claim tree, with the summary as the root. The summary is first decomposed and decontextualized with an LLM to preserve original, ensuring potential hallucinations are not either introduced or inadvertently corrected away. The system then filters out unverifiable subclaims (advice, opinions, or other statements not containing factually verifiable assertions) before classifying each subclaim as extractive (directly checkable against the source) or inferential (requiring multi-hop reasoning over evidence to verify).

*Equal contribution

Proceedings of The 5th New Frontiers in Summarization Workshop, pages 123–134

Extractive subclaims are verified using two existing lightweight NLI-based hallucination detectors, LettuceDetect (Ádám Kovács and Recski, 2025) and MiniCheck-FT5 (Tang et al., 2024a), followed by evidence retrieval for ease of understanding.

Inferential subclaims trigger a reasoning pathway that gathers supporting facts from the source and beyond, which may include textual evidence, mathematical reasoning, logical inference, or unrelated elementary knowledge that need not be verified. We attach the chain-of-thought trace from the LLM when verifying the subclaim based on the supporting facts in the tree to boost auditability and ease of understanding. More importantly, these supporting facts are developed by the LLM into a natural program – a natural language-based chain of reasoning which explicitly sets out premises and rigorously demonstrates how they are composed to support the claim (Ling et al., 2023). The reasoning of this natural program is then executed and validated step-by-step by an LLM for verification, with dynamic error correction applied to detected mistakes in the program.

The verification process is represented as a hierarchical claim-tree for easy visualization, where the root corresponds to the full summary, subclaims are the children of the root, and each subclaim's children are either the evidence or supporting facts which motivate its classification, making the motivation for each subclaim's classification clear. The summary is deemed supported only if all verifiable subclaims (extractive and inferential) are supported, ensuring conservative, evidence-grounded judgments that expose the full chain of evidence and reasoning behind complex decisions, yielding stronger reliability and greater interpretability than prior single-pass verifiers.

Our primary contributions are:

- **Typed, dual-path verification.** A framework that separates extractive and inferential subclaims, verifying the former with lightweight NLI models and the latter through natural programs that rigorously combine evidence and reasoning, yielding stronger multi-hop performance.

- **Interpretable claim tree.** A hierarchical structure that links each subclaim to evidence or inferences, with natural programs making inferential reasoning rigorous and auditable rather than opaque.

- **Empirical gains.** Evaluation on AggreFact-CNN (Hermann et al., 2015), AggreFact-XSUM (Narayan et al., 2018), TofuEval (Tang et al., 2024b), and WiCE (Kamoi et al., 2023) using balanced accuracy, showing improvements over strong baselines, including GPT-4o, while also providing enhanced human-interpretability and also setting a new state-of-the-art on WiCE.

2 Related Work

Entailment-Based Methods. A major line of work frames grounded factuality classification as natural language inference (NLI) (Dmonte et al., 2025; Huang et al., 2023). Several methods fine-tune an NLI model to predict the faithfulness of a generated claim, sentence, or summary against the ground-truth source documents (Goyal and Durrett, 2020; Tang et al., 2024a; Laban et al., 2022; Kryscinski et al., 2020; Zha et al., 2023). These approaches are very efficient compared to methods involving LLMs but often offer poor interpretability due to the low granularity of the classification and the lack of a provided rationale. This can often leave what is truly causing the classification ambiguous to humans. To combat this, some methods operate at a token or span level (Ádám Kovács and Recski, 2025; Rawte et al., 2025; Belyi et al., 2025). These provide a finer granularity of classification, often helping pinpoint the precise subclaims which are not faithful. However, since these methods only operate on spans of the generated text, when multi-hop reasoning is involved, they may still fail to reveal the intermediate inferences or supporting facts that connect the evidence to the claim, as such implicit inferences do not appear in the generated text.

QA-Based Metrics. Another prominent class of faithfulness evaluation methods frames factuality checking as a question-answering (QA) problem (Scialom et al., 2021; Fabbri et al., 2022; Wang et al., 2020). In these approaches, a set of questions is typically generated from the summary. A QA model is then used to answer these questions given the source documents, and the answers are compared to the corresponding content in the summary (Huang et al., 2023). High overlap or semantic similarity indicates factual consistency, whereas discrepancies signal potential hallucinations. While QA-based metrics offer the advantage of explicitly tying verification to discrete factual questions,

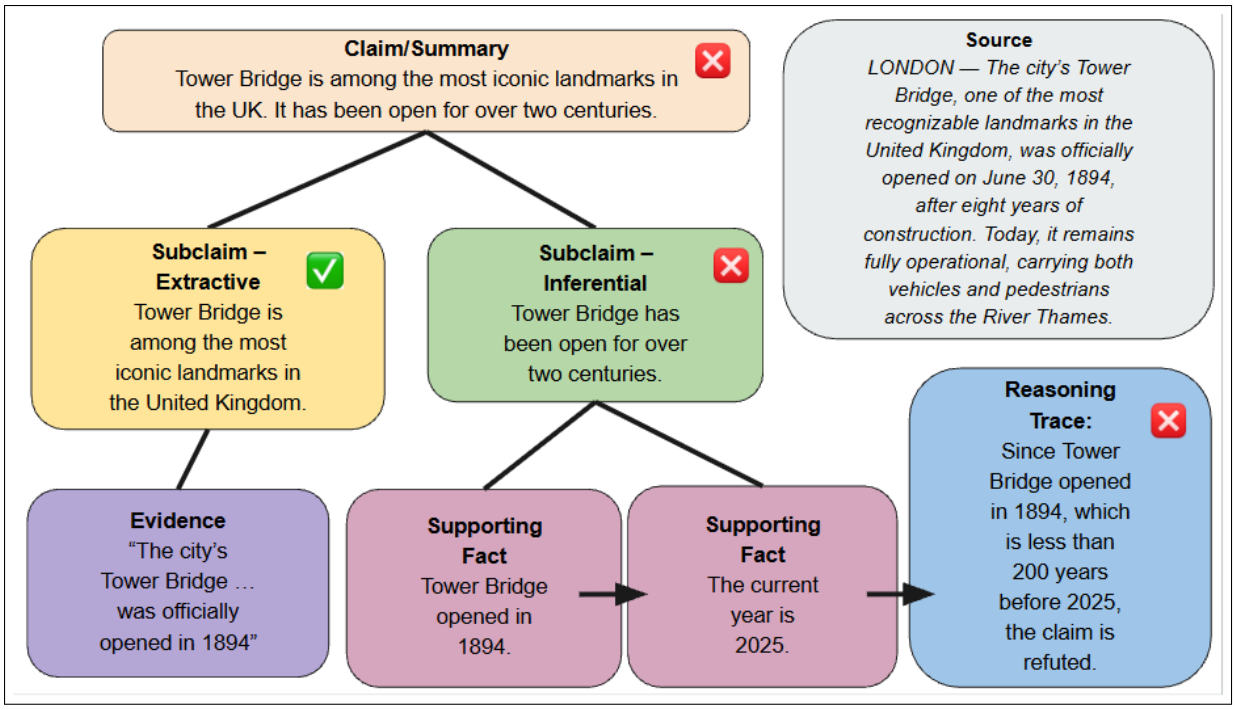

Figure 1: This diagram shows a visualization of our method's claim tree structure as well as a demonstration of how our chain-of-thought reasoning traces enhance explainability.

multi-hop reasoning may still be unexplored or left implicit as the reasoning required to support or refute a claim is often only able to emerge indirectly through evidence from the source rather than being explicitly modeled from the surface-level claims. Additionally, effectively verifying how inferences over evidence connect to multi-hop claims may require reasoning over both the source and the summary, rather than relying on isolated question–answer pairs.

LLM-based Metrics. A growing body of work leverages LLMs for factuality evaluation by decomposing outputs into smaller, verifiable units (Huang et al., 2025). A common approach, often termed *Decompose-then-Verify*, prompts an LLM to split a generated summary or claim into atomic statements, assess each against source evidence, and then aggregate the results into an overall factuality judgment (Hu et al., 2025; Lu et al., 2025; Zheng and Lee, 2025; Akbar et al., 2024). This approach improves interpretability over coarse-grained entailment scoring by providing statement-level judgements. While the faithfulness judge is typically an LLM, FENICE (Scirè et al., 2024) instead applies an NLI model to each subclaim.

Our method builds on this paradigm but introduces several extensions. First, we propose a dual-typed classification of subclaims into extractive and inferential, with separate verification pathways. Like FENICE, extractive claims are handled with lightweight NLI-based verifiers. In contrast, inferential claims trigger a reasoning process that collects supporting facts—which may include source evidence or logical and mathematical inferences—and evaluates whether they logically support the claim through a natural program, a structured reasoning sequence executed and validated step by step, marking its first use in factuality verification for summarization.

3 Methodology

Our claim verification framework operates through a multi-stage process that decomposes generated text into subclaims, verifies each subclaim according to its relation to the evidence, and organizes the verification results in an interpretable tree structure. Given a summary, the framework constructs a tree where the root node represents the full summary, intermediate nodes represent subclaims, and leaf nodes contain evidence snippets or supporting facts.

3.1 Claim Decomposition

We begin by prompting the LLM (we use GPT-4o for our purposes) to break the summary into decontextualized subclaims. Rather than decom-

posing into atomic claims, we decompose the summary into subclaims mirroring the structure of a sentence or complete clauses as closely as possible, to preserve semantic fidelity and avoid over-decomposition. For decontextualization, pronouns and ambiguous references are replaced by the LLM with explicit entity mentions, making each subclaim interpretable in isolation.

3.2 Subclaim Classification

The LLM then classifies each subclaim into one of three categories, determining the verification pathway that follows.

Extractive: Subclaims that can be directly supported or refuted by textual evidence from the source without requiring additional reasoning steps.

Inferential: Subclaims that are not directly supported or refuted by evidence in the documents and may require multi-hop reasoning or logical inference to judge their veracity.

Unverifiable: Subclaims that represent opinions, judgements, or unrelated elementary knowledge not about the subject(s) of the source. These are excluded from further processing.

3.3 Verification Process

Depending on the classification of the subclaim, the verification process differs.

Subclaims classified as extractive are evaluated on LettuceDetect (Ádám Kovács and Recski, 2025) and MiniCheck-FT5 (Tang et al., 2024a) for verification, lightweight NLI models for groundedness classification. We aggregate the results of these models by deeming the claim unsupported if both models find it unsupported; otherwise, we deem the claim supported.

For subclaims classified as inferential, the system initiates a reasoning pathway. The LLM first proposes a set of supporting facts—drawn from the source text, logical or mathematical reasoning, or elementary knowledge not requiring verification—and orders them so they can form a coherent reasoning chain. Finally, we prompt an LLM to judge the groundedness of the claim given the supporting facts with chain-of-thought reasoning. These facts are added as children of the subclaim node in the verification tree, and additionally, to increase explainability, we attach the chain-of-thought reasoning from the LLM's verification to provide the rationale and logical connection between the supporting facts and the claim.

After initial judgment, for each inferential subclaim, the LLM constructs a natural program based off its supporting facts using few-shot prompting adapted from Ling et al. (2023) in order to verify the judgment rigorously. A natural program is a natural language-based chain of reasoning that explicitly lays out premises and demonstrates how they compose to support the claim. Crucially though, this does not simply verify the judgment rigorously but provides an interpretable demonstration of the underlying reasoning to humans, allowing even the rigorous reasoning between the premises and the claim to be audited. This reasoning for the natural program is executed and validated step-by-step by the LLM, with mistakes detected within the original classification being dynamically corrected.

3.4 Evidence Retrieval for Extractive Subclaims

To collect the relevant evidence snippets for extractive subclaims, we first segment the source into manageable chunks. The LLM is then prompted iteratively, selecting the most relevant chunk with respect to verifying the subclaim and decides whether additional evidence is needed. This process continues until the model judges the gathered evidence sufficient for verification. Finally, we add all collected snippets as children of the respective subclaim.

3.5 Final Verification

After the independent verification of subclaims, the system considers that the original claim is supported only if all verifiable subclaims (extractive and inferential) are individually supported.

4 Experiments and Results

4.1 Datasets

To evaluate our method, we evaluate the balanced accuracy of our model on binary factuality verification tasks from several established datasets from the LLM-AggreFact benchmark (Tang et al., 2023) that are established in faithfulness verification for abstractive summarization.

AggreFact-CNN includes generated summaries of CNN/DailyMail articles from the CNN/DM dataset (Hermann et al., 2015). The dataset consists of source news articles from the CNN/DailyMail corpus, generated summaries produced by various summarization models, and binary hallucination

Model	AggreFact		TofuEval		WiCE	AVG
	CNN	XSUM	MediaS	MeetingB		
GPT-4o-2024-08-06	67.5	73.9	66.0	**81.1**	74.3	72.6
AlignScore	**73.2**	72.4	67.1	76.5	69.6	71.8
LettuceDetect-large-v1	58.3	67.7	65.8	69.6	79.2	68.1
MiniCheck-FT5	69.9	74.3	73.6	77.6	72.4	73.5
Bespoke-MiniCheck-7B	65.5	**77.8**	**76.0**	78.3	83.0	**76.1**
HalluTree	68.5	74.5	66.4	79.8	**83.7**	74.6

Table 1: Balanced accuracy (%) on datasets from LLM-AggreFact. Highest score is bolded. Second highest is underlined.

Model	AggreFact		TofuEval		WiCE	AVG
	CNN	XSUM	MediaS	MeetingB		
HalluTree (Dual-Pathed)	**68.5**	**74.5**	**66.4**	**79.8**	**83.7**	**74.6**
Treat All Extractive	58.4	71.0	57.8	68.6	80.1	67.2
Treat All Inferential	63.0	71.9	65.8	79.2	83.0	72.6

Table 2: Results of ablation study on dual-paths for subclaims.

labels indicating whether summaries contain factual inconsistencies with respect to their source articles. We evaluate on the 558 examples in LLM-AggreFact.

AggreFact-XSum contains generated summaries of BBC articles from the XSum corpus (Narayan et al., 2018). Like AggreFact-CNN, it provides binary hallucination labels for summaries generated by various models. We evaluate on the 558 examples in LLM-AggreFact.

WiCE is a fine-grained textual entailment dataset built on claim and evidence pairs extracted from Wikipedia (Kamoi et al., 2023). The data set uses real-world examples extracted from Wikipedia sentences, evidence articles to which the claims refer, fine-grained entailment judgments over subsentence units, and minimal subsets of evidence sentences supporting each sub-claim. WiCE includes challenging verification and retrieval problems involving multi-sentence reasoning. We evaluate on the 358 examples from this dataset in LLM-AggreFact.

TofuEval contains two factuality evaluation tasks – **MediaS** and **MeetingB** – drawn from the TofuEval benchmark (Tang et al., 2024b), which was designed to assess LLM factual consistency across multiple domains. MediaS consists of summaries of news and media sources with binary factuality annotations, while MeetingB consists of generated summaries of meeting transcripts, annotated for consistency with the meeting records. These datasets broaden evaluation coverage to conversational and multi-speaker domains, providing a more diverse testbed for factual verification methods.

4.2 Baselines

We compare our method against strong baselines spanning both NLI-based and LLM-based verification approaches. On the NLI side, we include **LettuceDetect** (Ádám Kovács and Recski, 2025), **MiniCheck-FT5** (Tang et al., 2024a), and **AlignScore** (Zha et al., 2023), which use lightweight natural language inference models to detect hallucinations and assess faithfulness. Among LLM-based systems, we consider chain-of-thought prompting **GPT-4o**, a state-of-the-art model frequently used for faithfulness assessment, as well as **Bespoke-MiniCheck-7B** (Tang et al., 2024a), a state-of-the-art finetuned model which outperforms frontier models on LLM-AggreFact.

4.3 Main Results

Table 1 presents balanced accuracy across the datasets selected from LLM-AggreFact. HalluTree achieves the second-highest average accuracy (74.6%), outperforming all baselines we tested except for Bespoke-MiniCheck. While Bespoke-MiniCheck attains a slightly higher average accuracy, HalluTree offers a key advantage in trans-

Snippet from Natural Program Output for Inferential Subclaim (WiCE)

```
Premises
#1. ROOT (R4): “Currently, SriLankan operates an all-Airbus fleet with the
exception of its discontinued Air-Taxi services.”
#2. S4 (INFERENTIAL): “SriLankan Airlines currently operates an all-Airbus
fleet with the exception of SriLankan Airlines' discontinued Air-Taxi services.”
#3. E4A: The context mentions a fiasco involving the launch of an air taxi
service
which was eventually abandoned causing millions of dollars in losses to
SriLankan Airlines.
#4. E4B: The context does not provide any specific information about the\
current  fleet composition of SriLankan Airlines being exclusively Airbus.

Reasoning
#5. (by #3) The air taxi service of SriLankan Airlines was discontinued.
#6. (by #4) There is no evidence in the context to confirm that SriLankan
Airlines operates an all-Airbus fleet currently.
#7. (by #5, #6) While the air taxi service is confirmed to be discontinued, the
claim that the current fleet is all-Airbus is not supported by the provided
evidence; S4 is not supported.

Subclaim Status
- S4: Not Supported — The context confirms the discontinuation of the air taxi
service but does not confirm that the current fleet is exclusively Airbus.
```

Figure 2: An example natural program generated during verification of a inferential claim from the WiCE dataset.

parency. The finer-grained decomposition into subclaims, coupled with hierarchical verification trees, makes the reasoning process auditable and interpretable.

Importantly, HalluTree outperforms all NLI-based methods such as AlignScore, MiniCheck-FT5, and LettuceDetect, demonstrating that structured decomposition paired with specialized verification pathways can yield stronger performance than flat entailment classification. This shows that HalluTree narrows the gap with Bespoke-MiniCheck while introducing interpretable reasoning, enabling both competitive accuracy and improved transparency.

4.4 Explainability in Practice

Unlike black-box verifiers that surface only a final label, HalluTree exposes the full reasoning trail for each decision. Consider the Natural Program excerpt for the WiCE subclaim about SriLankan Airlines' fleet: the system (i) lists concrete premises (#1–#4), separating source evidence from assumptions; (ii) derives intermediate conclusions (#5–#6) with provenance (e.g., “by #3”); and (iii) composes these steps into a final inference (#7) that justifies the verdict of "Not Supported" because the context confirms the air-taxi discontinuation but lacks evidence that the current fleet is all-Airbus. This structured trace makes the decision auditable: a reviewer can pinpoint exactly which premise would need revision to flip the outcome.

Empirically, HalluTree matches or exceeds most state-of-the-art baselines while providing superior transparency. For extractive claims, span-level evidence highlights show where the text is (or isn't) supported; for inferential claims, natural programs show why—linking premises to conclusions via explicit, checkable steps. The result is a verifier that not only performs competitively but also turns factuality judgments into explanations that users can inspect, contest, and improve.

4.5 Ablations

To better understand the effect of our dual-path routing, we conduct two ablations. First, we evaluate variants that route all subclaims through the extrac-

tive pathway (*All-Extractive*) or through the inferential pathway (*All-Inferential*). Second, we analyze the distribution of claim types across datasets along with our method's performance on those datasets considering the proportion of extractive and inferential claims.

Routing Variants. Table 2 shows balanced accuracy for the ablated models. Both constrained settings degrade performance: *All-Extractive* struggles on inference-heavy datasets, while *All-Inferential* incurs extra reasoning cost while having worse performance on extractive-heavy datasets. These results demonstrate that routing based on subclaim type enables our method to selectively apply natural program reasoning where it matters, improving accuracy on complex inferential claims, while avoiding unnecessary overhead on extractive ones.

Claim-Type Distribution. We also measure the proportion of extractive vs. inferential subclaims in each dataset (Table 3). Comparing our results from Table 1, our method performed strongly compared to other baselines on datasets with heavy inferential subclaim ratios, such as WiCE and AggreFact-XSUM, while generally maintaining more average performance on extractive-heavy datasets. This distribution provides an explanation for where type-aware routing yields the largest gains.

Dataset	Extractive	Inferential
AggreFact-CNN	96.9	2.1
AggreFact-XSUM	50.4	49.6
TofuEval-MediaS	85.6	14.4
TofuEval-MeetingB	81.2	18.8
WiCE	42.0	58.0

Table 3: Subclaim type distribution (% of verifiable subclaims).

Summary. These ablations highlight that type-aware decomposition and routing are not only interpretable but also empirically necessary: forcing all claims into a single pathway reduces accuracy, while claim-type distributions explain why balanced routing achieves consistent gains.

5 Conclusion

We present a hierarchical claim verification framework that advances the state-of-the-art in hallucination detection by providing both accurate classification and human-interpretable explanations. Unlike black-box approaches that output only binary classifications, our framework makes the verification process transparent through a tree-based structure that traces the pipeline from claim decomposition to evidence gathering and reasoning.

Our method provides several key advantages. It offers fine-grained explainability by attaching concrete evidence or inferences to each subclaim and by generating Natural Programs—explicit, natural language reasoning chains that demonstrate how inferential claims are logically supported. This hybrid verification design combines lightweight transformer-based models for extractive verification with LLM-based reasoning for more complex inferential claims, organized in a natural tree structure for clarity.

Experimental evaluation across four diverse datasets—AggreFact-CNN, AggreFact-XSUM, TofuEval, and WiCE—demonstrates the effectiveness of our approach. Our method achieves competitive performance with an average balanced accuracy of 74.6%, exceeding GPT-4o while providing detailed reasoning traces that enhance transparency. This combination of strong performance with explicit reasoning via Natural Programs represents a significant advance over existing black-box approaches.

Such transparency is crucial for practical deployment, where understanding not just whether a claim is supported but also why it is supported or refuted can help identify weaknesses, build trust, and improve reliability in real-world applications.

Limitations

Performance Tradeoffs. HalluTree improves on inference-heavy datasets but underperforms on extractive-heavy ones, where some simpler entailment-based baselines remain stronger. This reflects that our specialized reasoning pathway benefits complex claims, but introduces unnecessary overhead and noise when most claims can be directly verified against the source.

Granularity of Judgments. Our framework outputs binary faithful/unfaithful decisions at the claim level. While subclaims are verified individually, the final aggregation does not capture intermediate degrees of support or uncertainty, which could limit usefulness in downstream applications that require nuanced reliability scores.

Computational Overhead. Compared to single-pass verification methods, HalluTree incurs significantly higher cost. Each stage—decomposition, classification, evidence selection, and verification—requires separate LLM calls. This overhead grows with claim length and makes the method less practical for large-scale or latency-sensitive deployments.

Reliance on LLM Quality. Errors in early stages, such as decomposition or classification, propagate through the pipeline and can compromise verification accuracy. In particular, misclassification between extractive and inferential claims can route subclaims through an inappropriate verification pathway, lowering performance.

Evaluation Scope. Our experiments are limited to benchmark datasets that primarily focus on factual consistency in summarization. Broader domains—such as multimodal sources, conversational data, or more diverse factuality errors—may expose different challenges not addressed by our current framework.

References

Shayan Ali Akbar, Md Mosharaf Hossain, Tess Wood, Si-Chi Chin, Erica M Salinas, Victor Alvarez, and Erwin Cornejo. 2024. HalluMeasure: Fine-grained hallucination measurement using chain-of-thought reasoning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15020–15037, Miami, Florida, USA. Association for Computational Linguistics.

Catarina G Belém, Pouya Pezeshkpour, Hayate Iso, Seiji Maekawa, Nikita Bhutani, and Estevam Hruschka. 2025. From single to multi: How LLMs hallucinate in multi-document summarization. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 5276–5309, Albuquerque, New Mexico. Association for Computational Linguistics.

Masha Belyi, Robert Friel, Shuai Shao, and Atindriyo Sanyal. 2025. Luna: A lightweight evaluation model to catch language model hallucinations with high accuracy and low cost. In *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*, pages 398–409, Abu Dhabi, UAE. Association for Computational Linguistics.

Preetam Prabhu Srikar Dammu, Himanshu Naidu, Mouly Dewan, YoungMin Kim, Tanya Roosta, Aman Chadha, and Chirag Shah. 2024. ClaimVer: Explainable claim-level verification and evidence attribution of text through knowledge graphs. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 13613–13627, Miami, Florida, USA. Association for Computational Linguistics.

Alphaeus Dmonte, Roland Oruche, Marcos Zampieri, Prasad Calyam, and Isabelle Augenstein. 2025. Claim verification in the age of large language models: A survey. *Preprint*, arXiv:2408.14317.

Alexander 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, Seattle, United States. Association for Computational Linguistics.

Tanya Goyal and Greg Durrett. 2020. Evaluating factuality in generation with dependency-level entailment. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3592–3603, Online. Association for Computational Linguistics.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Qisheng Hu, Quanyu Long, and Wenya Wang. 2025. Decomposition dilemmas: Does claim decomposition boost or burden fact-checking performance? In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6313–6336, Albuquerque, New Mexico. Association for Computational Linguistics.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.*, 43(2).

Yichong Huang, Xiachong Feng, Xiaocheng Feng, and Bing Qin. 2023. The factual inconsistency problem in abstractive text summarization: A survey. *Preprint*, arXiv:2104.14839.

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

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.

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

Zhan Ling, Yunhao Fang, Xuanlin Li, Zhiao Huang, Mingu Lee, Roland Memisevic, and Hao Su. 2023. Deductive verification of chain-of-thought reasoning. In *Advances in Neural Information Processing Systems*, volume 36, pages 36407–36433. Curran Associates, Inc.

Yining Lu, Noah Ziems, Hy Dang, and Meng Jiang. 2025. Optimizing decomposition for optimal claim verification. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5095–5114, Vienna, Austria. Association for Computational Linguistics.

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

Vipula Rawte, S.m Towhidul Islam Tonmoy, Shravani Nag, Aman Chadha, Amit Sheth, and Amitava Das. 2025. FACTOID: FACtual enTailment fOr hallucInation detection. In *Proceedings of the 5th Workshop on Trustworthy NLP (TrustNLP 2025)*, pages 599–617, Albuquerque, New Mexico. Association for Computational Linguistics.

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano, Alex Wang, and Patrick Gallinari. 2021. QuestEval: Summarization asks for fact-based evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6594–6604, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Alessandro Scirè, Karim Ghonim, and Roberto Navigli. 2024. FENICE: Factuality evaluation of summarization based on natural language inference and claim extraction. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14148–14161, Bangkok, Thailand. Association for Computational Linguistics.

Wooseok Seo, Seungju Han, Jaehun Jung, Benjamin Newman, Seungwon Lim, Seungbeen Lee, Ximing Lu, Yejin Choi, and Youngjae Yu. 2025. Verifying the verifiers: Unveiling pitfalls and potentials in fact verifiers. *Preprint*, arXiv:2506.13342.

Liyan Tang, Tanya Goyal, Alex Fabbri, Philippe Laban, Jiacheng Xu, Semih Yavuz, Wojciech Kryscinski, 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)*, pages 11626–11644, Toronto, Canada. Association for Computational Linguistics.

Liyan Tang, Philippe Laban, and Greg Durrett. 2024a. MiniCheck: Efficient fact-checking of LLMs on grounding documents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8818–8847, Miami, Florida, USA. Association for Computational Linguistics.

Liyan Tang, Igor Shalyminov, Amy Wong, Jon Burnsky, Jake Vincent, Yu'an Yang, Siffi Singh, Song Feng, Hwanjun Song, Hang Su, Lijia Sun, Yi Zhang, Saab Mansour, and Kathleen McKeown. 2024b. TofuEval: Evaluating hallucinations of LLMs on topic-focused dialogue summarization. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4455–4480, Mexico City, Mexico. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.

Haoran Wang and Kai Shu. 2023. Explainable claim verification via knowledge-grounded reasoning with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6288–6304, Singapore. Association for Computational Linguistics.

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

Zhi Zheng and Wee Sun Lee. 2025. Reasoning-cv: Fine-tuning powerful reasoning llms for knowledge-assisted claim verification. *Preprint*, arXiv:2505.12348.

Ádám Kovács and Gábor Recski. 2025. Lettucedetect: A hallucination detection framework for rag applications. *Preprint*, arXiv:2502.17125.

A Appendix

A.1 Prompts

Decomposition and Decontextualization Prompt

> You are given a summary text. Your task is to decompose it into subclaims that mirror the sentence-like structure of the original as closely as possible. Each subclaim should

be decontextualized, meaning it must stand on its own and be understandable without reference to the surrounding text.

Guidelines:
Preserve sentence alignment: Each subclaim should correspond to one sentence in the original summary wherever possible, or the closest equivalent if sentences are not present.
Minimal splitting: Do not overdecompose by introducing claims not directly stated in the text.
No merging or compression: Each subclaim should stay as close as possible to its original sentence(s).
Decontextualize: Rewrite each subclaim so that it is fully interpretable in isolation, avoiding pronouns or vague references.
Output format: Output sentences in a numbered list (1. 2. 3. etc) with each sentence on its own line.

Source document for context: {}
Summary: {}
Subclaims:

Subclaim Classification Prompt

You are an expert at classifying sentences based on their relationship to the provided context and the subjects (the main entities or events the context is about) of that context.

Classifications:
UNVERIFIABLE: Contains opinions or judgments, and background/common-knowledge or definite-truth statements not about the subject(s) of the source. Includes math or logic truths, calendar arithmetic, unit conversions, definitional or taxonomic facts, and geographic containment that do not need verification against the context. These are often bridge facts used to connect evidence.
EXTRACTIVE: Contains claims that are directly supported or directly refuted by explicit spans in the context without reasoning.
INFERENTIAL: Contains claims about the subject(s) of the source that are not directly supported or refuted by the context and require multi-hop reasoning over the provided evidence. They may rely on UNVERIFIABLE background facts as bridges, but the claim itself is about the subject(s).

Rubric:
1) Identify the subject(s) of the source.
2) If the ENTIRE claim is a background or definite-truth proposition not about the subject(s) of the source, classify as UNVERIFIABLE.
3) Else, if explicit context spans support or refute the claim, classify as EXTRACTIVE.
4) Else, classify as INFERENTIAL.
Tie-breakers:
- Prefer UNVERIFIABLE for math, logic, calendar arithmetic, unit conversions, definitional or lexical truths, and geography containment that are not about the subject(s).
- Do not mark as UNVERIFIABLE if the statement asserts a property or relation of the subject(s), even if widely known; that is INFERENTIAL unless directly supported by the context.
- If deciding requires external, subject-specific facts not in the context, classify as INFERENTIAL.
1. First reason toward your decision. Do not decide until after you have reasoned.
2. After reasoning, output exactly one label from UNVERIFIABLE, EXTRACTIVE, INFERENTIAL on a new line and nothing else.

Context: {}
Claim: {}
Let's think step by step:

Evidence Collection Prompt

You are an expert at extracting evidence from context to support or refute a subclaim.

Critical Rules:
1. If possible, extract the span of evidence that is most directly relevant to the sub-

claim.
2. Don't repeat evidence that has already been collected.
3. If there is truly no additional relevant evidence in the context, output the token <NO_MORE_EVIDENCE>

Context: {}
Subclaim: {}
Already collected evidence (do not repeat):
Next evidence:

Supporting Fact Proposal Prompt

You are an expert at constructing logical bridges between evidence and an inferential subclaim to either support or refute the subclaim.

Terminology:
- EVIDENCE fact: directly supported by explicit spans in the context.
- BACKGROUND fact: elementary common knowledge or definite truth (math, logic, calendar arithmetic, definitions, geography containment) that is not about the subject(s) of the source and does not require verification against the context. Use only if needed to connect evidence to the subclaim.

Critical Rules:
1. Include EVIDENCE facts only if they are explicitly supported by the context. Closely paraphrase or directly copy the supporting span.
2. You may include BACKGROUND facts that are not about the subject(s) and are necessary to form the reasoning chain. Do not introduce subject-specific facts that are absent from the context.
3. Order the facts so they form a minimal, coherent chain that best supports or refutes the subclaim.
4. Do not add new, subject-specific information. If the context provides nothing usable, output the token <NO_SUPPORTING_FACTS>.
5. Reason first, then output the FACTS.

Example:
Context: "Aspirin was first synthesized in 1897 by chemist Felix Hoffmann at Bayer."
End of Context

Inferential subclaim: "Aspirin was synthesized over a century ago"

Let's think step by step: From the context we know the synthesis year is 1897. Using current-year arithmetic, 1897 is more than 100 years before 2025, so the subclaim is supported.

FACTS:
1. Aspirin was first synthesized in 1897
2. The current year is 2025.
3. 1897 is more than 100 years before 2025.

Context: {}
Inferential subclaim: {}
Let's think step by step:

Inferential Subclaim Verification Prompt

You are an expert at judging whether a set of proposed supporting facts logically supports an inferential subclaim.

Critical Rules:
1. Use only the facts provided; do not rely on any external knowledge or assumptions except for cases of common knowledge or facts that need not be verified.
2. The supporting facts should be able to form a coherent reasoning chain that directly supports the subclaim.
3. Output sections in this order: Reasoning, then final judgment ("YES" or "NO"). YES for supported, NO for refuted.
4. Don't be pedantic in your judgments, direct contradictions or completely unfounded statements are mainly what we seek to prevent. Refuted claims should be clearly, strongly refutable.

Example:
Context:
"Aspirin was first synthesized in 1897 by

chemist Felix Hoffmann at Bayer."
End of Context

Supporting facts:
1) [EVIDENCE] "Aspirin was first synthesized in 1897 ..."
2) [BACKGROUND] The current year is 2025.
3) [BACKGROUND] 1897 is more than 100 years before 2025.

Inferential subclaim: "Aspirin was synthesized over a century ago"

Let's think step by step: The facts provide the synthesis year, the current year, and the difference being more than 100 years. This supports the the subclaim.
Is the claim supported: YES

Context: {}
Supporting facts: {}
Inferential subclaim: {}
Let's think step-by-step:

GPT-4o Baseline Faithfulness Verification Prompt

Your task is to check if the Summary is accurate to the Evidence.
Generate 'Supported' if the Summary is supported when verified according to the Evidence, or 'Unsupported' if the Summary is inaccurate (contradicts the evidence) or cannot be verified.

Evidence: {}
End of Evidence

Summary: {}
End of Summary

Classification ('Supported' or 'Unsupported'):