Evidence-Grounded Claim-Level AI Hallucination Detector

Mrs. Radhika S N[1], Adithi S[2], Anagha G S[3], Ananya S J[4], Ashwini B[5]

[1]Assistant Professor, [2,3,4,5]UG Students, Department of Computer Science and Engineering, JNNCE, Visvesvaraya Technological University, Karnataka, India

Abstract

Large Language Models (LLMs) write smooth and convincing prose, yet that prose may include statements that are unsupported, wrong, or invented outright. These errors, known as hallucinations, weaken trust in LLMs when they are used for teaching, scholarly work, search, and automated writing. This paper describes a detector that grounds its judgments in external evidence and inspects text one claim at a time, instead of giving a whole answer a single verdict. The system cleans the incoming text, isolates factual claims and named entities, builds search queries aimed at verification, gathers evidence from web and news sources, and checks every claim against what was found. Each claim receives one of three labels (Supported, Contradicted or Unverified), and the response as a whole receives a hallucination-risk indication accompanied by the supporting evidence and a short explanation. The design draws on earlier work on characterising hallucination, sampling-based consistency checks, atomic-fact scoring, and hallucination analysis for retrieval-augmented generation. The implementation relies on Python, FastAPI, the Tavily API, the News API and MongoDB. In an initial test on true and false statements taken from several knowledge areas, the system reached 85% accuracy, 84% precision, 87% recall and an 87% F1-score, with 85% evidence coverage and roughly five seconds per verification on average. These findings suggest that claim-level checking against evidence gives users clearer and more transparent feedback than a single label per response, although ambiguous claims, uneven source quality, time-dependent facts and failed retrievals remain difficult cases.

Keywords

Large Language Models, Hallucination, Claim Verification, Evidence Retrieval, Natural Language Processing, Fact Checking, Factuality, Retrieval-Augmented Verification

I. INTRODUCTION

LLMs have become core building blocks of modern AI applications. They produce fluent, context-aware text for question answering, summarisation, tutoring, programming help, research support and content creation. Fluency, though, is no guarantee of accuracy. Surveys of natural language generation separate several kinds of factual inconsistency and stress that a passage can sound believable while containing material that nothing supports [1]. The danger is that readers tend to treat a confident tone as a sign of truth.

The problem grows when a single answer packs in many facts. An otherwise accurate response may hold one wrong date, place, name, figure or causal link, and a blanket 'hallucinated' tag gives no hint of which piece needs fixing. Evaluation therefore has to work at a finer grain. FActScore split long generated passages into atomic facts and reported the share of those facts backed by trustworthy sources [2]; that idea underlies the claim-level design used in this work.

A second line of work checks consistency without looking inside the model. SelfCheckGPT draws several random samples from an LLM and compares them to judge how likely a statement is to be true, needing neither token probabilities nor an external knowledge base [3]. Such signals work with any model, but an evidence-grounded detector offers something different: a concrete outside source that the reader can open and examine. This matters wherever transparency and traceability are required.

Retrieval, however, does not by itself make an answer correct. RAGTruth showed that retrieval-augmented generation still yields unsupported or conflicting statements, and it supplied a corpus for studying them [4]. A verification pipeline must therefore do more than fetch documents; it has to judge how each claim relates to the evidence found for it.

Our system meets this need by chaining preprocessing, claim extraction, named entity recognition, query generation, evidence retrieval, claim-evidence comparison, risk scoring and explanation generation. It is built as a self-contained

verification layer that sits behind an LLM. It neither alters nor retrains the model; it inspects the finished answer and marks which claims look supported, contradicted, or lacking adequate evidence.

The work pursues five goals: (i) to split AI-generated answers into separately checkable claims; (ii) to find external evidence for every claim; (iii) to decide whether that evidence agrees or conflicts with the claim;
(iv) to derive a response-level reliability indication from the claim outcomes; and (v) to present evidence and explanations so that users can see why a decision was reached.

II. RELATED WORK

Work on hallucination detection has progressed along several complementary paths: analysis of intrinsic versus extrinsic factuality, uncertainty estimation from the model, consistency checking, retrieval-based fact verification, and the creation of benchmarks. Ji et al. wrote a broad survey of hallucination in natural language generation that covers definitions, categories, evaluation approaches, mitigation options and task-specific difficulties [1]. The survey makes clear that hallucination is not one uniform phenomenon, so evaluation must take into account both the task and the source material available.

Manakul, Liusie and Gales proposed SelfCheckGPT, a black-box approach that compares several responses sampled from the same prompt. The intuition is that statements the model genuinely knows tend to stay stable across samples, while fabricated ones tend to drift or clash [3]. This is appealing when neither model internals nor an external database can be used. Our system takes the opposite route and relies on retrieved evidence, so that each verdict can be traced to a source. The two strategies may also be combined, since each supplies a different signal.

FActScore introduced a fine-grained measure of factual precision. It breaks generated text into atomic facts and tests each one against a reliable knowledge source [2]. This is directly relevant here because it justifies replacing one response-level decision with independent factual units. Our detector turns that general principle into a practical pipeline in which claims are extracted, searched for, matched with evidence, and reported one by one.

HaluEval is a large benchmark of generated and human-annotated hallucinated samples, built to test whether LLMs can recognise hallucinated content [5]. It shows that detectors need to be tried on varied and difficult examples, not only on blatantly false statements, and its authors report that supplying outside knowledge or reasoning steps improves recognition, which supports the use of evidence for verification.

RAGTruth concentrates on hallucinations in retrieval-augmented generation. It offers close to 18,000 naturally produced responses annotated at both the case level and the word level [4]. Its results matter for our design because they show that retrieval alone does not remove hallucination: a response can still contain claims that the retrieved text does not support or even contradicts. Retrieval must consequently be followed by an explicit comparison of claim and evidence.

Several other studies address neighbouring problems. TruthfulQA measures how far language models reproduce common human falsehoods [6]. Long-form factuality work evaluates open-ended generations at scale [7], and VeriScore concentrates on claims that can actually be verified [8]. Muhlgay et al. generate benchmarks for factuality evaluation [9], Chiang and Lee examine contradictions that arise when factual claims from long generations are merged [10], FactSelfCheck brings black-box detection down to the level of individual facts [11], and Li et al. present an empirical study of factuality hallucination [12].

Taken together, these studies point to three practical qualities for a detector: fine-grained claim analysis, grounding in external evidence, and output that people can interpret. We do not present the system as a replacement for the benchmarked research methods. It is an application-oriented framework that combines these ideas in a modular verification workflow, meant to help users find the problematic claims and examine the evidence instead of receiving a bare binary label.

III. PROBLEM STATEMENT AND OBJECTIVES

An LLM answer often mixes correct and incorrect statements, and its polished style makes mistakes hard to spot by hand. Existing methods may depend on model internals, repeated sampling, classifiers tied to one benchmark, or external fact-checking resources. A practical detector should take generated text as input, pick out the factual units that need checking, fetch relevant evidence, and report its decision in a form people can follow.

The problem is therefore stated as follows: given an AI-generated response, find the factual claims that may be hallucinated and establish the status of each one using evidence retrieved from outside sources.

The detailed objectives are listed below.

- Clean the generated response and divide it into meaningful text segments.
- Pick out separately verifiable factual claims instead of handling the response as a single block.
- Extract entities, dates, places, organisations and similar details that help in searching.
- Create focused search queries for each individual claim.
- Collect pertinent evidence from web and news sources.
- Compare each claim with its evidence and label it Supported, Contradicted or Unverified.
- Compute a response-level hallucination-risk indication from the claim-level outcomes.
- Provide evidence-backed explanations that users can easily examine.
- Keep a verification history so earlier analyses can be revisited.

IV. METHODOLOGY

Hallucination detection is handled here as a staged verification task. Rather than judging factuality from the surface wording of an answer, the system first locates the passages that actually assert facts. Those assertions are turned into retrieval queries and compared with external evidence, and the final decision rests on how the claim relates to what was retrieved.

A. Input Processing

The system receives an AI-generated response. Preprocessing removes superficial formatting differences and divides the text into sentences or smaller segments suited to analysis. This step matters because one sentence can carry several factual assertions. For instance, 'The Taj Mahal is located in Agra and was completed in 1653' makes at least two propositions that can be checked separately. Treating the sentence as a unit could conceal a partial error, so the system tries to keep the original factual granularity.

B. Claim Extraction

Once the text is prepared, the system picks out the statements that convey factual information. A claim might concern people, places, organisations, historical events, dates, scientific facts, quantities, or relations between entities. Extraction does not aim to rewrite the answer; it only produces units to verify. This follows the fine-grained view of factuality represented by FActScore, in which a long generation can hold both supported and unsupported information [2].

Working at claim level helps with partly hallucinated answers. When four of five claims are supported and one is contradicted, the detector can single out that one claim instead of tagging the entire response without explanation. The report becomes more useful to anyone who wants to correct only the unreliable part.

C. Named Entity Recognition

Named Entity Recognition finds people, organisations, locations, dates and other entities that help build precise search queries. In the claim 'The Taj Mahal is located in Agra', the entities 'Taj Mahal' and 'Agra' are strong retrieval cues. Keeping names and context terms in the query reduces the number of irrelevant results.

Entity information also assists with numerical and temporal claims. Errors often involve a believable but wrong year, measurement or count, so keeping these elements in the verification query lets the retrieval step look for them directly.

D. Query Generation

A search-oriented query is created for each extracted claim. It aims to keep the meaning of the claim while stressing the entities and factual terms that are needed to find relevant evidence. The goal is not to search for the exact sentence the LLM wrote, but to locate independent information against which the assertion can be assessed.

E. Evidence Retrieval

Web evidence is obtained through the Tavily API, and the News API is called when news-related information is needed. Returned results may include titles, snippets, URLs and body text, all of which go on to the verification stage. Outside retrieval is particularly valuable for claims about recent events or facts that change over time.

Retrieval can also introduce errors. A search engine may return material that is incomplete, outdated, duplicated or of poor quality, so a retrieved result is never taken as proof in itself. The verification stage has to decide whether the retrieved material truly agrees with the claim.

F. Claim-Evidence Matching

At this stage the meaning and factual content of each claim are compared with the retrieved evidence. A claim is marked Supported when the evidence clearly agrees with it, and Contradicted when reliable evidence directly conflicts with it. When the evidence is not enough to settle either way, the claim is marked Unverified.

The three-way split is important because missing evidence must not be read as proof of falsehood. A claim that is obscure, ambiguous, recent or highly specialised can be hard to confirm even when it is true. Keeping Unverified apart from Contradicted lowers the chance of mistaking a retrieval failure for factual disproof.

G. Hallucination Scoring

Claim-level outcomes are combined into one overall hallucination-risk indication. Contradicted claims raise the risk sharply, because conflicting evidence has been found. Unverified claims also push the assessment upward, since their accuracy could not be established, while Supported claims lower the concern about the response. The resulting score is meant as a decision-support signal, not as an absolute measure of truth.

H. Explanation Generation

A key feature of the system is that it explains itself with evidence. Instead of a bare label, the output names the claim, gives the decision with its confidence or score, and cites the evidence used. This transparency lets users inspect the source themselves. It is especially useful in education and research, where people need to know why a statement was flagged before they accept or dismiss it.

V. SYSTEM ARCHITECTURE

Fig. 1 shows the overall processing flow of the system.

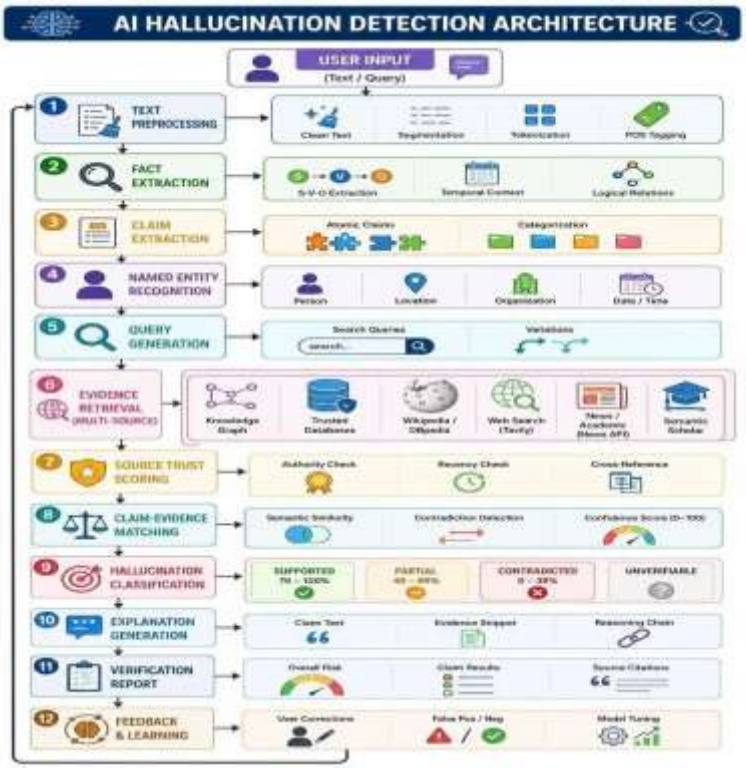


Fig. 1 System architecture of the proposed AI hallucination detection framework

The architecture is modular, so any component can be upgraded without redesigning the whole application. A stronger claim extraction model, for example, could replace the present claim-processing component while the retrieval and database layers stay as they are. In the same way, a source-credibility module could later be placed between retrieval and claim-evidence matching.

The input layer accepts an AI response produced elsewhere. Preprocessing and claim extraction convert it into verification units, and the NER component adds the entities that guide targeted queries. The retrieval module then collects external evidence, after which the verification module judges how each claim relates to its evidence. Finally, the scoring and explanation components assemble a verification report that people can read.

VI. IMPLEMENTATION

A. Backend Implementation

The backend is written in Python with a modular structure. FastAPI provides the REST endpoints through which responses are submitted and results returned. Because the user interface is kept apart from the verification services, the detector can later be connected to other AI applications. Python-based language-processing components perform preprocessing, sentence segmentation, claim identification and entity handling. Every stage hands structured data to the next, so each claim stays linked to its evidence and its final decision.

B. Claim and Entity Processing

The claim-processing module turns generated text into factual units and extracts the entities that aid retrieval, namely names of people, organisations and places, plus dates and numerical expressions. These details are carried into the search queries as context, which lowers the likelihood that a broad query returns unrelated material.

C. Evidence Retrieval

Tavily is the main web-retrieval service, with the News API as an extra source of news content when appropriate. The retrieval component gathers candidate evidence and passes it to the verification module. It does not assume that the top search result is correct; the evidence is simply input for the comparison that follows.

D. Verification and Classification

The verification component decides whether the retrieved evidence supports the claim, contradicts it, or fails to settle the matter. Output is therefore organised around three states: Supported, Contradicted and Unverified. Separating them helps most in uncertain cases, because the absence of a source does not mean the claim is false.

E. Database Implementation

MongoDB stores the application data: user details, claims, evidence, scores, verification results and history. Keeping these records lets users return to earlier analyses and gives a base for later evaluation and analytics. A stored record also preserves the link between the original claim and the evidence behind the decision.

F. Output Generation

The final report shows the original or extracted claim, its verification status, a confidence or risk indicator, and the relevant evidence. Because reporting happens per claim, users can go straight to the problem statements without rereading the whole response. The interface thus acts as a verification layer around an LLM and not as one more text generator.

VII. RESULTS AND DISCUSSION

A. Experimental Setup

The system was tested on a set of true and false AI-generated statements spanning science, history, geography, technology and general knowledge. Every statement went through the complete verification pipeline, which produced a decision and a confidence or score for each claim.

The preliminary evaluation gave an accuracy of 85%, precision of 84%, recall of 87%, F1-score of 87%, evidence coverage of 85%, and an average verification time of about five seconds. These figures reflect the current state of the project and should be treated as preliminary until the full dataset, the annotation procedure, the class distribution and

baseline comparisons have been documented.

B. Performance Metrics

Accuracy is the share of verification decisions that are correct across the tested examples. Precision shows how often claims flagged as problematic really are problematic according to the reference labels, while recall is the share of problematic claims that the system manages to catch. The F1-score is the harmonic mean of precision and recall. Evidence coverage is the proportion of tested claims for which usable evidence could be obtained, and average verification time is the end-to-end delay for a claim or response under the tested configuration.

Recall (87%) is higher than precision (84%). This means the preliminary system catches a large share of problematic examples but also issues some false positives. Such a pattern is to be expected with evidence retrieval, since ambiguous claims, weak sources and incomplete retrieval can all disturb classification.

C. Sample Verification Results

Fig. 2 presents representative examples from the current system.

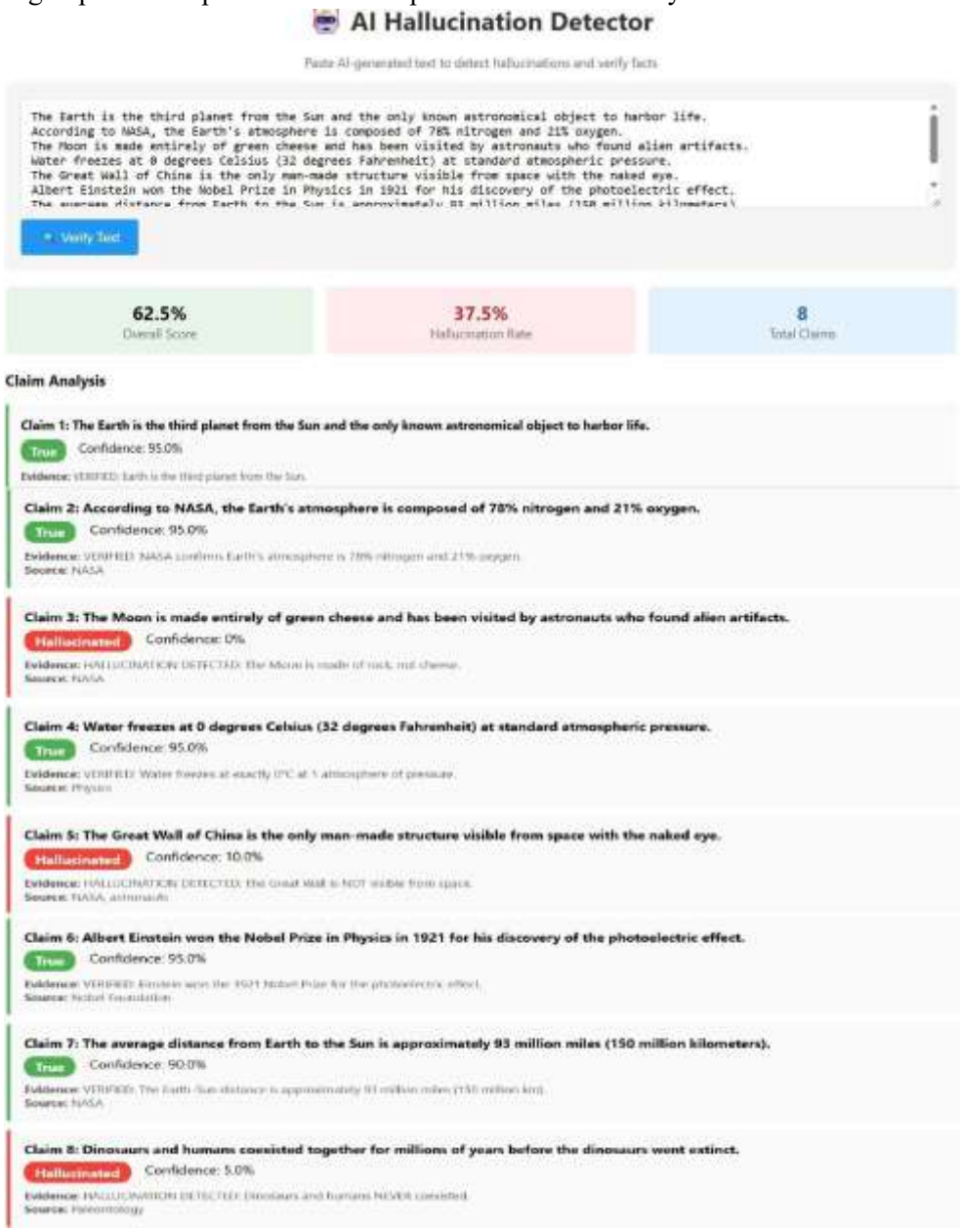


Fig. 2 AI hallucination detection and claim-level verification results

These examples show the basic behaviour of the detector, but they cannot serve as sufficient evidence of general performance, since several of them are deliberately obvious. A more demanding evaluation should add subtle errors, responses that mix true and false content, time-dependent and numerical claims, ambiguous statements, and domain-

specific cases. Benchmarks such as HaluEval or RAGTruth would give a firmer basis for future comparison [4], [5].

D. Claim-Level Analysis

A main strength of the design is its handling of mixed responses. Imagine an answer with five factual claims, four supported and one carrying a wrong date. A response-level classifier would call the whole answer correct or hallucinated and hide the actual error, whereas the claim-level approach isolates the wrong date and ties it to contradicting or insufficient evidence.

This agrees with the motivation behind FActScore, which measures factuality through atomic facts and not through one coarse judgment [2]. Here the same principle yields an actionable report, letting users concentrate on the claims that need further checking.

E. Evidence and Transparency

Showing external evidence makes it possible to explain why a claim received its label. The evidence itself, however, has to be read with care. Retrieval quality, source authority, publication date and agreement among sources all affect how reliable a result is. This echoes the finding from RAGTruth that retrieval-augmented systems can still produce unsupported or contradictory claims [4]. Evidence retrieval should therefore be regarded as an aid to verification, not a guarantee of truth.

F. Error Cases and Limitations

The system has trouble with claims that are ambiguous, highly specialised, outdated, fast-changing, or stated without enough context. A claim about a current event, for example, may become wrong as news develops, and a historical claim may be described differently by different sources. A search API may also miss the best source even when an authoritative one exists.

A further limitation is that semantic agreement between a claim and a passage does not necessarily mean the passage entails the claim. A passage may mention the same entities while describing another event or period. Future versions should therefore strengthen entailment and contradiction analysis rather than depending only on broad semantic similarity.

G. Discussion

The preliminary results suggest that pairing claim-level processing with external evidence gives a helpful verification layer for generated text. It works best for plainly factual statements whose evidence can be retrieved reliably, and it becomes harder to use when the evidence is incomplete or conflicting.

The findings also indicate that hallucination detection should not be collapsed into one model score. A useful detector should communicate uncertainty and expose its evidence, which makes results easier to interpret and lets users check decisions on their own. Future evaluations should accordingly measure evidence quality, confidence calibration, retrieval success, latency and robustness across domains, in addition to classification performance.

VIII. CONCLUSION

This paper introduced an evidence-grounded system for assessing the factual reliability of LLM-generated text. It brings together text preprocessing, claim extraction, named entity recognition, query generation, external evidence retrieval, claim-evidence matching, hallucination scoring and explanation generation. In contrast to a plain response-level classifier, it evaluates factual claims one at a time, so a response can contain both supported and problematic statements without losing that distinction.

The implementation shows that hallucination verification can be built into a practical application based on Python and FastAPI. Initial testing gave 85% accuracy, 84% precision, 87% recall and an 87% F1-score, along with 85% evidence coverage and about five seconds of average verification time. These numbers are encouraging, but they also underline the need for larger and more rigorous benchmark-based evaluation.

The work builds on established research on fine-grained factual evaluation, black-box consistency checking, hallucination benchmarks and retrieval-based verification [1]-[5]. It carries those ideas into a modular, user-facing verification pipeline that puts evidence and interpretability first.

IX. FUTURE WORK

- Test the detector on larger public hallucination benchmarks, including HaluEval and RAGTruth.
- Add explicit source-credibility assessment so that authoritative sources carry more weight in verification.
- Strengthen claim-to-evidence entailment and contradiction detection with more capable natural language inference models.
- Introduce temporal verification so that time-sensitive claims are checked against publication dates and current evidence.
- Develop calibration methods so that confidence scores better reflect the probability that a decision is correct.
- Add multi-source agreement analysis to reduce reliance on any single retrieved document.
- Extend the system to multilingual and domain-specific hallucination detection.
- Explore multimodal hallucination detection for content in images, charts and other generated media.
- Support automatic citation generation together with checking of citation correctness.
- Measure latency and retrieval cost for different query sizes and source configurations.
- Compare the pipeline with simple semantic-similarity, retrieval-only and black-box consistency baselines.

REFERENCES

[1] Z. Ji, N. Lee, R. M. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung, "Survey of Hallucination in Natural Language Generation," ACM Computing Surveys, vol. 55, no. 12, Article 248, pp. 1–38, 2023, doi: 10.1145/3571730.

[2] S. Min, K. Krishna, X. Lyu, M. Lewis, W.-t. Yih, P. Koh, M. Iyyer, L. Zettlemoyer, and H. Hajishirzi, "FActScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation," in Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pp. 12076–12100, 2023, doi: 10.18653/v1/2023.emnlp-main.741.

[3] P. Manakul, A. Liusie, and M. Gales, "SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models," in Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pp. 9004–9017, 2023, doi: 10.18653/v1/2023.emnlp-main.557.

[4] C. Niu, Y. Wu, J. Zhu, S. Xu, K. Shum, R. Zhong, J. Song, and T. Zhang, "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), pp. 10862–10878, 2023, doi: 10.18653/v1/2023.acl-long.585.

[5] J. Li, X. Cheng, X. Zhao, J.-Y. Nie, and J.-R. Wen, "HaluEval: A Large-Scale Hallucination Evaluation Benchmark for Large Language Models," in Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pp. 6449–6464, 2023, doi: 10.18653/v1/2023.emnlp-main.397.

[6] S. Lin, J. Hilton, and O. Evans, "TruthfulQA: Measuring How Models Mimic Human Falsehoods," in Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 3214–3252, 2023, doi: 10.18653/v1/2023.acl-long.229.

[7] J. Wei, C. Yang, X. Song, Y. Lu, N. Hu, J. Huang, D. Tran, D. Peng, R. Liu, D. Huang, C. Du, and Q. V. Le, "Long-form factuality in large language models," arXiv preprint arXiv:2403.18802, 2024.

[8] Y. Song, Y. Kim, and M. Iyyer, "VeriScore: Evaluating the factuality of verifiable claims in long-form text generation," in Findings of the Association for Computational Linguistics: EMNLP 2024, pp. 9447–9474, 2024, doi: 10.18653/v1/2024.findings-emnlp.552.

[9] D. Muhlgay, O. Ram, I. Magar, Y. Levine, N. Ratner, Y. Belinkov, O. Abend, K. Leyton-Brown, A. Shashua, and Y. Shoham, "Generating Benchmarks for Factuality Evaluation of Language Models," in Proceedings of the 18th

Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 49–66, 2024, doi: 10.18653/v1/2024.eacl-long.4.

[10] C.-H. Chiang and H.-y. Lee, "Merging Facts, Crafting Fallacies: Evaluating the Contradictory Nature of Aggregated Factual Claims in Long-Form Generations," in Findings of the Association for Computational Linguistics: ACL 2025, pp. 2734–2751, 2025, doi: 10.18653/v1/2025.findings-acl.160.

[11] A. Sawczyn, J. Binkowski, D. Janiak, B. Gabrys, and T. J. Kajdanowicz, "FactSelfCheck: Fact-Level Black-Box Hallucination Detection for LLMs," in Findings of the Association for Computational Linguistics: EACL 2026, pp. 5603–5621, 2026, doi: 10.18653/v1/2026.findings-eacl.296.

[12] J. Li, J. Chen, R. Ren, X. Cheng, X. Zhao, J.-Y. Nie, and J.-R. Wen, "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), pp. 10879–10899, 2026, doi: 10.18653/v1/2026.acl-long.586.