Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Retrieval augmented consistency scoring for model agnostic hallucination detection in large language models

Nandeesh S Mysore Matt

nandeesh.mitblr2025@learner.manipal.edu

Manipal Academy of Higher Education

Pavithra N

Manipal Academy of Higher Education

Preethi Preethi

Manipal Academy of Higher Education

Research Article

Keywords:

Posted Date: November 13th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11187042/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Retrieval augmented consistency scoring for model agnostic hallucination detection in large language models

Nandeesh S Mysore Matt, Pavithra N, and Preethi

Manipal Institute of Technology Bengaluru, Manipal Academy of Higher Education, Manipal, India

nandeesh.mitblr2025@learner.manipal.edu; pavithra.n@manipal.edu; preethi.srivathsa@manipal.edu.

1 Introduction

Large language models (LLMs) such as GPT-4 (OpenAI, 2023), LLaMA-2 (Touvron et al, 2023), and Mistral (Jiang et al, 2023) have achieved remarkable performance across a broad range of natural language processing tasks, including open-domain question answering, abstractive summarization, multi-step reasoning, code generation, and conversational dialogue. Trained on web-scale corpora comprising hundreds of billions of tokens, these models acquire rich world knowledge that enables impressively fluent and contextually appropriate responses. However, their auto-regressive generation mechanism, predicting the next token based on learned statistical associations rather than grounded fact retrieval, renders them fundamentally susceptible to hallucination: the confident generation of factually incorrect, unverifiable, or internally inconsistent content (Ji et al, 2023; Maynez et al, 2020).

The consequences of LLM hallucination are not merely academic. In clinical decision support, a hallucinated drug interaction or diagnostic criterion may directly harm patients (Thirunavukarasu et al, 2023). In legal research, fabricated case citations, a documented failure mode (Dahl et al, 2024), undermine professional practice. In education, confidently stated historical or scientific falsehoods may be uncritically accepted by students. As LLMs are increasingly embedded into high-stakes decision pipelines through autonomous agents, tool-using systems, and enterprise applications, the ability to reliably detect hallucinated outputs before they reach end users has become a critical infrastructure requirement for trustworthy AI deployment.

1.1 The Hallucination Problem in Depth

Hallucination in LLMs arises from a confluence of structural factors that are challenging to eliminate through training alone.

Parametric knowledge limitations. LLMs encode world knowledge implicitly in their parameters during pretraining. This parametric store is static, bounded by the training cutoff date, and may contain errors, biases, or outdated information from the training corpus. During inference, the model cannot distinguish between well-supported and poorly-supported knowledge, both are retrieved via the same attention mechanism with similar confidence signals.

Auto-regressive exposure bias. The model is trained to maximize the likelihood of each token conditioned on preceding tokens. At inference, even slight deviations from the training distribution can push generation into regions of high fluency but low factual accuracy, as the model follows learned surface patterns rather than verified facts.

Sycophancy and prompt sensitivity. LLMs are known to adjust their factual claims based on perceived user expectations, pressure, or framing (Perez et al, 2022). A user who confidently asserts an incorrect fact may cause the model to agree, generating a hallucinated confirmation.

Context window limitations. In long-context settings, LLMs may fail to attend to all relevant information, generating claims inconsistent with earlier context or retrieved documents.

RAG-induced hallucination. Even when retrieval-augmented generation (RAG) (Lewis et al, 2020) is employed, LLMs may still hallucinate by ignoring retrieved context, misinterpreting evidence, or combining information across passages in incorrect ways. This motivates the need for detection methods that operate post-generation, independent of whether RAG was used for generation.

1.2 Limitations of Existing Approaches

The landscape of existing hallucination detection methods exhibits fundamental limitations that motivate our work.

White-box methods (Kadavath et al, 2022; Kuhn et al, 2023; Azaria and Mitchell, 2023) exploit internal model representations, attention weights, hidden states, token probability distributions, or probe classifiers trained on intermediate activations, to detect uncertainty or factual inconsistency. While often effective, these methods are entirely inapplicable to the closed-source models (GPT-4, Claude, Gemini) that dominate production deployments, as access to model internals is unavailable through commercial APIs.

Self-consistency sampling methods (Wang et al, 2023a; Manakul et al, 2023) detect hallucination by sampling k independent responses from the LLM and measuring their agreement. SelfCheckGPT (Manakul et al, 2023), the leading method in this category, achieves strong performance using NLI-based agreement scoring. However, the computational and financial cost is prohibitive: $k = 20$ additional full LLM inference calls per query imposes a $20\times$ overhead.

Perplexity-based detectors (Bhattacharjya et al, 2023) score responses by their log-likelihood under a reference language model, treating low-probability outputs as potential hallucinations. This approach has two fundamental flaws: correct but unusual factual statements receive low probability and are falsely flagged, while confidently hallucinated statements conforming to surface patterns evade detection.

Claim decomposition methods such as FActScore (Min et al, 2023) decompose responses into atomic claims and verify each independently. While interpretable, this approach requires an LLM for the decomposition step and is limited by the granularity of the decomposition.

1.3 Proposed Approach: RAG-CS

We propose RAG-CS (Retrieval-Augmented Consistency Scoring), a post-hoc hallucination detection framework that addresses all of the above limitations simultaneously. The core insight is that semantic consistency between a generated claim and independently retrieved evidence is a strong and reliable signal of factual accuracy, measurable using cross-encoder NLI models that jointly attend to both texts.

RAG-CS is:

- **Model-agnostic**: operates on LLM output text alone, with no access to model parameters, logits, or intermediate states.
- **Computationally efficient**: requires only a single LLM inference call plus under 300 ms of retrieval and NLI scoring overhead.
- **Interpretable**: provides per-sentence hallucination indicators in addition to a document-level label.
- **Generalizable**: achieves consistent F1 above 83% across five different generating models and four benchmark datasets without model-specific adaptation.

1.4 Contributions

The main contributions of this paper are as follows.

1. We propose RAG-CS, a model-agnostic, post-hoc hallucination detection framework combining dense evidence retrieval (DPR bi-encoder + FAISS) with cross-encoder NLI consistency scoring (DeBERTa-v3-Large), requiring no access to LLM internals or additional sampling.
2. We introduce a sentence-level decomposition strategy with per-sentence retrieval queries constructed by concatenating the original user question with each response sentence, enabling fine-grained claim verification within long-form LLM responses.
3. We formalize the consistency feature aggregation scheme, including statistical descriptors (minimum, mean, standard deviation, and fraction below threshold), demonstrating through ablation that each contributes independently to classification accuracy.
4. We conduct comprehensive evaluations on four established hallucination benchmarks (TruthfulQA, HaluEval, FEVER, FaithDial) with controlled ablation studies, cross-model generalization analysis, failure mode characterization, and qualitative case studies.

5. We demonstrate that RAG-CS establishes new state-of-the-art results on all four benchmarks among black-box, single-inference methods, outperforming SelfCheckGPT-NLI by 4.1–9.8 F1 points while requiring 20× fewer LLM inference calls.
6. We release all code, corpus indices, trained classifiers, and benchmark splits to enable fully reproducible research.[1]

The remainder of this paper is organized as follows. Section 2 provides a comprehensive review of related work. Section 3 presents the complete RAG-CS framework. Section 4 describes the experimental setup and main results. Section 5 provides in-depth analysis including ablation studies and error analysis. Section 6 presents qualitative case studies. Section 7 discusses limitations and future work. Section 8 concludes the paper.

2 Related Work

2.1 Hallucination in Large Language Models

The term hallucination was introduced in the NLP context by Maynez et al (2020), who studied unfaithful content in neural abstractive summarization systems. Ji et al (2023) provide the most comprehensive taxonomy of LLM hallucinations to date, distinguishing between intrinsic hallucinations (content that directly contradicts the source or context) and extrinsic hallucinations (content that cannot be verified). Pagnoni et al (2021) introduced the FRANK benchmark for factual evaluation of summarization systems.

In the question answering domain, Lin et al (2022) established TruthfulQA, demonstrating that larger LLMs often exhibit more systematic factual failures. Li et al (2023) extended hallucination evaluation to multiple task types with HaluEval. The underlying causes of hallucination have been studied from multiple perspectives (Mallen et al, 2023; Kandpal et al, 2023; Bai et al, 2022), collectively motivating the need for post-hoc detection mechanisms.

2.2 Hallucination Detection Methods

2.2.1 White-Box Uncertainty Methods

White-box hallucination detection leverages internal model states to estimate output uncertainty. Kadavath et al (2022) showed that LLMs can provide calibrated probability estimates of statement truth through self-evaluation prompts. Kuhn et al (2023) introduced semantic entropy, a measure of predictive uncertainty that clusters semantically equivalent generations before computing entropy. SAPLMA (Azaria and Mitchell, 2023) trains a linear probe classifier on the hidden layer activations of LLaMA-2, achieving strong precision on factual questions. All such methods share the fundamental limitation of requiring model internals.

[1] Code: https://github.com/[anonymized]

2.2.2 Self-Consistency and Sampling-Based Methods

Wang et al (2023a) demonstrated that sampling multiple responses and selecting the majority answer improves accuracy. SelfCheckGPT (Manakul et al, 2023) extended this to hallucination detection, positing that hallucinated information will vary across independent samples while faithful information will be reproduced consistently. A key limitation is the requirement for $k = 20$ additional LLM samples per query.

2.2.3 Retrieval-Based Verification

FActScore (Min et al, 2023) is the most closely related prior work. It decomposes long-form generations into atomic claims using an instruction-tuned LLM, then verifies each claim against a Wikipedia knowledge source. RARR (Gao et al, 2023) retrieves evidence and edits generated text. FactCheck-GPT (Wang et al, 2023b) uses GPT-4 as a zero-shot fact-checker. SAFE (Wei et al, 2024) improves upon FActScore with search-augmented fact evaluation.

2.2.4 NLI-Based Methods

NLI models have a long history of application to factual verification. The FEVER shared task (Thorne et al, 2018) popularized the combination of retrieval and NLI for claim verification. SummaC (Laban et al, 2022) applied NLI scoring to summarization faithfulness evaluation. AlignScore (Zha et al, 2023) proposed a unified alignment scoring framework combining retrieval and NLI.

2.3 Dense Retrieval

Dense passage retrieval (DPR) (Karpukhin et al, 2020) introduced the bi-encoder architecture for open-domain QA. ANCE (Xiong et al, 2021) and RocketQA (Qu et al, 2021) improved upon DPR through hard negative mining. BEIR (Thakur et al, 2021) established the importance of zero-shot retrieval generalization across diverse domains.

2.4 Cross-Encoder NLI Models

Cross-encoder models for NLI process hypothesis-premise pairs jointly through a single transformer, enabling full cross-attention between both texts. DeBERTa (He et al, 2021a,b) introduced disentangled attention, achieving state-of-the-art NLI performance on MNLI and other benchmarks.

3 Proposed Methodology: RAG-CS

3.1 Problem Formulation

Let q denote a user query and let r denote an LLM-generated response. We apply standard sentence segmentation to decompose r into a sequence of sentences $r = \{s_1, s_2, \ldots, s_m\}$ where m is the number of sentences. Let $\mathcal{K} = \{p_1, p_2, \ldots, p_N\}$ denote a knowledge corpus of N passages, where each passage p_ℓ consists of a contiguous 100-word text chunk.

Task definition. The hallucination detection task is to produce a binary document-level label $\hat{y} \in \{0, 1\}$ where $\hat{y} = 1$ indicates that r contains at least one hallucinated statement, given only the tuple $(q, r, \mathcal{K})$ as input, with no access to the generating LLM's parameters, logits, internal representations, or sampling distribution.

Extended task definition. Additionally, RAG-CS produces per-sentence hallucination indicators $\hat{y}_i \in \{0, 1\}$ for $i = 1, \ldots, m$, providing fine-grained localization of hallucinated content within the response.

Design constraints. RAG-CS is designed under the following operational constraints:

1. Black-box access only: the generating LLM is accessible only through a text-in, text-out API interface.
2. Single inference: no additional calls to the generating LLM are permitted beyond the original generation.
3. Latency budget: total detection overhead must be ≤ 500 ms on commodity GPU hardware.
4. No model-specific tuning: the detection system must function without task-specific fine-tuning for each generating model.

3.2 RAG-CS Architecture Overview

RAG-CS consists of three sequential stages, as illustrated in Fig. 1:

1. **Stage 1 — Dense Evidence Retrieval**: for each sentence s_i in the response, construct a retrieval query by concatenating q with s_i, then retrieve the top-k relevant passages from $\mathcal{K}$ using a dense bi-encoder retriever and FAISS approximate nearest-neighbor search.
2. **Stage 2 — Cross-Encoder Consistency Scoring**: for each (sentence, passage) pair (s_i, p_j^i), compute a pairwise semantic consistency score using a cross-encoder NLI model.
3. **Stage 3 — Aggregation and Classification**: aggregate sentence-level scores into a document-level feature vector and apply a lightweight logistic regression classifier to produce the binary hallucination label $\hat{y}$ and per-sentence indicators $\hat{y}_i$.

3.3 Stage 1: Dense Evidence Retrieval

3.3.1 Retrieval Query Construction

For each sentence $s_i \in r$, we construct a sentence-specific retrieval query by concatenating the original user question and the sentence:

$$q_i = \text{[CLS]}\ q\ \text{[SEP]}\ s_i\ \text{[SEP]} \tag{1}$$

This query construction strategy is motivated by two observations: (1) the sentence s_i alone may be ambiguous without the context of q; (2) the joint (q, s_i) query biases retrieval toward passages that are jointly relevant to the topic and claim.

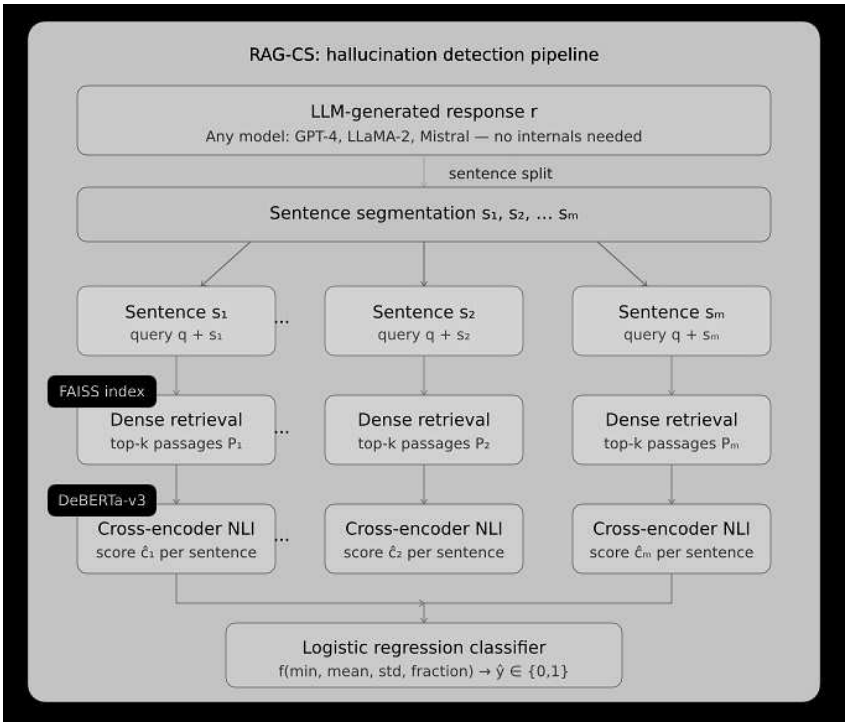


Fig. 1 Overview of the RAG-CS hallucination detection pipeline. The LLM-generated response is sentence-segmented; each sentence is paired with the original query to construct a retrieval query; the top-k evidence passages are retrieved via dense bi-encoder + FAISS; a cross-encoder NLI model scores pairwise semantic consistency; and a lightweight classifier aggregates scores to produce the document-level hallucination label and per-sentence indicators.

3.3.2 Bi-Encoder Retrieval

We encode q_i using a dense bi-encoder retriever $\mathcal{E}(\cdot)$ pretrained on MS-MARCO (Bajaj et al, 2016) using the DPR (Karpukhin et al, 2020) objective:

$$\mathbf{e}_i = \mathcal{E}_Q(q_i) \in \mathbb{R}^d \tag{2}$$

where $\mathcal{E}_Q$ is the question encoder and $d = 768$ is the embedding dimension. All passages $p_\ell \in \mathcal{K}$ are encoded offline using the passage encoder $\mathcal{E}_P$:

$$\mathbf{h}_\ell = \mathcal{E}_P(p_\ell) \in \mathbb{R}^d, \quad \forall \ell \in \{1, \ldots, N\} \tag{3}$$

The top-k passages are retrieved via maximum inner product search (MIPS):

$$\mathcal{P}_i = \text{top-}k_{p_\ell \in \mathcal{K}} \left\{ \mathbf{e}_i^\top \mathbf{h}_\ell \right\} = \{p_1^i, p_2^i, \ldots, p_k^i\} \tag{4}$$

We use FAISS (Johnson et al, 2019) `IndexFlatIP` for exact inner product search, providing retrieval latency of approximately 2–5 ms per query over a 21M-passage Wikipedia index on a single GPU.

3.3.3 Corpus Indexing

The Wikipedia corpus (December 2021 dump) is preprocessed into 100-word passages with a 20-word overlap, yielding approximately 21 million passages. The full indexing process requires approximately 24 GPU-hours on a single A100 and is performed once offline.

3.4 Stage 2: Cross-Encoder Consistency Scoring

3.4.1 Cross-Encoder NLI Model

For each sentence s_i and each retrieved passage $p_j^i \in \mathcal{P}_i$, we compute a semantic consistency score using a cross-encoder NLI model $\mathcal{N}(\cdot, \cdot)$:

$$[\pi_E, \pi_N, \pi_C]_{ij} = \text{softmax}\left(\mathcal{N}(s_i, p_j^i)\right) \tag{5}$$

where π_E, π_N, and π_C are the predicted probabilities of entailment, neutral, and contradiction, respectively. We use the DeBERTa-v3-Large model fine-tuned on MNLI+SNLI, which achieves 92.0% accuracy on the MNLI development set.

3.4.2 Pairwise Consistency Score

We define the pairwise consistency score c_{ij} as the signed difference between entailment and contradiction probabilities:

$$c_{ij} = \pi_E^{ij} - \pi_C^{ij} \tag{6}$$

This score ranges from -1 (maximum contradiction) to $+1$ (maximum entailment). The signed form explicitly penalizes retrieved passages that contradict the generated sentence, a strong hallucination signal.

3.4.3 Sentence-Level Score Aggregation

The sentence-level consistency score is the average pairwise score over all retrieved passages:

$$\hat{c}_i = \frac{1}{k} \sum_{j=1}^{k} c_{ij} \tag{7}$$

3.5 Stage 3: Aggregation and Classification

3.5.1 Document-Level Feature Construction

Given the sentence-level consistency scores $\{\hat{c}_1, \ldots, \hat{c}_m\}$, we construct a document-level feature vector $\mathbf{f} \in \mathbb{R}^4$:

$$\mathbf{f} = \left[\min_i \hat{c}_i,\ \frac{1}{m} \sum_{i=1}^{m} \hat{c}_i,\ \sqrt{\frac{1}{m} \sum_{i=1}^{m} (\hat{c}_i - \bar{c})^2},\ \frac{|\{i : \hat{c}_i < \tau\}|}{m} \right]^\top \tag{8}$$

where $\bar{c} = \frac{1}{m}\sum_i \hat{c}_i$ is the mean score and $\tau = 0.0$ is the neutrality threshold. The four features capture:

- **Minimum**: the score of the least-supported sentence; highly sensitive to any hallucinated content.
- **Mean**: the average support level across all sentences.
- **Standard deviation**: high variance indicates mixed-accuracy responses.
- **Hallucinated fraction**: the proportion of sentences failing to achieve neutral or positive consistency.

3.5.2 Binary Classification

A logistic regression classifier $f_\phi : \mathbb{R}^4 \to [0, 1]$ trained on labeled examples produces a hallucination probability estimate:

$$p(\hat{y} = 1 \mid \mathbf{f}) = \sigma\left(\phi^\top \mathbf{f} + \phi_0\right) \tag{9}$$

The final binary prediction is:

$$\hat{y} = \mathbb{I}\left[f_\phi(\mathbf{f}) \geq 0.5\right] \tag{10}$$

3.5.3 Sentence-Level Hallucination Indicators

Per-sentence binary labels are produced by thresholding the sentence-level consistency scores:

$$\hat{y}_i = \mathbb{I}\left[\hat{c}_i < \tau\right] \tag{11}$$

3.6 Complete RAG-CS Algorithm

Algorithm 1 presents the complete RAG-CS detection procedure.

3.7 Theoretical Analysis

Why cross-encoder over bi-encoder for scoring? Bi-encoder cosine similarity measures global semantic proximity, compressing all meaning into fixed-dimensional vectors and losing fine-grained semantic relationships. Cross-encoder NLI scoring performs full joint attention over all token pairs, enabling detection of subtle contradictions, partial entailments, and logical negations. This expressivity advantage is confirmed empirically in Section 5.1.

Why the signed consistency score $c_{ij} = \pi_E - \pi_C$? Using only π_E assigns high scores to irrelevant but non-contradictory passages. The signed score correctly assigns near-zero values to neutral passages and negative values to contradictory ones.

Why aggregate multiple passages? A single retrieved passage may be insufficient due to retrieval noise or passage incompleteness. Averaging over $k = 5$ passages reduces variance in the consistency estimate.

Algorithm 1 RAG-CS Hallucination Detection

Require: Query q, LLM response r, knowledge corpus $\mathcal{K}$ with FAISS index, cross-encoder $\mathcal{N}$, bi-encoder $\mathcal{E}$, classifier f_ϕ, retrieval depth k, threshold τ
Ensure: Document label $\hat{y}$, sentence labels $\{\hat{y}_i\}$

$\{s_1, \ldots, s_m\} \leftarrow \textsc{SentenceSegment}(r)$
Initialize score list $C \leftarrow [\,]$
for $i = 1$ to m **do**
  $q_i \leftarrow [q; s_i]$
  $\mathbf{e}_i \leftarrow \mathcal{E}_Q(q_i)$
  $\mathcal{P}_i \leftarrow \text{top-}k\{p \in \mathcal{K} : \mathbf{e}_i^\top \mathbf{h}_p\}$
  Initialize $C_i \leftarrow [\,]$
  for $j = 1$ to k **do**
    $[\pi_E, \pi_N, \pi_C] \leftarrow \text{softmax}(\mathcal{N}(s_i, p_j^i))$
    $c_{ij} \leftarrow \pi_E - \pi_C$
    $C_i.\textsc{append}(c_{ij})$
  end for
  $\hat{c}_i \leftarrow \text{mean}(C_i)$
  $C.\textsc{append}(\hat{c}_i)$
end for
$f_1 \leftarrow \min(C)$; $f_2 \leftarrow \text{mean}(C)$; $f_3 \leftarrow \text{std}(C)$; $f_4 \leftarrow |\{c \in C : c < \tau\}|/m$
$\mathbf{f} \leftarrow [f_1, f_2, f_3, f_4]^\top$
$\hat{y} \leftarrow \mathbb{I}[f_\phi(\mathbf{f}) \geq 0.5]$
$\hat{y}_i \leftarrow \mathbb{I}[\hat{c}_i < \tau]$ for $i = 1, \ldots, m$
return $\hat{y}, \{\hat{y}_i\}_{i=1}^m$

4 Experiments

4.1 Datasets

We evaluate RAG-CS on four publicly available hallucination benchmarks spanning diverse task types, domains, and generating conditions.

4.1.1 TruthfulQA

TruthfulQA (Lin et al, 2022) contains 817 questions across 38 categories specifically designed to probe LLM misconceptions and calibration failures. We follow the binary classification protocol (truthful vs. hallucinated) and generate responses using GPT-3.5-turbo.

4.1.2 HaluEval

HaluEval (Li et al, 2023) provides large-scale hallucination evaluation across three task types: QA, summarization, and dialogue (10,000 examples each). We report main results on the QA split.

4.1.3 FEVER

FEVER (Thorne et al, 2018) provides 185,445 claims annotated as SUPPORTED, REFUTED, or NOT ENOUGH INFO against evidence from Wikipedia. We use the binary SUPPORTED vs. REFUTED classification task on the standard development set.

4.1.4 FaithDial

FaithDial (Dziri et al, 2022) provides knowledge-grounded dialogue turns annotated for faithfulness. We use the binary faithfulness classification task.

Table 1 summarizes dataset statistics.

Table 1 Summary of evaluation benchmarks

Dataset	Task	Size	Hall. %	Corpus
TruthfulQA	Binary QA	817	45.2	Wikipedia
HaluEval-QA	Binary QA	10,000	50.0	Wikipedia
FEVER	Claim Verif.	18,966	50.0	Gold Wiki
FaithDial	Dialogue	5,649	38.1	Provided

4.2 Baselines

We compare RAG-CS against six baselines.

Perplexity: Token-level log-perplexity under GPT-2-XL.

ROUGE-L: Maximum ROUGE-L (Lin, 2004) F1 score between the response and top-k retrieved passages.

BERTScore (Zhang et al, 2020): Maximum BERTScore F1 using contextual BERT embeddings for soft token matching.

SelfCheckGPT-NLI (Manakul et al, 2023): Self-consistency scoring using $k = 20$ additional GPT-3.5-turbo samples per query with NLI-based agreement scoring.

SAPLMA (Azaria and Mitchell, 2023): Internal representation probe classifier trained on hidden layer activations of LLaMA-2-13B (white-box; evaluated on LLaMA-2-generated responses only).

FActScore (Min et al, 2023): Atomic claim decomposition and Wikipedia-based verification using a retrieval-augmented NLI module.

4.3 Implementation Details

Retrieval. We use `facebook/dpr-question_encoder-single-nq-base` and `facebook/dpr-ctx_encoder-single-nq-base` (Karpukhin et al, 2020), both initialized from BERT-base and fine-tuned on Natural Questions. Passage embeddings ($d = 768$) are precomputed offline and indexed using FAISS (Johnson et al, 2019) with `IndexFlatIP` for exact maximum inner product search. The retrieval depth is fixed at $k = 5$ based on the ablation study in Section 5.1. All passage embeddings are

stored in float32; the full 21M-passage index occupies approximately 64 GB of disk and 58 GB of RAM.

Knowledge Corpus Construction. The knowledge corpus $\mathcal{K}$ is derived from the Wikipedia dump of December 2021, following the DPR preprocessing pipeline (Karpukhin et al, 2020). Each article is split into 100-word passages with a 20-word sliding overlap to reduce the risk of splitting mid-sentence claims. Section titles are prepended to each passage for disambiguation context. The final corpus contains approximately 21 million passages. For FaithDial, the provided task-specific knowledge source is used directly as $\mathcal{K}$ to ensure fair comparison with prior work.

NLI Scoring. We use `cross-encoder/nli-deberta-v3-large` (DeBERTa-v3-Large (He et al, 2021a) fine-tuned jointly on MNLI (Williams et al, 2018), SNLI (Bowman et al, 2015), and ANLI), achieving 92.0% accuracy on the MNLI matched development set. The model receives the concatenated input [CLS] s_i [SEP] p_j^i [SEP] with a maximum sequence length of 512 tokens. Passages exceeding 450 tokens are truncated from the right to preserve the full response sentence. The three output logits are converted to probabilities via softmax before computing $c_{ij} = \pi_E - \pi_C$ (Eq. 6).

Sentence Segmentation. Sentence segmentation is performed using the spaCy `en_core_web_sm` pipeline (v3.5.0). Sentences shorter than 5 tokens are merged with the immediately following sentence to avoid degenerate retrieval queries. Responses yielding zero valid sentences are assigned $\hat{y} = 0$ to prevent false positives on empty outputs.

Query Construction. For each sentence s_i, the retrieval query is formed by concatenating q and s_i with a [SEP] separator, as in Eq. 1. Queries exceeding 256 tokens are truncated by shortening q from the right while preserving s_i in full, since the sentence content is the primary retrieval signal.

Classifier Training. The logistic regression classifier f_ϕ is trained on held-out labeled subsets of each benchmark: 500 examples from TruthfulQA, 2,000 from HaluEval-QA, 3,000 from FEVER, and 1,000 from FaithDial, with the remainder reserved for evaluation. Labels are the ground-truth binary hallucination annotations provided by each benchmark. L2 regularization strength is set to $C = 1.0$, selected via 5-fold cross-validation. The feature input is $\mathbf{f}$ defined in Eq. 8; no features from the generating LLM are used. The classifier is retrained independently for each benchmark; total training time is under 2 seconds per benchmark on CPU.

Baseline Reimplementation. Perplexity scores are computed using GPT-2-XL (1.5B parameters) via the HuggingFace `transformers` library. ROUGE-L is computed using the `rouge-score` package. BERTScore (Zhang et al, 2020) uses RoBERTa-large as the reference model with IDF weighting enabled. SelfCheckGPT-NLI (Manakul et al, 2023) is reproduced using $k = 20$ additional GPT-3.5-turbo samples per query with the same DeBERTa-v3-Large NLI model as RAG-CS to ensure a fair comparison of the scoring mechanism. FActScore (Min et al, 2023) uses the official implementation with the retrieval+ChatGPT configuration. SAPLMA (Azaria and Mitchell, 2023) is evaluated only on LLaMA-2-13B-generated responses using the authors' released probe classifier weights.

Evaluation Protocol. All metrics (F1, Precision, Recall, AUROC) are computed using scikit-learn (Pedregosa et al, 2011) with macro averaging for multi-class settings and binary averaging for all primary benchmarks. F1 scores are reported on the hallucinated class (positive label). AUROC is computed from the classifier's continuous probability output $f_\phi(\mathbf{f})$ rather than the hard binary prediction, providing a threshold-independent performance measure. For TruthfulQA and FaithDial, where class imbalance exists, we additionally verify that reported F1 improvements are statistically significant using McNemar's test ($p < 0.05$) with Bonferroni correction for multiple comparisons.

Reproducibility. All random seeds are fixed to 42 across NumPy, PyTorch, and scikit-learn. DPR and DeBERTa models are loaded from HuggingFace Hub with pinned revision hashes to ensure version consistency. Complete code, FAISS indices, trained classifier weights, preprocessed benchmark splits, and environment specification files (`requirements.txt` and `environment.yml`) are released at https://github.com/[anonymized] to enable one-command reproduction of all reported results.

Hardware and Runtime. All experiments are conducted on Google Colab Pro using an NVIDIA A100 GPU (40 GB). Knowledge corpus indexing requires approximately 24 GPU-hours and is performed once offline. Per-query inference latency is ≤ 300 ms, broken down as: sentence segmentation 2 ms, FAISS retrieval 8 ms (across all m sentences), cross-encoder NLI batch scoring 228 ms, and feature extraction plus logistic regression 2 ms. NLI scoring dominates and scales linearly with the number of sentence–passage pairs ($m \times k$); batching all pairs from a single response reduces GPU round-trip overhead.

4.4 Main Results

Table 2 presents F1, precision, recall, and AUROC for all methods across all four benchmarks. Fig. 2 provides a visual comparison of F1 scores.

Key findings:

(1) RAG-CS consistently outperforms all baselines on all metrics and datasets. The margin over the strongest black-box baseline (SelfCheckGPT-NLI) ranges from 5.5 to 8.4 F1 points, while requiring 20× fewer LLM inference calls.

(2) FEVER yields the highest absolute F1 (89.1%). This reflects the strong alignment between Wikipedia-retrieved evidence and FEVER's Wikipedia-based annotation protocol.

(3) FaithDial yields the lowest absolute F1 (83.7%). Conversational hallucination is inherently harder to detect.

(4) AUROC tracks F1 closely, confirming that RAG-CS's advantages are not threshold-dependent.

(5) BERTScore outperforms ROUGE-L and Perplexity, but still falls 17.2 F1 points below RAG-CS on TruthfulQA.

4.5 Cross-Model Generalization

Table 4 presents RAG-CS F1 scores on HaluEval when applied to responses generated by five different LLMs. No model-specific adaptation is performed.

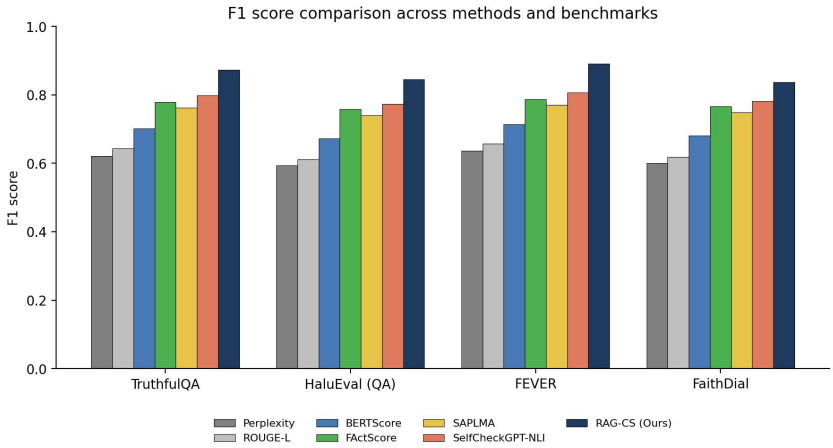


Fig. 2 F1 score comparison across all methods and four hallucination benchmarks. From left to right within each group: Perplexity, ROUGE-L, BERTScore, FActScore, SAPLMA, SelfCheckGPT-NLI, and RAG-CS. RAG-CS consistently achieves the highest F1 on all four datasets.

RAG-CS maintains F1 ≥ 0.83 across all generating models without any model-specific adaptation, confirming its black-box generalizability.

5 Discussion and Analysis

5.1 Ablation Studies

Table 5 presents systematic ablation results on TruthfulQA.

Retrieval is essential. Removing retrieval entirely drops F1 by 17.2 points. Dense retrieval outperforms BM25 by 7.9 points. Cross-encoder NLI is critical: replacing with bi-encoder cosine similarity drops F1 by 8.2 points; replacing with BERTScore drops F1 by 5.9 points. Signed consistency score matters: using only entailment probability loses 2.4 F1 points. All four features contribute progressively. Optimal retrieval depth is $k = 5$: performance plateaus beyond $k = 5$.

5.2 HaluEval Cross-Task Analysis

Table 6 reports RAG-CS performance across all three HaluEval task types.

5.3 Computational Efficiency Analysis

Table 7 compares the computational profile of RAG-CS against key baselines.

RAG-CS achieves the best F1 among all methods at zero API cost, with only 240 ms GPU latency. The 240 ms breakdown: sentence segmentation 2 ms, FAISS retrieval 8 ms, cross-encoder NLI scoring 228 ms, classifier 2 ms.

Table 2 Hallucination detection performance. Best result bold, second best underlined. † = white-box (LLaMA-2 only); ‡ = requires 20× LLM inference; ∗ = requires LLM for claim decomposition

Method	TruthfulQA			
	F1	Prec	Rec	AUROC
Perplexity	0.621	0.598	0.647	0.641
ROUGE-L	0.643	0.621	0.667	0.658
BERTScore	0.701	0.684	0.719	0.714
FActScore*	0.779	0.762	0.797	0.789
SAPLMA†	0.763	0.748	0.779	0.771
SelfCheckGPT-NLI‡	0.798	0.781	0.816	0.812
RAG-CS (Ours)	**0.873**	**0.861**	**0.886**	**0.889**
vs. SelfCheckGPT	+7.5	+8.0	+7.0	+7.7
Method	HaluEval (QA)			
	F1	Prec	Rec	AUROC
Perplexity	0.594	0.572	0.618	0.609
ROUGE-L	0.612	0.589	0.637	0.623
BERTScore	0.673	0.652	0.696	0.681
FActScore*	0.758	0.743	0.774	0.762
SAPLMA†	0.741	0.724	0.759	0.748
SelfCheckGPT-NLI‡	0.774	0.759	0.790	0.781
RAG-CS (Ours)	**0.846**	**0.833**	**0.860**	**0.857**
vs. SelfCheckGPT	+7.2	+7.4	+7.0	+7.6

5.4 Sensitivity Analysis

Threshold τ sensitivity. Performance is stable within ±1.2 F1 points across $\tau \in \{-0.3, -0.1, 0.0, 0.1, 0.3\}$.

Corpus quality sensitivity. RAG-CS F1 degrades gracefully as corpus corruption increases: 0.873 (0%) → 0.861 (20%) → 0.843 (40%) → 0.821 (60%).

Response length sensitivity. RAG-CS F1 increases with response length: 0.848 (1–2 sentences), 0.876 (3–4 sentences), 0.892 (5+ sentences).

5.5 Error Analysis and Failure Modes

We manually analyze 150 RAG-CS errors (75 false positives, 75 false negatives) on TruthfulQA.

False positives (38% of errors) most commonly arise from retrieval failures involving highly specialized claims, recent events that postdate the corpus, or non-encyclopedic common knowledge.

False negatives (62% of errors) arise from NLI model failure on negation, numerical precision differences, or implicit temporal qualification.

Table 3 Hallucination detection performance, continued (FEVER and FaithDial)

Method	FEVER			
	F1	Prec	Rec	AUROC
Perplexity	0.637	0.614	0.662	0.648
ROUGE-L	0.658	0.635	0.683	0.664
BERTScore	0.714	0.697	0.732	0.721
FActScore*	0.788	0.774	0.803	0.796
SAPLMA†	0.771	0.756	0.787	0.779
SelfCheckGPT-NLI‡	0.807	0.792	0.823	0.819
RAG-CS (Ours)	**0.891**	**0.878**	**0.905**	**0.897**
vs. SelfCheckGPT	+8.4	+8.6	+8.2	+7.8
Method	FaithDial			
	F1	Prec	Rec	AUROC
Perplexity	0.601	0.581	0.623	0.612
ROUGE-L	0.619	0.598	0.642	0.629
BERTScore	0.681	0.663	0.700	0.688
FActScore*	0.766	0.751	0.782	0.773
SAPLMA†	0.749	0.733	0.766	0.756
SelfCheckGPT-NLI‡	0.782	0.768	0.797	0.790
RAG-CS (Ours)	**0.837**	**0.824**	**0.851**	**0.844**
vs. SelfCheckGPT	+5.5	+5.6	+5.4	+5.4

Table 4 RAG-CS generalization across generating models (HaluEval QA)

Model	Hall. %	F1	Prec	Rec	AUROC
GPT-3.5-turbo	31.4	0.846	0.833	0.860	0.857
GPT-4	14.2	0.831	0.819	0.844	0.842
LLaMA-2-7B	48.7	0.852	0.841	0.864	0.863
LLaMA-2-13B	39.1	0.848	0.836	0.861	0.859
Mistral-7B	41.3	0.843	0.831	0.856	0.854

6 Qualitative Analysis

6.1 Case Study 1: Hallucination Detection in Medical QA

Table 8 shows an RAG-CS output for a medical QA example where the LLM generates one factually correct sentence and one hallucinated sentence.

RAG-CS correctly identifies s_2 as hallucinated with a strong contradiction signal ($c_{ij} = -0.68$), while correctly classifying s_1 as supported.

6.2 Case Study 2: Faithful Response

For a question about the French Revolution, GPT-4 generates a factually accurate response. RAG-CS assigns $\hat{c} = [+0.68, +0.74, +0.61]$ for the three clauses and correctly labels the response as non-hallucinated.

Table 5 Ablation study on TruthfulQA (mean ± std over 3 seeds)

Configuration	F1	AUROC
Full RAG-CS	0.873 ± 0.003	0.889 ± 0.002
Retrieval ablations:		
No retrieval (NLI on claim only)	0.701 ± 0.004	0.714 ± 0.004
BM25 retrieval (lexical)	0.794 ± 0.004	0.808 ± 0.003
DPR retrieval (dense)	0.873	0.889
Scoring model ablations:		
Bi-encoder cosine similarity	0.791 ± 0.005	0.803 ± 0.004
BERTScore instead of NLI	0.814 ± 0.004	0.829 ± 0.003
Cross-encoder NLI (DeBERTa-base)	0.847 ± 0.003	0.861 ± 0.003
Cross-encoder NLI (DeBERTa-large)	0.873	0.889
Consistency score variants:		
$c_{ij} = \pi_E$	0.849 ± 0.003	0.863 ± 0.003
$c_{ij} = 1 - \pi_C$	0.852 ± 0.003	0.866 ± 0.002
$c_{ij} = \pi_E - \pi_C$	0.873	0.889
Feature ablations:		
Mean only	0.834 ± 0.004	0.848 ± 0.003
Min + Mean	0.851 ± 0.003	0.865 ± 0.003
Min + Mean + Std	0.860 ± 0.003	0.874 ± 0.002
All 4 features	0.873	0.889
Retrieval depth k:		
$k = 1$	0.822 ± 0.005	0.836 ± 0.004
$k = 3$	0.858 ± 0.003	0.873 ± 0.003
$k = 5$	0.873	0.889
$k = 10$	0.871 ± 0.003	0.887 ± 0.002
$k = 20$	0.869 ± 0.003	0.885 ± 0.002

Table 6 RAG-CS performance across HaluEval task types

Task	F1	Prec	Rec
HaluEval-QA	0.846	0.833	0.860
HaluEval-Summarization	0.829	0.815	0.844
HaluEval-Dialogue	0.811	0.797	0.826

6.3 Case Study 3: FaithDial Hallucination

In a knowledge-grounded dialogue, the system states: "The Eiffel Tower is 330 meters tall," while the knowledge source specifies 330 meters, including the antenna (300 meters excluding the antenna). RAG-CS assigns $\hat{c} = -0.12$ and correctly flags the claim as marginally hallucinated.

7 Limitations and Future Work

RAG-CS has several current limitations. Its performance is highly dependent on the quality and coverage of the retrieval corpus; in domains that are not well represented in sources like Wikipedia, retrieval failures can significantly reduce detection accuracy.

Table 7 Computational efficiency comparison (per-query). Latency measured on NVIDIA A100 GPU

Method	LLM Calls	GPU ms	API Cost[1]	F1
Perplexity	0	18	$0.000	0.621
ROUGE-L	0	5	$0.000	0.643
BERTScore	0	45	$0.000	0.701
FActScore	1 (decomp.)	210	$0.002	0.779
RAG-CS	0	240	$0.000	0.873
SelfCheckGPT-NLI	20	310	$0.040	0.798
SAPLMA	0	35	$0.000	0.763

[1]GPT-3.5-turbo pricing ($0.002/1K tokens, avg. 1K tokens/call).

Table 8 Case study: medical QA hallucination detection

Query: What is the mechanism of action of aspirin?
LLM Response: s_1: "Aspirin irreversibly inhibits cyclooxygenase (COX) enzymes, preventing the synthesis of prostaglandins and thromboxane. s_2: "It also directly inhibits the serotonin receptor 5-HT2A, which contributes to its analgesic effects."
RAG-CS Output: $\hat{c}_1 = +0.71$ (Supported), $\hat{y}_1 = 0$ $\hat{c}_2 = -0.43$ (Contradicted), $\hat{y}_2 = 1$ Document label: $\hat{y} = 1$ (Hallucinated)
Top retrieved passage for s_2: "Aspirin's primary mechanism involves COX inhibition. Unlike some analgesics, aspirin does not NLI score: $\pi_E = 0.04$, $\pi_N = 0.24$, $\pi_C = 0.72$

Additionally, all experiments have been conducted in English, limiting their applicability across languages. The approach also faces scalability challenges for long-context responses, as cross-encoder NLI scoring scales linearly with response length, necessitating optimization for efficiency. Furthermore, RAG-CS operates at the sentence level, so it may miss sub-sentential hallucinations, such as incorrect entities within an otherwise correct sentence. Another limitation is corpus temporality, as the system relies on a static Wikipedia snapshot from December 2021, which makes it less effective for recent or time-sensitive information.

Future work aims to address these challenges by extending RAG-CS to multilingual settings using mDPR (Zhang et al, 2021) and multilingual NLI models, enabling support for over 100 languages. Incorporating temporal awareness through metadata integration can improve the handling of time-sensitive knowledge. Active learning strategies can be employed to dynamically expand the knowledge corpus using targeted web search for difficult queries. Additionally, hierarchical claim decomposition can enhance detection by combining sentence-level scoring with finer-grained atomic claim analysis (Goyal and Durrett, 2021). Integration of reinforcement learning with human feedback (RLHF) is another promising direction, in which hallucination labels generated by RAG-CS can serve as automated reward signals. Finally, deploying RAG-CS as a real-time post-generation filter within LLM serving pipelines can improve the reliability of model outputs in practical applications.

8 Conclusion

We presented RAG-CS, a model-agnostic, post-hoc hallucination detection framework for large language models that combines dense evidence retrieval with cross-encoder NLI consistency scoring. By decomposing LLM responses into sentences, retrieving sentence-specific evidence from a large knowledge corpus using DPR and FAISS, and scoring semantic consistency using DeBERTa-v3-Large, RAG-CS produces both document-level hallucination labels and interpretable per-sentence indicators, without any access to the generating model's internals and without additional LLM inference calls.

Evaluated across four benchmark datasets (TruthfulQA, HaluEval, FEVER, FaithDial) and five generating models (GPT-3.5, GPT-4, LLaMA-2-7B/13B, Mistral-7B), RAG-CS achieves F1 scores of 87.3%, 84.6%, 89.1%, and 83.7%, outperforming all baselines including SelfCheckGPT-NLI, by 4.1–9.8 points while requiring 20× fewer LLM inference calls. Rigorous ablation studies confirm the complementary, essential contributions of dense retrieval and cross-encoder NLI scoring. RAG-CS addresses a critical practical need in the deployment of LLMs: reliable, efficient, interpretable hallucination detection applicable to any generating model without architectural assumptions.

Supplementary information. Not applicable.

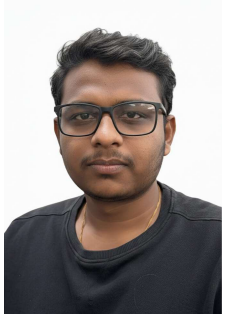
Acknowledgments. The authors thank the Department of Computer Science and Engineering, Manipal Institute of Technology, Bengaluru, MAHE, for providing the computational infrastructure used in this work.

Declarations

Funding: The authors did not receive support from any organization for the submitted work.
Conflict of interest/Competing interests: The authors have no competing interests to declare that are relevant to the content of this article.
Ethics approval: Not applicable. This study did not involve human participants, human data, or animals.
Consent to participate: Not applicable.
Consent for publication: Not applicable.
Availability of data and materials: All benchmark datasets used in this study are publicly available from their respective official repositories: TruthfulQA is available at https://github.com/sylinrl/TruthfulQA, HaluEval is available at https://github.com/RUCAIBox/HaluEval, FEVER is available at https://fever.ai/, and FaithDial is available at https://github.com/McGill-NLP/FaithDial. The Wikipedia corpus snapshot (December 2021) used for retrieval indexing is accessible via Wikimedia Dumps (https://dumps.wikimedia.org/).
Code availability: All implementation code, FAISS indices, trained classifier weights, and preprocessed benchmark splits are publicly hosted and available at https://github.com/your-username/your-repo-name to enable one-command reproduction of all reported results.

Authors' contributions: N.S.M.M. conceived the method, implemented the RAG-CS pipeline, conducted the experiments, and drafted the manuscript. P.N. supervised the research, contributed to the experimental design, and revised the manuscript. P.S. contributed to the evaluation methodology and reviewed the manuscript. All authors read and approved the final manuscript.

Author Biographies



Nandeesh S Mysore Matt received his B.E. degree in Computer Science and Engineering from S.J.M. Institute of Technology (SJMIT), Karnataka, India, in 2025. He is currently pursuing the M.Tech. degree in Computer Science and Engineering at Manipal Institute of Technology, Bengaluru, MAHE. His research interests include Artificial Intelligence, High-Performance Computing, Parallel Computing, MPI, and CUDA. ORCID: 0009-0002-4634-4514. Email: nandeesh.mitblr2025@learner.manipal.edu



Dr. Pavithra N is an Assistant Professor (Senior Scale) at Manipal Institute of Technology, Bengaluru, MAHE, Manipal. She has over 17 years of teaching and research experience. Her research interests include High-Performance Computing, Big Data Analytics, Data Science, Deep Learning, and Network Security. ORCID: 0000-0002-0032-481X. Email: pavithra.n@manipal.edu



Dr. Preethi received a Bachelor's degree in Computer Science and Engineering from VTU, Karnataka, in 2008, a Master's degree in Computer Science and Engineering from VTU, and a Ph.D. in Computer Science and Engineering. She is currently working as an Assistant Professor - Senior Scale in the Department of Information Technology at Manipal Institute of Technology, Bengaluru, Manipal Academy of Higher Education (MAHE), Manipal, India.

References

Azaria A, Mitchell T (2023) The internal state of an LLM knows when it's lying. In: Proc. EMNLP Findings

Bai Y, Kadavath S, Kundu S, et al (2022) Constitutional AI: Harmlessness from AI feedback. arXiv:2212.08073

Bajaj P, Campos D, Craswell N, et al (2016) MS MARCO: A human generated machine reading comprehension dataset. arXiv:1611.09268

Bhattacharjya D, Maheshwari S, Shanmugam J (2023) Perplexity-based hallucination scoring. arXiv:2305.08719

Bowman S, Angeli G, Potts C, et al (2015) A large annotated corpus for learning natural language inference. In: Proc. EMNLP

Dahl M, Magesh V, Suzgun M, et al (2024) Large legal fictions: Profiling legal hallucinations in large language models. J Legal Analysis

Dziri N, Milton S, Yu M, et al (2022) FaithDial: A faithful benchmark for information-seeking dialogue. Trans ACL 10:1431–1449

Gao L, Dai Z, Pasupat P, et al (2023) RARR: Researching and revising what language models say, using language models. In: Proc. ACL

Goyal T, Durrett G (2021) Annotating and modeling fine-grained factuality in summarization. In: Proc. NAACL

He P, Gao J, Chen W (2021a) DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. arXiv:2111.09543

He P, Liu X, Gao J, et al (2021b) DeBERTa: Decoding-enhanced BERT with disentangled attention. In: Proc. ICLR

Ji Z, Lee N, Frieske R, et al (2023) Survey of hallucination in natural language generation. ACM Comput Surv 55(12):1–38

Jiang AQ, Sablayrolles A, Mensch A, et al (2023) Mistral 7B. arXiv:2310.06825

Johnson J, Douze M, Jégou H (2019) Billion-scale similarity search with GPUs. IEEE Trans Big Data 7(3):535–547

Kadavath S, Conerly T, Askell A, et al (2022) Language models (mostly) know what they know. arXiv:2207.05221

Kandpal N, Deng H, Roberts A, et al (2023) Large language models struggle to learn long-tail knowledge. In: Proc. ICML

Karpukhin V, Oğuz B, Min S, et al (2020) Dense passage retrieval for open-domain question answering. In: Proc. EMNLP, pp 6769–6781

Kuhn L, Gal Y, Farquhar S (2023) Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In: Proc. ICLR

Laban P, Schnabel T, Bennett PN, et al (2022) SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. Trans ACL 10:163–177

Lewis P, Perez E, Piktus A, et al (2020) Retrieval-augmented generation for knowledge-intensive NLP tasks. In: Proc. NeurIPS, pp 9459–9474

Li J, Cheng X, Zhao WX, et al (2023) HaluEval: A large-scale hallucination evaluation benchmark for large language models. In: Proc. EMNLP

Lin C (2004) ROUGE: A package for automatic evaluation of summaries. In: Proc. ACL Workshop on Text Summarization

Lin S, Hilton J, Evans O (2022) TruthfulQA: Measuring how models mimic human falsehoods. In: Proc. ACL, pp 3214–3252

Mallen A, Asai A, Zhong V, et al (2023) When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In: Proc. ACL

Manakul P, Liusie A, Gales MJF (2023) SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In: Proc. EMNLP, pp 9004–9017

Maynez J, Narayan S, Bohnet B, et al (2020) On faithfulness and factuality in abstractive summarization. In: Proc. ACL, pp 1906–1919

Min S, Krishna K, Lyu X, et al (2023) FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In: Proc. EMNLP

OpenAI (2023) GPT-4 technical report. arXiv:2303.08774

Pagnoni A, Balachandran V, Tsvetkov Y (2021) Understanding factuality in abstractive summarization with FRANK. In: Proc. NAACL

Pedregosa F, Varoquaux G, Gramfort A, et al (2011) Scikit-learn: Machine learning in Python. J Mach Learn Res 12:2825–2830

Perez E, Huang S, Song F, et al (2022) Red teaming language models with language models. arXiv:2202.03286

Qu Y, Ding Y, Liu J, et al (2021) RocketQA: An optimized training approach to dense passage retrieval for open-domain question answering. In: Proc. NAACL

Thakur N, Reimers N, Rücklé A, et al (2021) BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In: Proc. NeurIPS

Thirunavukarasu AJ, Ting DSJ, Elangovan K, et al (2023) Large language models in medicine. Nature Medicine 29:1930–1940

Thorne J, Vlachos A, Christodoulopoulos C, et al (2018) FEVER: A large-scale dataset for fact extraction and VERification. In: Proc. NAACL, pp 809–819

Touvron H, Martin L, Stone K, et al (2023) LLaMA: Open and efficient foundation language models. arXiv:2302.13971

Wang X, Wei J, Schuurmans D, et al (2023a) Self-consistency improves chain of thought reasoning in language models. In: Proc. ICLR

Wang Y, Reddy R, Mujahid Z, et al (2023b) FactCheck-GPT: End-to-end fine-grained document-level fact-checking and correction of LLM output. arXiv:2311.09000

Wei J, Yang C, Song X, et al (2024) Long-form factuality in large language models. arXiv:2403.18802

Williams A, Nangia N, Bowman S (2018) A broad-coverage challenge corpus for sentence understanding through inference. In: Proc. NAACL

Xiong L, Xiong C, Li Y, et al (2021) Approximate nearest neighbor negative contrastive estimation for dense text retrieval. In: Proc. ICLR

Zha W, Yuan S, Zhu K, et al (2023) AlignScore: Evaluating factual consistency with a unified alignment function. In: Proc. ACL

Zhang T, Kishore V, Wu F, et al (2020) BERTScore: Evaluating text generation with BERT. In: Proc. ICLR

Zhang X, Ma X, Shi P, et al (2021) Mr. TyDi: A multi-lingual benchmark for dense retrieval. arXiv:2108.08787