University of Milano-Bicocca
**Department of Informatics, Systems and Communication**
**Master's degree in Data Science**

# Hallucination Reduction in Large Language Models: A Multi-Step Framework for Detection and Correction

**Supervisor:** Prof. Silvio Gerli

**Co-supervisor:** Dr. Teresa Cigna

**Master's degree thesis by:**
Cristian Longoni
871070

**Academic Year 2024-2025**

# Abstract

Large Language Models (LLMs) have revolutionized Natural Language Processing by enabling fluent and contextually rich text generation. However, they remain vulnerable to *hallucinations*, producing statements that appear plausible but are factually incorrect or unsupported. This thesis affront this issue by proposing a **multi-step verification and correction framework** that systematically detects, corrects, and traces hallucinations across multiple levels of analysis. To guarantee that every generated sentence is supported by verifiable evidence, the suggested method combines explicit source traceability, question-answer reasoning, and iterative validation. The framework prioritizes factual correction and clear justification of each retained claim, in contrast to current approaches that only concentrate on detection. The system was implemented using open-source language models and evaluated across diverse domains, showing consistent improvements in factual reliability and text coherence. Although tested on a limited number of medium-length texts, the results confirm the feasibility and effectiveness of the approach. Overall, this work contributes a replicable methodology for enhancing the factual integrity of LLM-generated content and represents a step toward more **responsible, verifiable, and transparent Artificial Intelligence (AI) systems**.

# Contents

# List of Figures

# List of Tables

# Acronyms

| | |
|---|---|
| **AC** | Attribution Coverage |
| **AI** | Artificial Intelligence |
| **BCE** | Banca Centrale Europea (European Central Bank scenario) |
| **CIN** | Cinema and Art scenario |
| **HAM** | Hamilton–Ferrari scenario |
| **ICL** | In-Context Learning |
| **LLM** | Large Language Model |
| **LLaMA** | Large Language Model Meta AI |
| **GPT** | Generative Pre-trained Transformer |
| **PaLM** | Pathways Language Model |
| **MED** | Medical scenario (Nobel Prize in Physiology or Medicine 2025) |
| **MIC** | Microplastics and Gut Microbiome scenario |
| **NLP** | Natural Language Processing |
| **NLG** | Natural Language Generation |
| **NES** | Normalized Edit Similarity |
| **QA** | Question Answering |
| **QA1** | QA_Accuracy_ Concept-level |
| **QA2** | QA_Accuracy_Sentence-level |
| **RSR** | Removal Success Rate |
| **RR** | Retention Rate |
| **SSR** | Sentence Support Rate |
| $\mathbf{SSR_{strict}}$ | Strict Support Rate |
| **IG** | Information Gain |
| **UCRR** | Unsupported Claim Ratio |

| | |
|---|---|
| **Prec4** | Fourth Precision |
| **Rec4** | Fourth Recall |
| **RLHF** | Reinforcement Learning with Human Feedback |
| **RAG** | Retrieval-Augmented Generation |
| **SFT** | Supervised Fine-Tuning |
| **CoT** | Chain of Thought |
| **CoVe** | Chain of Verification |
| **GPU** | Graphics Processing Unit |
| **CPU** | Central Processing Unit |
| **RAM** | Random Access Memory |
| **VRAM** | Video Access Memory |
| **OS** | Operation System |
| **SOTA** | State-Of-The-Art |
| **JSON** | JavaScript Object Notation |

# Introduction

The rapid evolution of Artificial Intelligence over the past decade has been largely driven by the emergence of **Large Language Models (LLMs)**, which have redefined the landscape of **Natural Language Processings (NLPs)** and machine reasoning.

represent a new generation of general-purpose Artificial Intelligence systems built primarily on the transformer architecture [33]. Unlike traditional task-specific models, LLMs are trained on massive corpora of text, enabling them to acquire broad linguistic knowledge, world facts, and reasoning capabilities. Prominent examples include **GPTs-3** [2], **PaLMs** [14], **LLaMA** [32], **GPTs-4** [42], and **Gemini** [46]. By scaling both data and model parameters, these models exhibit remarkable **emergent abilities**, such as In-Context Learning[2], Chain of Thought reasoning [34], and robust instruction following [20]. These advances, however, come with growing challenges in factual reliability and interpretability, especially as model complexity and deployment scale increase.

Important rules are the **scaling laws** that show that language model performance predictably improves as a power-law function of model size, dataset size, and compute [6]. Follow-up work further argues that, for a fixed compute budget, there exists a more compute-optimal regime that favors training on larger datasets with comparatively smaller models [16]. These findings help explain the emergence of strong generalization behaviors as LLMs scale.

The development of LLMs has profoundly transformed the field of **NLP**, surpassing earlier architectures such as **recurrent** and **convolutional neural networks**. These advances have enabled breakthroughs in language understanding, generation, and reasoning tasks [43]. Their generality allows them to be applied across diverse domains without task-specific fine-tuning, making them indispensable for applications such as conversational agents, recommender systems, search engines, and decision-support tools. Furthermore, the vast parametric knowledge encoded in LLMs positions them as potential **foundation models**, reshaping how humans interact with information and redefining the boundaries of AI research [12].

Despite these successes, the reliance on large-scale web data, imperfect training objectives, and probabilistic inference introduces fundamental challenges. While biases, lack of transparency, and ethical concerns remain important issues, one of the most critical shortcomings is the phenomenon of **hallucination**, where LLMs produce outputs that are fluent and convincing yet factually incorrect or unsupported [43]. This problem is particularly pressing because hallucinations are often indistinguishable from correct outputs, making them difficult to detect automatically and potentially harmful in high-stakes applications such as

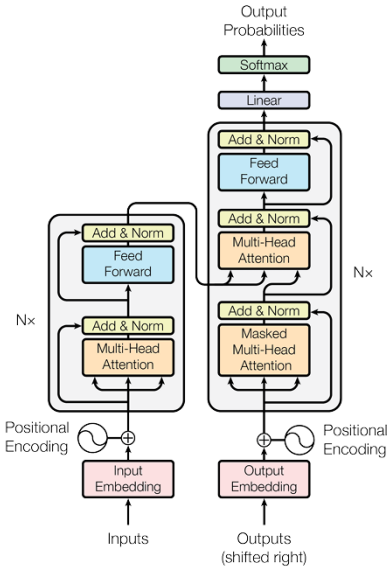


Figure 1: The Transformer model architecture. [33]

medicine, law, or education. Addressing this challenge is therefore central to ensuring the reliability and trustworthiness of LLMbased systems, and it forms the core motivation of this thesis.

The importance of LLMs lies in their ability to act as **general-purpose reasoning** and **knowledge systems**. Unlike traditional machine learning models designed for narrow, task-specific purposes, LLMs demonstrate broad adaptability, handling a wide spectrum of Natural Language Processing tasks with minimal or no task-specific fine-tuning [2], [43]. This flexibility allows them to serve as foundational technologies across diverse application domains.

From a practical perspective, LLMs power widely used tools such as conversational agents, intelligent search engines, content generation systems, and recommender platforms. Their human-like fluency and capacity to integrate contextual data enable seamless human–computer interaction, lowering barriers to access information and automating knowledge-intensive tasks. In high-stakes domains, including medicine, law, education, and finance, LLMs hold the potential to support decision-making, improve accessibility of expertise, and democratize knowledge. [12].

From a scientific perspective, the emergence of LLMs has reshaped the study of language and intelligence, providing a tool for theories of reasoning, learning, and generalization at scale. Their performance highlights the power of scaling data and parameters, leading to emergent behaviors that were not anticipated in smaller models [34]. These findings provide critical insights into the nature of generalization, alignment, and human–AI interaction.

However, the very importance of LLMs also amplifies the consequences of their limitations. When deployed in real-world systems, hallucinations and biases can mislead users, propagate misinformation, or reinforce stereotypes. This duality—enabling unprecedented opportunities while posing new risks—makes the study of hallucinations in LLMs both relevant and urgent.

### 0.0.1 Hallucinations in LLMs: Definition

In the context of **Natural Language Processing (NLP)**, the term **hallucination** refers to the generation of text that is fluent and seemingly plausible, yet factually incorrect, unverifiable, or inconsistent with the provided input [24]. The notion is loosely inspired by its psychological counterpart, where a hallucination denotes the perception of something that does not exist in reality. In LLMs, hallucinations may take several forms: contradictions with real-world facts, fabricated entities or events, failure to follow instructions, misrepresentation of contextual information, or internal logical inconsistencies [43]. For example, a model might incorrectly attribute the invention of the telephone to Thomas Edison instead of Alexander Graham Bell (**factual contradiction**), or invent a "Parisian tiger" said to have gone extinct due to the construction of the Eiffel Tower (**factual fabrication**).

Understanding and categorizing these forms of hallucination is essential for designing robust mitigation systems, such as the multi-step verification framework proposed in this thesis.

Despite extensive research on prompt engineering, self-consistency, and retrieval-augmented methods, a unified and systematic framework capable of both detecting and correcting hallucinations remains largely unexplored — a gap this work aims to address.

### 0.0.2 Proposed Solution: A Multi-Step Framework for Hallucination Reduction

As highlighted in the previous section, hallucinations remain a persistent challenge for Large Language Models, with no single strategy reliably preventing them across tasks and domains [43]. Existing approaches often trade factuality for fluency, or lack mechanisms to repair text once errors are detected.

To address this limitation, this thesis proposes a **multi-step verification and correction framework**. The central idea is to systematically refine model outputs through iterative validation against trusted sources, combining both detection and correction phases. The design takes inspiration from the **Chain of Verification (CoVe)** [15], but extends it with modular stages, each targeting a different type of error or omission. Rather than relying on a single post-hoc filter, the framework progressively enforces factual grounding at multiple levels.

The framework consists of the following steps:

1. **Initial screening (Zero Check)**: The generated article is validated sentence by sentence against the reference sources. Sentences unsupported by **all** sources are flagged and removed at this stage.

2. **Concept-level validation (First Check)**: Question Answering pairs are automatically generated from the sources, and the article's responses are compared against them. Incorrect or missing answers are identified, and the article is iteratively corrected to ensure factual consistency.

3. **Fine-grained QA analysis (Second Check)**: Questions are generated at the sentence level from the sources, and the article is further revised to integrate missing but relevant information or correct subtle contradictions.

4. **Hallucination identification (Third Check)**: Questions are generated from the article itself and tested against all sources. Claims that cannot be supported by any source are flagged as hallucinated and either corrected or removed.

5. **Source traceability (Fourth Check)**: In this final stage, every sentence of the revised article undergoes a strict source validation process. For each source, the system verifies whether the sentence is explicitly supported and extracts a literal quote (up to 50 words) as textual evidence. If at least one valid citation is found, a *secondary validation model* confirms whether the quote fully supports the factual core of the sentence, including entities, actions, objects, and numerical or temporal details. Sentences that fail both validation stages are automatically removed. This ensures that the final article is composed solely of claims that are explicitly traceable to at least one trusted source.

This structured pipeline ensures that hallucinations are addressed at multiple levels: coarse filtering, conceptual alignment, fine-grained correction, and source traceability. By combining detection with iterative rewriting, the framework preserves as much valid content as possible while eliminating unsupported or fabricated information. The result is an article that is not only linguistically fluent but also factually reliable and explicitly grounded in the provided sources. The proposed system iteratively compares model outputs with trusted sources through layered verification steps, combining factual consistency checks, Question Answering (QA) validation, and source traceability mechanisms. This design directly addresses the growing need for **faithful and auditable LLM outputs**, aligning with recent literature that emphasizes multi-pass verification and self-reflection strategies [24], [37], [43].

### 0.0.3 Thesis Objective

The central objective of this thesis is to design, implement, and evaluate a **multi-step framework** for the detection and correction of hallucinations in Large Language Models.

This thesis aims to fill a gap in the current literature by developing a modular verification system that progressively refines generated text through multiple stages of analysis and correction. There are two goals:

1. **Improve factual reliability**: ensure that every sentence in the final output is grounded in at least one trusted source, reducing the risk of misinformation.

2. **Preserve linguistic fluency and content richness**: apply corrections iteratively, so that unsupported claims are either revised or eliminated, while maintaining the overall coherence and expressiveness of the text.

By achieving these objectives, the work contributes both **practically**, by offering a replicable solution for domains such as automated journalism and knowledge extraction, and **theoretically**, by advancing our understanding of hallucination mechanisms and multi-phase mitigation strategies in LLMs.

## Evaluation Metrics

The evaluation of the proposed framework integrates a set of quantitative indicators designed to measure factual reliability, correction coverage, and source traceability. Key metrics include the **Sentence Support Rate (SSR)** and **Attribution Coverage (AC))** for factual grounding, **QA accuracy** and **Information Gain** for correction efficiency, the **Hallucination Removal Success Rate (RSR)** for evaluating hallucination elimination, and the **Normalized Edit Similarity (NES))** to quantify the preservation of linguistic coherence. Together, these metrics provide a comprehensive picture of how effectively the system detects, corrects, and removes hallucinations while maintaining fidelity to the original content and sources. These indicators also enhance transparency and reproducibility, allowing quantitative comparison across models, datasets, and verification stages. In addition, the evaluation introduces **Fourth Precision (Prec4)** and **Fourth Recall (Rec4)**, which extend the analysis to the action level where Fourth Precision and Fourth Recall respectively measure the accuracy and completeness of the final factual cleanup performed in the Fourth Check.

### 0.0.4 Thesis Structure

This thesis is organized as follows:

- **Chapter 1 – Related Works**(State of the Art): reviews existing literature and current approaches to hallucination detection and mitigation in LLMs. It identifies the strengths and limitations of previous works and outlines the gaps this thesis aims to address.
- **Chapter 2 – Proposed Approach**(Methodology and Models): describes the multi-step framework developed in this work. It explains the rationale behind each verification step (Zero-check, First-check, Second-check, Third-check, Fourth-check) and the models used in the implementation.
- **Chapter 3 – Analysis, Results and Discussion**(Experimental Model and Indicators): details the experimental setup, including datasets, evaluation metrics (SSR, AC, QA accuracy, RSR, NES, etc.), and performance indicators. It explains how the framework was applied to real-world data and how these metrics were used to assess factual grounding, correction efficiency, hallucination removal, and text preservation. The chapter presents and analyzes the experimental results. It discusses both quantitative outcomes (accuracy, coverage, reliability) and qualitative aspects (corrections, examples of removed hallucinations).
- **Chapter 4 – Conclusions**(and Future Work): summarizes the key findings and contributions of the thesis, discusses its practical and theoretical implications, and outlines future research directions to further improve hallucination detection and correction.

Given these challenges, the next chapter surveys detection and mitigation strategies, along with benchmarks and evaluation protocols, to situate our framework within the current State-Of-The-Art.

# Chapter 1

# Related Works

This chapter surveys the State-Of-The-Art on hallucination *detection*, *evaluation*, and *mitigation* in Large Language Models (LLMs). We first review detection methods (factuality vs. faithfulness), then the main benchmarks used to evaluate models and detectors, and finally mitigation strategies across data, training, and inference. We conclude with a synthesis that highlights gaps and positions this thesis.

In particular, we want to answer to four questions:

- **Which methods are currently most effective for detection and mitigation?**
- **How do these methods work, and which techniques do they rely on?**
- **What are their weaknesses and limitations?**
- **How does our multi-step framework relate to and extend existing work?**

### 1.0.1 Introduction

The problem of hallucinations in Large Language Models has received increasing attention in recent years, leading to the development of various methods for their detection, evaluation, and mitigation. This chapter reviews the most relevant approaches, with a focus on those discussed in recent surveys such as Huang et al [43]. The objective is divided into two tasks: first, to understand the current State-Of-The-Art in identifying and addressing hallucinations; second, to position the proposed multi-step framework of this thesis within the broader research landscape.

There will be discussed the novel approaches of the three dimensions :

- **Detection methods**: techniques for identifying hallucinations, either by verifying factual consistency with external knowledge (factuality) or by ensuring alignment with input and instructions (faithfulness).
- **Benchmarks**: are datasets and evaluation protocols developed to measure the prevalence of hallucinations in LLMs and to assess the effectiveness of a specific detection methods.

- **Mitigation strategies**: approaches for reducing hallucinations at differ ent stages of the LLM lifecycle, including data curation, training objec tives, and inference-time techniques.

By examining these three areas, this chapter will highlight the strengths and weaknesses of existing methods and identify gaps that motivate the need for the modular, multi-step verification and correction framework proposed in this thesis.

### 1.0.2 The Origins of Hallucinations in LLMs

Through recent works, it has been found that the origins of hallucinations are multifaceted, spanning the entire life cycle of LLM development and usage. Huang et al. [43] categorize their sources into three main stages:

**Data-related origins.** Training corpora collected from the web inevitably contain misinformation, biases, and inconsistencies. LLMs, due to their strong memorization capabilities, may reproduce these falsehoods as if they were correct, leading to **imitative falsehoods** [8]. Moreover, knowledge boundaries emerge when models are asked about rare or domain-specific topics (**long-tail knowledge**) or events that occurred after the training cutoff (**up-to-date knowledge**) [47]. Alignment data, when noisy or inconsistent, can further exacerbate hallucinations [48].

**Training-related origins.** During the pre-training phase, the next-token prediction objective introduces **exposure bias**, where starting from small errors, they can accumulate into long sequences of hallucinated content [9]. Models that use Supervised Fine-Tuning often force them to produce an answer even when the correct response is unknown, instead of admitting their uncertainty [39]. In **Reinforcement Learning with Human Feedback (RLHF)**, models may exhibit **sycophancy**, prioritizing agreement with user preferences or annotator biases over factual accuracy [45].

**Inference-related origins.** Even with curated data and careful training, hallucinations may arise during inference. **Stochastic decoding strategies** (e.g., high-temperature sampling) increase the chance of sampling unlikely tokens and thus hallucinations [5]. Models may also become overconfident in their partially generated outputs, prioritizing fluency over accuracy [38]. Additional limitations such as the **softmax bottleneck** [13] and reasoning failures in multi-step tasks [35] further contribute to hallucination risks.

In summary, hallucinations are not random artifacts but rather systematic outcomes of imperfect data, training objectives, and inference strategies. These diverse sources demonstrate that no single intervention is sufficient, motivating the **multi-step verification and correction framework** introduced in this thesis.

### 1.0.3 Taxonomy of Hallucinations in LLMs

Early research on hallucinations in **Natural Language Generation (NLG)** typically distinguished between two categories: **intrinsic** and **extrinsic** halluci-

nations [24]. Intrinsic hallucinations occur when the generated output contradicts the source input, while extrinsic hallucinations arise when the model introduces information that cannot be verified against the input, regardless of its factual correctness. Although this dichotomy provided a useful starting point, it is often too coarse to capture the variety and complexity of hallucinations produced by modern LLMs, especially given their general-purpose nature and open-ended generation capabilities.

To address these limitations, Huang et al. [43] propose a refined framework that classifies hallucinations into two broad categories: **factuality hallucinations** and **faithfulness hallucinations**. This taxonomy shifts the focus from merely comparing output to input, toward evaluating whether generated content is aligned with established facts (factuality) and whether it remains consistent with the given instructions, context, and reasoning process (faithfulness). In this thesis, we treat a generated statement as a hallucination if it is factually incorrect or unverifiable against trusted external sources (factuality), or inconsistent with the provided instructions, context, or its own intermediate reasoning (faithfulness). This operationalization directly informs both the design of our multi-step framework and the selection of evaluation indicators.

**Factuality hallucinations.** Those types of hallucinations arise when the generated content diverges from real-world facts, resulting in factually incorrect or unverifiable statements. Two main subtypes are distinguished:

- **Factual contradiction:** the output conflicts with verifiable knowledge. For example, a model might claim that Thomas Edison invented the telephone, when in fact it was Alexander Graham Bell. Such errors can involve either incorrect entities (*entity-error hallucinations*) or incorrect relations between entities (*relation-error hallucinations*).
- **Factual fabrication:** the output introduces unverifiable or fabricated information. An example is the claim that the construction of the Eiffel Tower led to the extinction of the "Parisian tiger," a species that never existed. Fabrications can also take the form of **overclaims**, where the model exaggerates or attributes phenomena to single causes without evidence.

**Faithfulness hallucinations.** These occur when the generated text fails to remain faithful to the user's instructions, the provided context, or its own reasoning. They can be further divided into three subtypes:

- **Instruction inconsistency:** the model does not follow the given instruction. For example, when asked to translate "What is the capital of France?" into Spanish, the model responds with "The capital of France is Paris," rather than providing the translation.
- **Context inconsistency:** when the model misrepresents or contradicts the contextual information. For example, if the source text states that the Nile originates from the Great Lakes region of central Africa, the model may incorrectly claim that it originates from mountain ranges.
- **Logical inconsistency:** the model generates content that is internally contradictory within a reasoning chain. For example, after correctly reduc-

| Category | Type | Example Dialog | Explanation |
|---|---|---|---|
| Factuality Hallucination | Factual Contradiction | **Instruction:** What are Thomas Edison's main contributions to science and technology?<br>**Response:** Thomas Edison developed the first practical telephone and invented the light bulb. | The response is factually incorrect. In reality, **Edison improved the light bulb, building on earlier designs by others**, and **Alexander Graham Bell invented the telephone**. |
| | Factual Fabrication | **Instruction:** What are the major environmental impacts of the construction of the Eiffel Tower?<br>**Response:** The construction of the Eiffel Tower in 1889 led to the extinction of the Parisian tiger, a species that played a crucial role in the region's ecosystem. Additionally, it is widely recognized as the event that sparked the global green architecture movement. | The response erroneously states that the construction of the Eiffel Tower led to the extinction of the 'Parisian tiger' and sparked the global green architecture movement. In fact, **there is no evidence of a 'Parisian tiger' ever existing, making this a fabricated claim.** Moreover, **attributing the origins of green architecture to the Eiffel Tower is an exaggeration**, as this movement has diverse roots and cannot be traced back to a single event. |
| Faithfulness Hallucination | Instruction Inconsistency | **Instruction:** Translate the English question into Spanish: "What is the capital of France?"<br>**Response:** The capital of France is Paris. | The user explicitly provided an instruction to translate the English question into Spanish. **Instead of performing the translation as instructed, the LLM chose to answer the question.** |
| | Context Inconsistency | **Retrieved context:** The Nile is a major north-flowing river in northeastern Africa. It's traditionally considered the longest river in the world. Originating in the Great Lakes region of central Africa, it flows through several countries before emptying into the Mediterranean Sea.<br>**Instruction:** Summarize the retrieval document above.<br>**Response:** The Nile originates from the mountain ranges of central Africa and flows into the Mediterranean. | The summary incorrectly states that the Nile originates from the "mountain ranges of central Africa", while the **paragraph mentions that the Nile originates in the Great Lakes region of central Africa.** |
| | Logical Inconsistency | **Instruction:** Solve the equation 2x + 3 = 11 step by step.<br>**Response:** Step 1: Subtract 3 from both sides to get: 2x = 8. Step 2: Divide both sides by 2 to get: x = 3. So the result of the equation is x=3. | The first step correctly isolates the term with x to get 2x = 8. However, **the next step inaccurately divides 8 by 2 to yield a result of x = 3**, which is inconsistent with the earlier reasoning. |

Figure 1.1: In the image there are examples of each category of LLM hallucinations.The sentence in red indicate the hallucinatory output, on the other hand content marked in Blue stands for user instruction or provided context that contradicts the LLM hallucination [43].

ing the equation $2x + 3 = 11$ to $2x = 8$, the model may wrongly conclude that $x = 3$ instead of the correct $x = 4$.

This taxonomy highlights two complementary dimensions of hallucinations: when deviations from external reality (**factuality**) and when deviations from the intended or provided input (**faithfulness**). Together, they provide a comprehensive framework for analyzing and mitigating hallucinations in LLMs. In this thesis, we adopt this taxonomy as conceptual foundation for the proposed multi-step verification methodology, ensuring that both factual correctness and faithfulness to instructions are systematically enforced.

## 1.1 Detection Methods

Hallucination detection methods aim to automatically identify when an LLM output is factually incorrect, unverifiable, or inconsistent with its input. Following the taxonomy proposed by Huang et al. [43], these methods can be broadly divided into those targeting **factuality hallucinations** and those addressing **faithfulness hallucinations**.

### 1.1.1 Detecting Factuality Hallucinations

Factuality detection asks whether generated statements are supported by trusted knowledge. Methods can be divided into **fact-checking** approaches, which verify

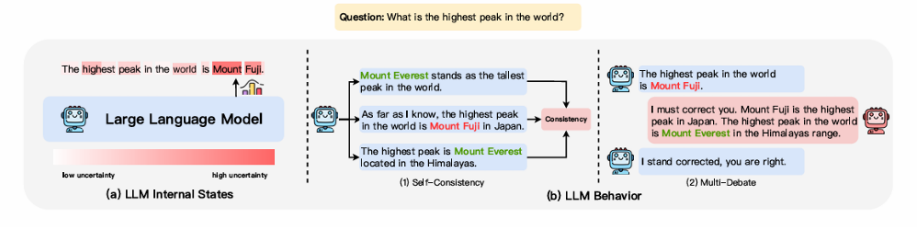


Figure 1.2: Taxonomy of Uncertainty Estimation Methods in Factual Hallucination Detection, featuring a) LLM Internal States and b)LLM Behavior,with LLM Behavior encompassing two main categories:Self-Consistency and Multi-Debate. [43]

claims against evidence, and **uncertainty estimation** approaches, which flag low-confidence content.

**Fact-checking.** A common pipeline decomposes the output into atomic claims and verifies each one: The first is *External retrieval* methods that fetch evidence from corpora, the web, or knowledge bases, and then assess whether the claims are supported. Fine-grained metrics such as **FactScore** compute the fraction of supported claims [21]. Query expansion strategies mitigate topic drift by conditioning retrieval on both the question and the model's draft answer. On the other hand *Internal checking* methods leverage the model's own parametric knowledge, e.g., through **Chain of Verification (CoVe)** style prompts [15], which generate verification questions and compare the answers back to the draft. Another example is the use of internal truth-probability judgments for atomic statements. While practical, these methods are instable since LLMs are not reliable factual databases.

**Uncertainty estimation.** Other approaches infer hallucination risk from uncertainty. These include inspection of *internal states* (token probabilities, entropy, familiarity or reconstruction scores, adversarial-entropy gaps) and *behavioral signals* (self-consistency across multiple generations, or multi-agent cross-examination to expose contradictions). For instance, **SelfCheckGPT** [27] compares multiple outputs for the same input to detect inconsistency. These approaches are lightweight and retrieval-free, but thresholds and stability may depend on the model or prompt design.
Despite these advances, the majority of detection systems rely on qualitative assessments or implicit model judgments. To move toward more rigorous and reproducible evaluation, researchers have started introducing **quantitative indicators of factual reliability**. Metrics such as *FactScore*, *ConsistencyScore*, or *SelfCheck Accuracy* attempt to quantify factual alignment, but often fail to capture sentence-level traceability or the effectiveness of correction mechanisms. To enable rigorous, sentence-level assessment and to capture the effect of *corrections*, such as **Sentence Support Rate (SSR)**, **Attribution Coverage (AC)**, and **Removal Success Rate (RSR)**, which we operationalize later in this thesis.

### 1.1.2 Detecting Faithfulness Hallucinations

Faithfulness detection evaluates whether the output remains consistent with the given input, instruction, or reasoning process. Huang et al. [43] identify five families of methods.

**Fact-based metrics.** These compute overlaps between generated and source content at n-gram, entity, or relation-triple level. Relation-level comparisons can capture subtle errors (e.g., incorrect relations despite correct entities), but depend on robust information extraction pipelines.

**Classifier-based metrics.** Here, consistency or fact-checking classifiers (often adapted from **natural language inference** models) are trained to decide whether the generation is supported by the input [24]. While effective, they can be sensitive to domain shift and annotation scarcity; adversarial training and aggregation strategies can help mitigate these issues.

**QA-based metrics.** This paradigm reformulates evaluation into question answering: key facts are extracted from the generation as questions, answered using the input, and compared to the model's answers. QA-based methods have shown improved sensitivity to semantic misalignment but depend on the quality of question/answer generation [43].

**Uncertainty-based metrics.** Faithfulness risk can also be inferred from uncertainty signals such as entropy, log-probability, or ensemble-style variance. Sequence-level confidence, often length-normalized, is widely used. Some methods penalize keywords or high-impact terms to prevent the propagation of confidently wrong content.

**LLM-based judgment.** Finally, LLMs themselves can act as evaluators. By prompting models with explicit guidelines—direct scoring, Chain of Thought (CoT), or In-Context Learning (ICL)—researchers obtain binary or graded ratings of factuality/faithfulness [36]. While flexible and scalable, these methods risk bias and inconsistency from the evaluator model.

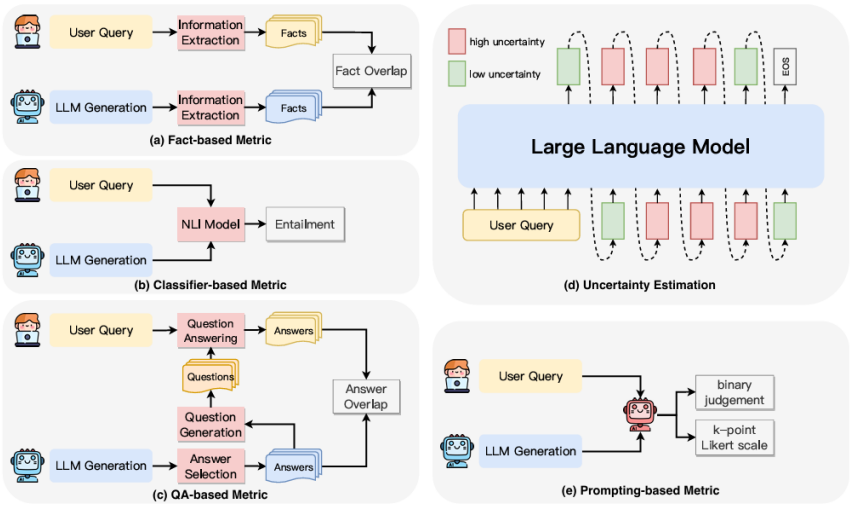


Figure 1.3: The following is an example of fidelity hallucination detecting techniques: Metrics that measure faithfulness include: a) Fact-based Metrics, which measure the overlap of facts between the generated content and the source content; b) Classifier-based Metrics, which use trained classifiers to differentiate the degree of entailment between the generated content and the source content; c) QA-based Metrics, which use question-answering systems to verify the consistency of information between the generated content and the source content; d) Uncertainty Estimation, which measures the model's confidence in its outputs; e) Prompting-based Metrics, in which LLMs are made to act as evaluators by measuring the faithfulness of generated content using particular prompting strategies.[43]

## 1.2 Benchmarks for Hallucinations

To systematically evaluate hallucinations in LLMs and the performance of detection methods, a wide range of **benchmarks** has been proposed. Huang et al. [43] categorize them into two groups: **evaluation benchmarks**, which measure the hallucination tendencies of LLMs, and **detection benchmarks**, which assess the quality of automatic hallucination detection methods. Together, they form the backbone of empirical progress in this area.

### 1.2.1 Hallucination Evaluation Benchmarks

Evaluation benchmarks are designed to stress-test LLMs under conditions where hallucinations are more likely to occur, often by probing long-tail knowledge, recency, or domain-specific expertise.

**Long-tail factual knowledge.** Datasets such as **PopQA** [26] and **Head-to-Tail** [41] explicitly test rare entities and facts. These benchmarks expose how LLMs, despite strong generalization on frequent knowledge, struggle with low-resource concepts.

**Recency and up-to-date knowledge.** **REALTIMEQA** and **FreshQA** [40] evaluate a model's ability to incorporate recent events and debunk false premises. These are particularly relevant because most LLMs are trained on static corpora, leading to hallucinations when queried about post-cutoff information.

**Imitative falsehoods.** **TruthfulQA** [19] it targets whether models reproduce common misconceptions or human-like falsehoods. It still is one of the most influential hallucination benchmarks, with multilingual and Chinese variants expanding its reach.

**Domain-specific hallucinations.** Benchmarks such as **Med-HALT** for medicine [30] and **HaluEval 2.0** for multi-domain QA [25] evaluate hallucinations in critical expert contexts. These datasets are especially important for high-stakes applications, where hallucinated outputs may cause real-world harm.

### 1.2.2 Hallucination Detection Benchmarks

Detection benchmarks provide annotated corpora to train and evaluate hallucination detection methods, often offering sentence or claim-level labels.

**Sentence-level factuality.** **SelfCheckGPT-WikiBio** [27] contains annotated hallucinations in biographical text, focusing on verifying claims sentence by sentence.

**Balanced classification.** **FELM** [22] it provides a balanced benchmark for hallucination detection, when labels are evenly distributed among supported, unsupported, and fabricated claims. This helps avoid skewed evaluation due to label imbalance.

| Benchmark | Datasets | Data Size | Language | Attribute | | | Task | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Factuality | Faithfulness | Manual | Task Type | Input | Label | Metric |
| TruthfulQA [182] | - | 817 | English | ✔ | ✗ | ✔ | Generative QA Multi-Choice QA | Question | Answer | LLM-Judge & Human |
| REALTIMEQA [148] | - | Dynamic | English | ✔ | ✗ | ✔ | Multi-Choice QA Generative QA | Question | Answer | Acc EM & F1 |
| SelfCheckGPT-Wikibio [213] | - | 1,908 | English | ✗ | ✔ | ✗ | Detection | Paragraph & Concept | Passage | AUROC |
| HaluEval [169] | Task-specific | 30,000 | English | ✗ | ✔ | ✗ | Detection | Query | Response | Acc |
| | General | 5,000 | English | ✗ | ✔ | ✗ | Detection | Task Input | Response | Acc |
| Med-HALT [303] | - | 4,916 | Multilingual | ✔ | ✗ | ✗ | Multi-Choice QA | Question | Choice | Pointwise Score & Acc |
| FACTOR [223] | Wiki-FACTOR | 2,994 | English | ✔ | ✗ | ✗ | Multi-Choice QA | Question | Answer | likelihood |
| | News-FACTOR | 1,036 | English | ✔ | ✗ | ✗ | Multi-Choice QA | Question | Answer | likelihood |
| BAMBOO [76] | SenHallu | 200 | English | ✗ | ✔ | ✗ | Detection | Paper | Summary | P & R & F1 |
| | AbsHallu | 200 | English | ✗ | ✔ | ✗ | Detection | Paper | Summary | P & R & F1 |
| ChineseFactEval [311] | - | 125 | Chinese | ✔ | ✗ | ✔ | Generative QA | Question | - | Score |
| HaluQA [49] | Misleading | 175 | Chinese | ✔ | ✗ | ✔ | Generative QA | Question | Answer | LLM-Judge |
| | Misleading-hard | 69 | Chinese | ✔ | ✗ | ✔ | Generative QA | Question | Answer | LLM-Judge |
| | Knowledge | 206 | Chinese | ✔ | ✗ | ✔ | Generative QA | Question | Answer | LLM-Judge |
| FreshQA [308] | Never-changing | 150 | English | ✔ | ✗ | ✔ | Generative QA | Question | Answer | Human |
| | Slow-changing | 150 | English | ✔ | ✗ | ✔ | Generative QA | Question | Answer | Human |
| | Fast-changing | 150 | English | ✔ | ✗ | ✔ | Generative QA | Question | Answer | Human |
| | False-premise | 150 | English | ✔ | ✗ | ✔ | Generative QA | Question | Answer | Human |
| FELM [42] | - | 3,948 | English | ✔ | ✔ | ✗ | Detection | Question | Response | Balanced Acc & F1 |
| PHD [340] | PHD-LOW | 100 | English | ✗ | ✔ | ✗ | Detection | Entity | Response | P & R & F1 |
| | PHD-Meidum | 100 | English | ✗ | ✔ | ✗ | Detection | Entity | Response | P & R & F1 |
| | PHD-High | 100 | English | ✗ | ✔ | ✗ | Detection | Entity | Response | P & R & F1 |
| ScreenEval [158] | - | 52 | English | ✗ | ✔ | ✗ | Detection | Document | Summary | AUROC |
| RealHall [90] | COVID-QA | N/A | English | ✗ | ✔ | ✗ | Detection | Question | Answer | AUROC |
| | DROP | N/A | English | ✗ | ✔ | ✗ | Detection | Question | Answer | AUROC |
| | Open Assistant | N/A | English | ✗ | ✔ | ✗ | Detection | Question | Answer | AUROC |
| | TriviaQA | N/A | English | ✗ | ✔ | ✗ | Detection | Question | Answer | AUROC |
| LSum [85] | - | 6,166 | English | ✗ | ✔ | ✗ | Detection | Document | Summary | Balanced Acc |
| $SAC^3$ [364] | HotpotQA | 250 | English | ✗ | ✔ | ✗ | Detection | Question | Answer | AUROC |
| | NQ-Open | 250 | English | ✗ | ✔ | ✗ | Detection | Question | Answer | AUROC |
| HaluEval 2.0 [168] | Biomedicine | 1,535 | English | ✔ | ✗ | ✗ | Generative QA | Question | Answer | MiHR & MaHR |
| | Finance | 1,125 | English | ✔ | ✗ | ✗ | Generative QA | Question | Answer | MiHR & MaHR |
| | Science | 1,409 | English | ✔ | ✗ | ✗ | Generative QA | Question | Answer | MiHR & MaHR |
| | Education | 1,701 | English | ✔ | ✗ | ✗ | Generative QA | Question | Answer | MiHR & MaHR |
| | Open domain | 3,000 | English | ✔ | ✗ | ✗ | Generative QA | Question | Answer | MiHR & MaHR |

Figure 1.4: A summary of current benchmarks for hallucinations. For Attribute, Manual indicates whether the data inputs are handwritten, and Factuality and Faithfulness indicate whether the benchmark is used to assess LLM's factuality or to identify faithfulness hallucinations.[43]

**Aggregated multi-domain corpora.** **HaluEval** [25] aggregates human-labeled examples across multiple domains, serving as a large-scale detection benchmark. It complements evaluation benchmarks by focusing specifically on detection quality, often reported with AUROC, accuracy, and F-scores.

### 1.2.3 Discussion and Limitations

While these benchmarks have driven rapid progress, their coverage remains incomplete. Many are domain-specific, constrained in scale, or focus narrowly on certain hallucination types. As noted by Huang et al. [43], there is no unified benchmark that comprehensively covers both factuality and faithfulness hallucinations across tasks. This motivates the design of new frameworks and evaluation strategies such as the one proposed in this thesis that can generalize across domains while ensuring fine-grained traceability of factual grounding.

## 1.3 Mitigation Strategies

In addition to detection, substantial effort has been devoted to **mitigating hallucinations** in LLMs. Huang et al. [43] classify mitigation approaches according to the stage of the LLM lifecycle in which they are applied: data, training, and inference. Each stage offers complementary methods, but also presents significant limitations.

### 1.3.1 Data-centric Mitigations

Improving the quality of training data is a natural starting point. **Data filtering** pipelines aim to remove noisy, contradictory, or low-quality samples, thereby reducing the propagation of misinformation in the model's parametric knowledge [4]. **Model editing** techniques, such as MEMIT [28], enable targeted corrections of factual associations inside LLMs without retraining the full model. The most widely adopted defense is **Retrieval-Augmented Generation (RAG)** [11], where the model is coupled with an external retriever to ground outputs in retrieved evidence. RAG substantially improves factual grounding, especially in knowledge-intensive tasks, but remains highly dependent on retrieval precision/recall, query drift, and aggregation of multiple evidence passages.

### 1.3.2 Training-centric Mitigations

At the training stage, adjustments to objectives and alignment strategies can reduce hallucinations. In **Supervised Fine-Tuning (SFT)**, methods have been proposed to teach models to **abstain** when uncertain rather than fabricating answers **zhang2023sft**. **Reinforcement Learning with Human Feedback (RLHF)** [20] is now standard for aligning models to user intent, but it can introduce **sycophancy**, where models prioritize agreement over accuracy [45]. Extensions include reinforcement from AI feedback, multi-signal objectives, or hybrid reward models that directly optimize for factual correctness and penalize unsupported statements. However, alignment data itself may contain inconsistencies, and pushing beyond the model's intrinsic knowledge boundary can still induce fabrications.

### 1.3.3 Inference-centric Mitigations

A rich line of research addresses hallucinations at inference time. **Factuality-enhanced decoding** incorporates retrieval, rescoring, or constrained sampling to prefer outputs supported by evidence [18]. **Faithfulness-constrained decoding** integrates mechanisms that enforce consistency with the input (e.g., coverage penalties, constrained beam search) [3]. **Self-checking approaches**, such as SelfCheckGPT [27], generates multiple candidate responses and detects inconsistencies, while **multi-agent debate** frameworks encourage collaborative cross-examination among multiple model instances [23]. Finally, structured prompting strategies such as the **Chain of Verification (CoVe)** [15] decompose answers into verifiable claims, iteratively checking them against evidence before producing the final output. This method serves as the conceptual foundation for our original framework, which extends CoVe beyond detection to incorporate

**iterative correction, explicit source traceability, and integrated quantitative evaluation**. By combining verification, rewriting, and metric-based assessment, our approach enhances both factual reliability and interpretability.

### 1.3.4 Limitations of Current Approaches

Despite significant progress, no single mitigation strategy reliably prevents hallucinations across tasks and domains. Data filtering cannot fully eliminate misinformation; model editing is powerful but lacks scalability; and RAG, although effective, inherits the limitations of retrieval systems. Training-time methods like RLHF and SFT improve alignment but may optimize for preference rather than truth, and cannot address knowledge cutoffs. Inference-time strategies add robustness but often increase computational cost and latency, and their effectiveness depends on prompt design and evaluator reliability. As emphasized by Huang et al. [43], hallucination mitigation remains an open challenge, motivating more holistic and multi-stage solutions. The framework proposed in this thesis builds upon these insights by combining detection, correction, and traceability into an integrated pipeline.

## 1.4 Synthesis and Positioning of The Work

The literature on hallucinations in LLMs highlights substantial progress in both **detection** and **mitigation**, yet significant gaps remain. A synthesis across factuality and faithfulness-oriented methods shows where the field currently stands, what challenges remain, and how this thesis is positioned.

### 1.4.1 Best methods available.

For **factuality detection**, the most reliable pipelines combine claim decomposition with external retrieval and fine-grained verification (e.g., FACTSCORE, PopQA) [21], [26]. For **faithfulness detection**, QA-based evaluation methods and LLM-as-judge approaches demonstrate strong sensitivity to subtle semantic mismatches [36], [43]. Uncertainty-based techniques such as **SelfCheckGPT** [27] are attractive in zero-resource scenarios because they do not require external corpora, although they remain less stable.

### 1.4.2 Weaknesses and limitations.

Despite these advances, several weaknesses persist:

- **Retrieval quality and coverage** remain bottlenecks in fact-checking pipelines, limiting reliability when evidence is sparse [11].
- **Classifier-based detectors**, often adapted from NLI models, suffer from domain shift and annotation scarcity [24].
- **LLM-as-judge** methods, while flexible, inherit the biases and inconsistencies of the evaluator model [36].
- **Uncertainty-based metrics** are usually instable: entropy thresholds and self-consistency signals often fail to generalize across tasks.

- Critically, most approaches focus only on **detection**, leaving unsupported content flagged but unrepaired, which can still mislead end users.

### 1.4.3 Positioning of this thesis.

The framework proposed in this thesis is designed to address these limitations through a **multi-step verification and correction pipeline**. Specifically, it:

1. Operates across multiple granularities (sentence, claim, concept), enabling fine-grained localization of hallucinations.

2. Combines **external verification** (QA against trusted sources) with **internal cross-checks** (iterative LLM prompts), balancing factuality and faithfulness.

3. Incorporates not only detection but also **correction and rewriting**, ensuring that unsupported claims are revised or removed while preserving coherence and fluency.

4. Enforces explicit **traceability and metric-driven evaluation**, ensuring that each sentence in the final article is supported by at least one reference source and that this linkage is quantifiable through metrics such as Sentence Support Rate and Attribution Coveragee.

In this way, the thesis builds directly upon State-Of-The-Art approaches while contributing a novel methodology. Unlike prior monolithic or single-pass pipelines, it operationalizes hallucination mitigation as a **modular, iterative process**. This design increases robustness across domains and provides a replicable strategy for deploying LLMs in knowledge-intensive and high-stakes contexts, such as journalism and in this case we can identify the kind of hallucinations we will treat as **Faithfulness hallucinations**, in particular increases robustness across domains. In our experiments, we particularly stress **context inconsistency** within faithfulness, while still enforcing factuality through external-source grounding.

### 1.4.4 Relation to Chain of Verification (CoVe).

Among the existing approaches, the **Chain of Verification (CoVe)** [15] is particularly relevant to this thesis. CoVe demonstrates the effectiveness of decomposing model outputs into verifiable claims and iteratively checking them against trusted evidence. However, CoVe is primarily designed as a detection framework and does not directly address the correction or rewriting of unsupported statements. Building on this foundation, the framework proposed in this thesis extends the CoVe paradigm in three directions:

1. It integrates **multi-granularity analysis** (sentence, claim, and concept level) rather than focusing on a single verification layer.

2. It couples detection with **iterative correction and rewriting**, ensuring that hallucinated or unsupported claims are not only flagged but also repaired or removed.

3. It enforces full **traceability**, grounding every retained sentence in at least one reference source.

In this way, the proposed framework can be seen as a natural evolution of CoVe, transforming it from a detection-oriented method into a holistic multi-step pipeline for both **detection and correction** of hallucinations in LLMs.

Overall, while existing works have laid the groundwork for detecting and partially mitigating hallucinations, none provide an integrated framework that combines **multi-granularity verification**, **automated correction**, and **metric-based traceability**. This thesis fills that gap by operationalizing theoretical advances from detection and mitigation research into a practical, auditable pipeline that ensures measurable factual reliability in LLM-generated content. Unlike prior verification-only frameworks such as *SelfCheckGPT* [27] and *CoVe* [15], our approach explicitly integrates **corrective rewriting** and **dual-model traceability**. This transforms the hallucination detection task into a measurable and reproducible factual refinement process, bridging the gap between diagnostic verification and faithful text reconstruction.

# Chapter 2

# Proposed approach

## 2.1 Introduction and Motivation

The phenomenon of hallucinations in Large Language Models Large Language Models (LLMs) has emerged as one of the most pressing challenges for their reliable deployment in real-world applications. As highlighted in the literature review in the previous chapter, hallucinations manifest in two primary dimensions: **factuality**, where generated content contradicts or fabricates external knowledge, and **faithfulness**, where the output diverges from the given input, instructions, or reasoning process [24], [43].

These issues are especially sensitive and important in knowledge-intensive and high-stakes domains such as medicine [30], law, and journalism, where misinformation may lead to severe consequences.

Existing approaches to hallucination mitigation can be broadly categorized into two families: **detection methods**, which identify unsupported or inconsistent statements [21], [27]; and **mitigation strategies**, which attempt to reduce hallucinations through data curation [4], training objectives [20], [39], or inference-time techniques [18], [23]. While valuable, these methods share common limitations: they often operate at a single level of granularity (sentence, claim, or document), rely heavily on retrieval quality [11], and—crucially—tend to stop at detection without providing mechanisms for **correction and rewriting**. As a result, users are frequently left with flagged but unrepaired text, which undermines trust and usability.

To address these gaps, this thesis proposes a **multi-step verification and correction framework** for hallucination reduction in LLMs. The approach is inspired by the **Chain of Verification (CoVe)** method [15], which decomposes model outputs into verifiable claims and validates them iteratively, but extends it in several key ways:

- introducing multiple verification passes at different granularities (concept, sentence, and claim level);
- combining **external verification**, through QA against trusted sources and retrieval-augmented validation, to balance coverage and precision;
- incorporating correction and rewriting stages to preserve linguistic fluency while eliminating unsupported content;

- enforcing explicit traceability, ensuring that every sentence in the final output can be grounded in at least one trusted source.

This chapter introduce the methodology underlying the proposed framework. First we provide an overview of the overall pipeline, and then describe each step in detail: (0)Initial screening, (1)Concept-level validation, (2)Fine-Grained QA Analysis, (3)Hallucination identification and (4)Source Traceability. Finally, we discuss design considerations, potential challenges, and ethical implications of the approach.

### Granularity of Verification: concept, sentence, and claim

We operationalize verification at three complementary granularities. Each level uses a different unit of analysis and (crucially) a different verification direction.

**Concept-level (coverage-oriented; sources → draft). Unit**: high-level concepts (key entities, processes, relations).
**Method**: generate questions *from the sources* and check that the draft answers them correctly and does not omit core facts.
**Goal**: ensure coverage and coarse alignment with the evidence.

**Sentence-level (precision-oriented; sources → draft). Unit**: minimal factual units extracted from *source sentences* (numbers, dates, named entities, relations).
**Method**: fine-grained QA that compares the draft's answers with source-derived answers; minimally rewrite to fix local inconsistencies.
**Goal**: catch subtle mismatches and missing details.

**Claim-level (assertion-oriented; draft → sources). Unit**: atomic statements *extracted from the draft* (often representable as subject–relation–object with qualifiers).
**Method**: for each claim, search and verify explicit support in the sources; remove or repair unsupported claims.
**Goal**: enforce explicit grounding even for paraphrases and cross-sentence assertions that may not align one-to-one with source phrasing.

*Concept ≠ claim.* Concept-level starts from the sources to test whether the draft covers the main ideas; claim-level starts from the draft to test whether each assertion is supported by at least one source.

**Research Question.** Can a multi-step, evidence-first pipeline effectively detect, correct, and trace hallucinations in Large Language Model (LLM) outputs while preserving linguistic fluency and factual completeness?

## 2.2 Overview of the Framework

The proposed solution is a **multi-step verification and correction framework** designed to systematically reduce hallucinations in LLM-generated outputs. The central motivation is that hallucinations do not originate from a single cause, but from multiple factors across data, training, and inference stages [43]. Therefore, a single filtering step is insufficient. Instead, hallucination mitigation

requires a **pipeline of complementary stages**, each operating at a different level of granularity and using diverse verification strategies.

At a high level, the framework is built on three core principles:

1. **Progressive refinement.** Rather than treating hallucination detection as a one-shot task, the framework applies multiple passes of validation and correction. This ensures that coarse errors are filtered early (e.g., unsupported sentences), while finer inconsistencies are addressed later (e.g., entity-level or logical contradictions).

2. **Evidence-first verification.** The framework prioritizes *external grounding* (QA against trusted sources and retrieval-augmented validation) as the primary verification signal.

3. **Correction, not only detection.** While much prior work stops at flagging hallucinated content [22], this framework explicitly includes **rewriting** stages, where unsupported or inconsistent content is corrected or removed, while preserving fluency and coherence. This ensures that the final output is not only trustworthy but also ready for end-user consumption.

*Anticipation for future work.* Internal signals such as self-consistency or uncertainty estimation [27] could be integrated in future versions of the framework, but they are not part of the current implementation.

The framework is conceptually inspired by the **Chain of Verification (CoVe)** methodology [15], which decomposes model outputs into claims and validates them iteratively. However, it extends CoVe by applying verification at multiple granularities (sentence, concept, claim) and integrating explicit correction and rewriting mechanisms based on external evidence.

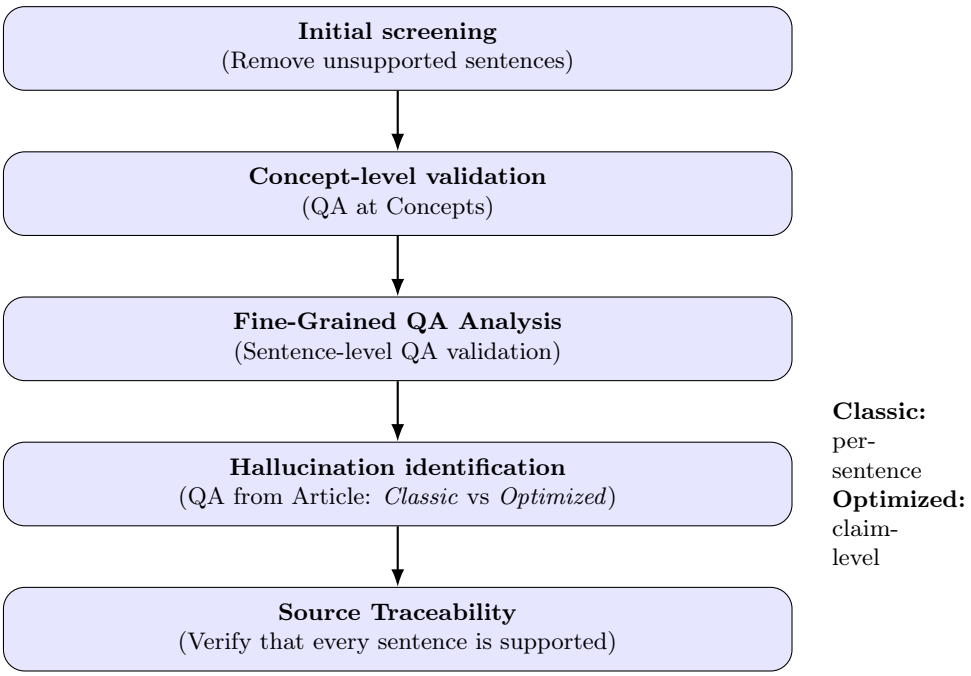


Figure 2.1: Overview of the proposed multi-step verification and correction framework. An initial draft is refined through each stage using external grounding to progressively eliminate or correct unsupported content.

As illustrated in Figure 2.1, each stage progressively refines the draft through increasingly strict verification.

This modular design ensures extensibility: new verification or correction modules can be integrated as the field evolves, while maintaining the pipeline's overall logic. By enforcing end-to-end traceability, the framework guarantees that the final text is **factually reliable**, **faithful to the sources**, and **usable in real-world applications**.

## Draft Generation

Before running the local verification pipeline, the initial article draft is produced by a large hosted LLM, specifically **GPT-4o**, accessed via a web interface (not locally). To condition the model on trustworthy evidence, we provide one or more sources (PDFs, or Journalism articles in .txt format) directly in the chat session and instruct the model to write a first-pass article *based strictly on those sources*.

**Procedure.**

1. Select and curate the source set.
2. Upload the sources into the web interface and give a standard structured instruction that the article must rely only on the provided material, avoiding unsupported claims.

3. Export the resulting *Draft* and hand it to the local, evidence-first pipeline for screening, QA-based verification, correction, and source traceability.

**Reproducibility notes.** Since the upstream model is hosted, decoding settings are controlled by the provider. We mitigate variance by fixing a stable prompt template that emphasizes source faithfulness, attaching the same source set, and archiving the exact prompt, timestamps, and the raw *Draft*. All subsequent processing (screening, QA checks, rewriting, and exports) is performed locally as detailed below.

## 2.3 Initial Screening: Filtering Unsupported Content

The first stage of the framework is the **Initial screening**, which acts as a coarse-grained filter to remove content that is entirely unsupported by any of the reference sources. The motivation for this step is that some hallucinations are so detached from the provided evidence that they cannot be reliably corrected in later stages. Instead of wasting verification and correction resources on such sentences, the Initial screening eliminates them at the outset.

### Methodology

The Initial screening operates sentence by sentence on the LLM-generated draft using a tokenizer to separate each phrase. For each sentence:

1. The system compares the sentence against all available reference sources (documents, datasets, or knowledge bases).
2. If **at least one source** provides partial or full support (e.g., existing entities, relations, or contextual overlap), the sentence is retained for refinement in following checks.
3. If the sentence is **unsupported by all sources**, it is flagged as a hallucination and removed at this stage.

This conservative rule ensures that potentially correct but imprecise statements are not prematurely deleted. For example:

- If the article states that "the concert had 800 attendees" while the source reports "100 attendees," the sentence is kept, since later steps (Concept-Level QA Validation and Fine-Grained QA Analysis) are designed to handle factual correction.
- If the article introduces an entity or event absent from all sources (e.g., "a Parisian tiger extinct due to the Eiffel Tower's construction"), the sentence is removed, as it represents a pure fabrication [19].

### Relation to Existing Work

The Initial screening is conceptually aligned with prior work on sentence-level factuality evaluation. Datasets such as **SelfCheckGPT-WikiBio** [27] and **FELM** [22] have shown that hallucinations often manifest as fully unsupported statements, which can be reliably identified with minimal context. TruthfulQA [19] further highlights how LLMs tend to generate imitative falsehoods that have no grounding in reality, underscoring the need for early removal.

Unlike later stages of the framework, the Initial screening does not attempt fine-grained correction or semantic alignment. Its sole purpose is to ensure that the draft entering subsequent stages is at least loosely anchored to the sources. By pruning out entirely fabricated content upfront, the Initial screening increases the efficiency and reliability of downstream verification modules.

### Output

The output of the Initial screening is a filtered draft in which:

- All sentences unsupported by any source have been removed.
- All sentences partially or fully supported are retained for deeper analysis.

This filtered version serves as the foundation for the **Concept-Level QA Validation**, where concept-level question answering will be used to refine factual consistency in a more targeted manner.

## 2.4 Concept-Level QA Validation

After the coarse filtering of the Initial screening, the **Concept-Level QA Validation** stage performs a more fine-grained validation at the level of **concepts and facts**. The goal is to ensure that the generated text not only avoids fully fabricated statements but also aligns with the factual content present in the sources.

### Methodology

The Concept-level validation operates as follows:

1. **Question generation.** A set of Question Answering (QA) pairs is automatically generated from the trusted sources.

   *Example:* if the source text states ‘‘`The Nile originates from the Great Lakes region of central Africa`’’`[43]`,
   the QA pair might be:

   - Q: *Where does the Nile originate?*
     A: The Great Lakes region of central Africa

2. **Article interrogation.** Each generated question is posed to the primary LLM model using only the draft article as ground truth. The response extracted from the article is compared against the ground-truth answer from the source.

3. **Consistency check.**
    - If the article provides the correct answer (in accord with the source), the corresponding passage is marked as valid.
    - If the answer is incorrect, incomplete, or missing, the article is flagged for correction at the relevant location.
4. **Iterative correction.** The article is updated to incorporate missing information or fix factual errors based on the relative ground-truth answer", while preserving fluency and coherence.

This stage transforms the problem of hallucination detection into a structured QA validation task, leveraging the fact that many factual errors can be revealed by targeted questions [21], [24].

## Relation to Existing Work

QA-based evaluation has proven particularly effective for assessing factual consistency in LLM outputs. For example, **QAGS** [7] introduced the paradigm of converting summaries into QA for factuality evaluation, while later works such as FactScore [21] refined this approach with more robust scoring mechanisms. The Concept-Level QA Validation extends these ideas from evaluation to **active correction**: instead of monly scoring factuality, the system uses QA mismatches to drive targeted revisions of the draft.

This approach is also complementary to classifier-based metrics (e.g., NLI models for entailment) [24], offering better sensitivity to missing information. By focusing on concept-level QA, the Concept-Level QA Validation directly addresses one of the main weaknesses of the Initial screening: its inability to capture subtle factual mismatches.

## Output

The output of the Concept-Level QA Validation is an article draft that:

- Retains only conceptually supported content.
- Incorporates factual corrections where the original text diverged from the sources.
- Includes newly added information when important concepts present in the sources were missing from the initial draft.

This updated draft is then passed to the **Fine-Grained QA Analysis**, which operates at an even finer granularity by validating the article at the sentence level.

## 2.5 Fine-Grained QA Analysis: Sentence-Level Validation

While the Concept-Level QA Validation operates at the concept granularity, the **Fine-Grained QA Analysis** examines factual consistency at the level of individual sentences. The objective is to capture subtle contradictions, omissions, or misrepresentations that may have survived earlier stages.

### Methodology

The **Fine-Grained QA Analysis** proceeds as follows:

1. **Question generation from sources (sentence-level).** For each sentence in the sources, generate one or more factual questions that capture entities, relations, numbers, dates, and outcomes, dividing each phrase using a tokenizer.
2. **Article interrogation.** Ask the same questions to the primary LLM model using only draft article as ground truth and extract the answers from the article text.
3. **Consistency check.** If the article's answer matches the source answer, mark the corresponding part as valid; otherwise flag it as incorrect or missing.
4. **Targeted correction.** Rewrite minimally to fix contradictions and add missing details, preserving fluency and style.

This approach ensures that every statement in the final output is anchored in at least one trusted source, thereby enforcing factual traceability.

### Relation to Existing Work

Sentence-level factuality evaluation is widely used in summarization and data-to-text generation research. Benchmarks such as **FELM** [22] and **SelfCheckGPT-WikiBio** [27] highlight how hallucinations often occur in small but critical factual details, which can be overlooked by coarser methods.

The Fine-Grained QA Analysis extends the QA-based paradigm of the Concept-level validation [21] by grounding questions in the smallest meaningful text units sentences thereby maximizing coverage of the factual space. Compared to classifier-based entailment metrics [24], sentence-level QA offers greater interpretability and fine control, since each correction can be tied back to a specific question and sentence.

### Output

The output of the Fine-Grained QA Analysis is a refined draft in which:

- Subtle contradictions between article and sources are corrected.
- Missing but relevant details from the sources are proposed to the user and incorporated if selected.

- Each sentence is more tightly aligned with the ground truth of the reference material.

This refined draft is then passed to the **Hallucination Identification**, which inverts the perspective: instead of testing the article against the sources, it tests the article's claims to see if they are supported by *any* source.

## 2.6 Hallucination Identification

The **Hallucination Identification** stage shifts the verification perspective: instead of generating questions from the sources, questions are derived directly from the **article draft**. The purpose is to identify statements that cannot be grounded in any of the reference sources, thus exposing hallucinations that have survived previous checks.

### Methodology

The **Hallucination Identification** proceeds as follows:

1. **Question generation from the draft.** Transform each sentence (classic) or extracted claim (optimized) into one or more factual questions.
2. **Cross-examination with sources.** Ask the generated questions to all reference sources. If at least one source provides a consistent answer, retain the claim; otherwise flag it as hallucinated.
3. **Identification and action.** Remove or rewrite hallucinated content based on available evidence to restore factuality.

This approach ensures that every statement in the final output is anchored in at least one trusted source, thereby enforcing factual traceability.

**Illustrative example.** *Draft sentence:* "Ferrari will introduce an aerodynamic upgrade at Suzuka that will improve lap times by 0.3 seconds." *Verification:* the system transforms the statement into factual questions such as "Did Ferrari announce an aerodynamic upgrade for Suzuka?" and "What is the reported time improvement?" When no supporting evidence is found in any of the reference sources, the sentence is flagged as hallucinated and either removed or rewritten to align with verified information.

### Two Alternative Checkers (Classic vs. Optimized)

In practice, the Hallucination Identification is instantiated with two alternative implementations, which can be selected at run-time:

**(1) Classic sentence-by-sentence checker.** This variant verifies hallucinations *per sentence*. For each sentence in the draft, the system creates one or more verification questions and queries all sources. It is simple and fast, and works well when the draft closely mirrors source phrasing. However, it can miss hallucinations embedded in multi-sentence claims or paraphrases that span larger contexts.

**(2) Optimized claim-level checker.** This variant first performs *claim extraction* (grouping one or more sentences into an atomic assertion) and then generates verification questions at the *claim* level. By reasoning beyond sentence boundaries, it achieves higher recall on paraphrased or cross-sentence assertions, at the cost of higher latency (more questions) and stricter aggregation of evidence.

**When to use which.**

- Use the *classic* checker for short drafts, few sources, or when you expect sentence-level alignment.
- Use the *optimized* checker for long-form articles, many sources, or when claims are distributed across sentences (paraphrases, coreference, or summarization artifacts).

**Logging and reproducibility.** The chosen checker is recorded in the run metadata (e.g., in the exported Excel report as **Hallucinations verification method: CLASSIC/OPTIMIZED**), enabling consistent analysis across experiments.

## Relation to Existing Work

This stage is inspired by verification paradigms such as **Chain of Verification (CoVe)** [15], where model outputs are decomposed into verifiable claims and systematically validated. It also resonates with the principle behind benchmarks like **TruthfulQA** [19], which explicitly probe whether LLMs reproduce falsehoods without grounding. The novelty here lies in turning the article itself into the source of interrogation (via sentence- or claim-level questions), thus detecting hallucinations that would otherwise remain hidden. This complements earlier stages (Initial screening and Concept-level validation), which are primarily concerned with aligning the draft to what is present in the sources.

## Output

The output of the Hallucination Identification is a further filtered draft in which:

- Any statement unsupported by all sources has been flagged as hallucinated and either removed or rewritten.
- The remaining content is guaranteed to be anchored in at least one trusted source.

This draft is then passed to the **Source Traceability**, which focuses on ensuring explicit source traceability for each surviving sentence.

## 2.7 Source Traceability

The **Source Traceability** is the final stage of the proposed framework. Its objective is to guarantee that every surviving sentence in the draft can be explicitly linked to at least one source reference, ensuring complete **traceability** of the information. This step enforces a strict standard of factual accountability: not only the text must be free from hallucinations, but each claim must also carry a clear source as anchor.

### Methodology

The Source Traceability stage is implemented as a rigorous two-phase verification process that combines literal evidence extraction and secondary validation.

**Step 1: Evidence extraction.** For each sentence in the article, the system queries all available sources to determine whether the central meaning of the sentence is explicitly supported. If support is found, a **literal citation** (up to 50 words) is automatically extracted from the source text. This ensures that every claim in the article is backed by verifiable textual evidence rather than paraphrased similarity.

**Step 2: Secondary validation.** Each sentence–citation pair is then passed to a **secondary LLM** (a lighter validation model) that checks whether the quote fully supports the key elements of the sentence — including entities, actions, objects, and any numerical or temporal details. If at least one validated quote satisfies this criterion, the sentence is retained. Otherwise, it is removed using a controlled rewriting prompt to preserve fluency.

**Illustrative example.** *Draft sentence:* "Sergio Casaro collaborated with director Sergio Leone on several film posters." The system searches the sources and extracts a literal quote ($\leq$ 50 words) such as: *"Casaro designed posters for Leone's Western trilogy during the 1960s."* The secondary validator then checks whether the quote fully supports all factual elements of the sentence (entities, action, timeframe). If validation is successful, the sentence is retained; otherwise, it is removed or rewritten for factual consistency.

**Effect of the Step.** This dual verification ensures that only sentences with explicit, validated grounding remain in the final article. The process also logs all judgments (*sentence, source, quote, final outcome*) in a structured format, enabling full traceability and transparency. The dual verification also not only guarantees factual accuracy but also produces structured data used to compute the framework's quantitative indicators (SSR, AC, and $\text{SSR}_{\text{strict}}$), linking qualitative validation with measurable evaluation.

### Relation to Existing Work

Source attribution is a long-standing issue in Natural Language Generation, particularly in summarization and data-to-text tasks. While earlier approaches often relied on lexical overlap or heuristic alignment, recent work has moved

towards more semantic and evidence-based grounding [24], [43]. The Source Traceability extends this by introducing a two-phase process: *attribution* followed by *verification*.

This design is inspired by the ideas behind Retrieval-Augmented Generation (RAG)[11], where outputs are explicitly tied to retrieved passages. However, unlike RAG, which ensures grounding only at inference time, the Source Traceability guarantees post-hoc that the final draft is fully anchored to the sources, even after multiple rounds of rewriting and correction.

### Output

The output of the Source Traceability is the final version of the article, where:

- Each sentence is supported by at least one verified source.
- Unsupported sentences have been removed, ensuring no untraceable claims remain.
- The draft has explicit grounding, maximizing both factual reliability and accountability.

This guarantees that the final output is not only fluent and factually correct but also transparent and verifiable, meeting the standards required for high-stakes applications such as automated journalism, medicine, or legal reasoning.

## 2.8 Design Considerations

The design of the proposed multi-step framework is guided by several key considerations that aim to balance factual reliability, computational efficiency, and practical usability. These principles explain why specific methodological choices were made and how they contribute to the robustness of the pipeline across domains and tasks.

### 2.8.1 Why this methodology is appropriate

This modular, iterative design ensures robustness and generalization by:

- **Combining detection and correction.** Unlike most methods that stop at flagging errors [22], this framework integrates rewriting stages, so hallucinations are not only identified but actively resolved.
- **Operating at multiple granularities.** Verification occurs at the concept, sentence, and claim levels, capturing both coarse and fine-grained inconsistencies [24], [43].
- **Preserving valid content.** By deferring nuanced corrections to later steps, the framework maximizes retention of useful information while eliminating unsupported claims.
- **Enforcing explicit grounding.** Every surviving sentence in the final draft is linked to a supporting source, which is critical for high-stakes applications such as medicine and law [30].

- **Quantitative validation.** Each verification stage contributes to measurable indicators (SSR, AC, RSR, NES), automatically computed at the end of the pipeline (`metrics.py`) and stored for reproducibility.

### 2.8.2 Granularity of Verification

A central principle is the use of **progressive refinement at different levels of granularity**. Single-level approaches, such as sentence-only verification, risk missing hallucinations that emerge across multiple sentences or in abstracted summaries. By refining from coarse to fine checks, the framework ensures broad coverage while minimizing false positives [24], [43].

### 2.8.3 Detection vs. Correction

Existing pipelines often stop at detection, leaving correction to the user [22]. This framework integrates **iterative correction and rewriting** after each verification step, ensuring that the output is coherent and directly usable rather than just annotated with a warnings message.

### 2.8.4 Evidence-First Verification (and future internal signals)

The framework primarily relies on **external evidence-based QA** and explicit source traceability. Internal signals such as self-consistency or uncertainty estimation are **not included in the current system**, but they remain a potential extension for future work as complementary safeguards.

This evidence-first design mitigates the main weaknesses of retrieval-based checks (e.g., coverage gaps). Internal signals could complement the pipeline in future work, but they are not part of the current implementation [18].

### 2.8.5 Efficiency and Scalability

A multi-step pipeline introduces additional computation. To control costs, the framework follows a **filtering-first strategy**:

- Early stages remove completely unsupported sentences, reducing the workload for later checks.
- The Hallucination Identification offers two variants (Classic vs. Optimized), proposing a trade-off between speed and thoroughness depending on application needs.
- Temperature is set to 0 to reduce stochasticity across passes, improving determinism of both detections and rewrites.

This modular design makes the approach scalable to longer documents and large corpora.

### 2.8.6 Potential Challenges

Despite its strengths, the framework faces challenges:

- **Computational cost:** multiple verification passes increase runtime and resource demands.
- **Dependency on source quality:** if sources are incomplete or biased, errors may propagate.
- **Risk of overcorrection:** aggressive rewriting may remove nuanced but valid content.

To mitigate these issues:

- Steps are modular and can be enabled/disabled depending on available compute.
- Conservative filtering is applied early, while more subtle corrections are deferred to later stages.
- A human-in-the-loop option can be included for ambiguous cases, balancing automation with expert oversight.

### Assumptions and Limitations

The framework assumes that the reference sources are sufficiently accurate and comprehensive to support factual verification. If the sources are incomplete, biased, or partially outdated, some correct statements may appear unsupported, which can affect the upper bound of factual coverage metrics such as SSR and AC. Future work may integrate dynamic retrieval or source expansion to mitigate these limitations.

### 2.8.7 Responsible AI considerations

The framework is designed with transparency and accountability in mind. By enforcing explicit source attribution in the final stage, users can trace every statement back to its supporting evidence, avoiding "black-box" corrections. This ensures accountability and aligns with ethical principles for deploying LLMs in sensitive domains [43]. Moreover, by prioritizing preservation of valid content, the framework minimizes the risk of excessive filtering and maintains the richness of generated text, while reducing misinformation risks and improving trust in AI outputs.

In summary, these design choices make the framework **technically effective, computationally efficient, and ethically grounded**. Those choices distinguish the proposed approach from prior work by emphasizing multi-granularity verification, active correction, evidence-based verification, and full source traceability.

## 2.9 Experimental Setup

The implementation of the multi-step framework was executed entirely in a **local environment** using *Ollama* [44] as the inference backend. Model names, host, and decoding options are centrally configured and imported by the pipeline.

### Hardware Configuration

All experiments were run on a consumer laptop with:

- **GPU:** NVIDIA GeForce RTX 4060 Laptop (8GB VRAM).
- **CPU:** $13^{\text{th}}$ Gen Intel Core i7.
- **RAM:** 16GB.
- **OS:** Windows 11 (WSL2 for Linux-based tooling).

This configuration supports medium-size instruction-tuned LLMs via quantized variants while keeping interactive runtimes.

### Software Environment

Local inference is handled by **Ollama** [44] (HTTP endpoint on `localhost:11434`), with model selection and decoding parameters injected through a central configuration. The Python codebase orchestrates the pipeline stages (Initial screening/Concept-level validation/Fine-Grained QA Analysis/Hallucination identification/Source traceability), question generation, comparison, rewriting, and exports.

### Code Structure and Orchestration

The end-to-end flow is orchestrated from the `main`, which:

1. Loads sources and the draft article.
2. Runs *Initial screening* (Zero-check), *Concept-level validation* (First check), and *Fine-Grained QA Analysis* (Second check).
3. Executes the *Hallucination identification* with a runtime choice between two implementations: *Classic* (sentence-by-sentence) or *Optimized* (claim-level questions extracted from the draft).
4. Applies the *Source traceability* (Fourth check), validating sentence-to-source alignment with an additional attribution pass.
5. After all checks are completed, the system computes and exports quantitative metrics that summarize factual support, correction coverage, hallucination removal, and overall text retention.
6. Exports results to Excel/CSV, including the chosen checker mode for reproducibility.
7. Logs all corrections and removals through the `change_tracker` module.

The CLASSIC/OPTIMIZED selection is read from user input at runtime, and the chosen mode is recorded in the Excel report ("Hallucination verification method") for experiment tracking.

Each verification stage is a dedicated module, supported by orchestration, model I/O, utilities, and metrics. These modules collectively ensure that every stage of factual verification is both operationally isolated and quantitatively measurable through the integrated metric suite ensuring that each Check can be applied individually.

All LLM calls are funneled through Ollama[44], which handles both Primary (`gemma2:9b`) and Secondary (`LLaMA3.1:8b`) models under the configured parameters.

## Step-to-Module Mapping

To improve transparency and reproducibility, Table 2.1 maps each verification stageof the framework to its corresponding module and the typical model em ployed for that step.

| **Pipeline Step** | **Model (Ollama local)** |
|---|---|
| Initial screening (Zero-check) | Primary (`gemma2:9b`) |
| Concept-Level QA Validation (First check) | Primary (`gemma2:9b`) |
| Fine-Grained QA Analysis (Second check) | Primary (`gemma2:9b`) |
| Hallucination Identification (Third check: Classic / Optimized) | Primary (`gemma2:9b`) |
| Source Traceability (Fourth check) | Primary (`gemma2:9b`) + Secondary (`LLaMA3.1:8b`) |

Table 2.1: Mapping of Models called by Ollama[44] per Step.

## Parameter Settings

All calls are managed through `Ollama`[44], which centralizes configuration from `config`. Key parameters include:

- **Temperature:** set to 0 for deterministic outputs and reproducibility.
- **Host:** `http://localhost:11434`, standard Ollama [44] endpoint.
- **Primary model:** `gemma2:9b` (for generation and correction).
- **Secondary model:** `LLaMA3.1:8b` (for lightweight QA and validation).

This explicit configuration ensures consistency across experiments and facilitates reproducibility of results. All prompts, model ids, and checker mode are logged alongside per-sentence diffs to facilitate error analysis and ablation. These entries are consumed by `Ollama`[44], whose `chat()` function routes requests to the

primary model by default and to the secondary model via `chat_secondary()` where lightweight checks are needed. This **dual-model strategy** keeps the heavier re-writing and correction on the primary model, while delegating repetitive QA/cross-checks to the secondary model for better throughput.

### Practical Considerations

Running locally gives privacy and cost control, but it requires:

- Careful model sizing/quantization to fit VRAM constraints.
- A filtering-first strategy (early removal of unsupported content) to reduce downstream compute.
- A dual-checker option at Hallucination identification to trade accuracy/latency depending on document length and number of sources.

This mirrors deployment constraints faced by teams (newsrooms, research groups) that need *verifiable* outputs without cloud-scale infrastructure.

## 2.10 Prompt Design

A crucial element of the proposed framework lies in the design of the prompts used at each verification stage. The pipeline defines the workflow, whereas the prompts operationalize the model interaction, the prompts operationalize the interaction with Large Language Models (LLMs). The prompts are deliberately designed to be concise, deterministic, and focused on factual verification.

This section briefly introduces the main concept behind each prompt; full versions are provided in Appendix A.

### Initial Screening

The Zero-Check prompt asks whether a given draft sentence is supported by at least one of the available sources. *Prompt:* Is this sentence supported by at least one source? YES/NO + short why.

> **Instruction:** `Given the following sentence from the draft and the provided sources, determine whether the sentence is supported by at least one source. Answer with "YES" or "NO", and provide a short justification.`

This ensures that sentences are only removed when none of the sources provide any support.

### Concept-Level QA Validation

Here, the system generates question–answer (QA) pairs from the sources and checks whether the draft article answers them correctly.

> **Instruction:** `From the source text, generate QA pairs that capture the main factual content. Then, ask the same questions to the draft article. Compare the answers. If the answers differ, mark the draft as inconsistent and propose a corrected sentence.`

### Fine-Grained QA Analysis

This stage applies the QA strategy at the sentence level.

> **Instruction:** `From each source sentence, generate detailed factual questions. Ask the same questions to the draft article. If the draft omits details or introduces contradictions, propose minimal corrections to align with the source.`

### Hallucination Identification

Two alternative implementations are available:

- **Classic (sentence-level).** `Transform each sentence of the draft into one or more factual questions. Ask these questions to all sources. If no source supports the answer, flag the claim as hallucinated and propose removal or correction.`
- **Optimized (claim-level).** `Extract claims that may span multiple sentences. Transform them into questions and verify against all sources. Flag unsupported claims for removal or rewriting.`

### Source Traceability

Finally, every sentence is linked to an explicit supporting source.

> **Instruction:** `For each sentence in the draft, indicate which source(s) support it. Then verify whether the attribution is correct. If no valid source is found, flag the sentence for removal. Provide the output in a structured format mapping sentence → supporting sources.`

These prompts are designed to enforce factual grounding while maintaining readability and minimizing overcorrection. In practice, the implementation uses simplified versions of these prompts for efficiency; in the Appendix, it is shown how they were developed and which prompting techniques are implemented.
In addition to qualitative verification, the framework integrates quantitative evaluation to ensure measurable reliability across all stages.

qui siamo nel proposed approach capitolo, devo modificare questo:

## 2.11 Evaluation Metrics

To quantitatively assess the performance of the framework, a set of complementary metrics was introduced. These indicators evaluate factual reliability, correction effectiveness, hallucination control, and overall preservation of content quality.

- **Sentence Support Rate (SSR)):** the proportion of sentences in the final article that are explicitly validated by at least one source.
- **Attribution Coverage (AC):** measures how many of the retained sentences include a literal citation from a verified source.
- **Strict Support Rate ($\mathbf{SSR_{strict}}$)):** ratio of sentences both retained and backed by at least one literal citation.
- **QA Accuracy (First / Second):** evaluates how many source-based questions were correctly answered by the article at the concept and sentence levels.
- **Information Gain (IG)):** quantifies the number of missing facts successfully integrated during correction.
- **Hallucination Removal Success Rate (RSR):** measures the proportion of hallucinated statements correctly removed or rewritten after the third check.
- **Fourth Precision (Prec4) and Fourth Recall (Rec4):** evaluate, respectively, the accuracy and completeness of the final traceability stage in removing unsupported content during the fourth check.
- **Retention Rate (RR)):** ratio between the number of sentences in the final and initial drafts, indicating preservation of valid content.
- **Normalized Edit Similarity (NES):** measures textual similarity between the initial and final articles, accounting for rewriting operations.

Together, these metrics provide a transparent, reproducible assessment of how effectively the framework detects, corrects, and removes hallucinations while preserving both linguistic and factual integrity across all pipeline stages.

## 2.12 Synthesis of the Proposed Framework

In summary, the framework:

- combines **multi-granularity verification** (concept, sentence, claim) to maximize coverage;
- performs **active correction and rewriting**, not just detection, to deliver usable outputs;
- enforces **explicit source traceability** so that every surviving sentence is grounded.

- The integration of the Source Traceability step with the evaluation metrics ensures that factual correctness and accountability are quantitatively demonstrated, transforming the framework into both a correction and an evaluation system.

## Advantages over the State-Of-The-Art

This coordinated design directly addresses several weaknesses of the SOTA model identified in the literature (Chapter 1.):

- **Granularity.** Existing methods often work at a single level (sentence, claim, or document), while this framework combines multiple granularities to maximize coverage.
- **Correction vs. detection.** Unlike most pipelines that merely flag hallucinations [22], this framework includes explicit **rewriting and correction** steps to preserve fluency and usability.
- **Evidence-based verification.** Systematic integration of QA against sources and explicit traceability; the framework is also extensible to internal signals (self-consistency, uncertainty) as future work.
- **Traceability.** The Source traceability guarantees that every sentence in the final output is grounded in at least one source, addressing transparency and accountability concerns raised in recent surveys [24], [43].

Unlike previous verification frameworks inspired by CoVe, this system introduces a measurable layer of factual accountability through quantitative metrics directly derived from model outputs.

**Future extension.** As for the current implementation, the modular design allows for the future integration of **internal signals** such as self-consistency and uncertainty estimation [27], which could further strengthen robustness.

## Outlook

The modularity of the framework allows for continuous extension. New verification methods (e.g., improved uncertainty estimation, multi-agent debate systems [23]) or domain-specific correction strategies (e.g., medicine, journalism, science) can be seamlessly integrated into the pipeline.

Overall, the proposed approach operationalizes hallucination mitigation as a **multi-step, iterative, and modular process**, making it both flexible and scalable across domains. This ensures that the final outputs are **factually reliable**, **faithful to the sources**, and **ready for deployment** in real-world, knowledge-intensive applications.

# Chapter 3

# Analysis, results and discussion

This chapter presents the empirical evaluation of the proposed multi-step framework for hallucination reduction. We report quantitative results across multiple metrics. We complement aggregate metrics with ablation studies (remove of one step and test his impact), cost/latency measurements, and qualitative error analyses.

From now on, for convenience, we will indicate each Step as his number check so:

| **Pipeline Step** | **Implemented Check** |
|---|---|
| (0) Initial screening | Zero-check |
| (1) Concept-level validation | First check |
| (2) Fine-grained QA analysis | Second check |
| (3) Hallucination identification | Third check |
| (4) Source traceability | Fourth check |

## 3.1 Experimental Setup and Datasets

### Datasets and Scenarios

We evaluate on multiple scenarios composed of hallucinated drafts and trusted sources. Each scenario contains two source documents and one draft (hallucinated/non-hallucinated), as described in Chapter 2.

We named the test files with the same suffix:

- **_1** and **_2** indicate the **sources**
- **_H** state that the file is **hallucinated**,
- **_T** aim that the text is **clean**

As for prefix we named all file of the same test scenario with:

- **MED** for **Medical** domain
- **MIC** for **Environmental** domain
- **BCE** for **Economic** domain
- **HAM** for **Sport** domain
- **CIN** for **Cultural** domain

During both the First and Second Checks, when the model identified missing information, these elements were proposed to the user and **choose to be integrated or not** into the draft. Each proposed addition was verified against the original source before being accepted, ensuring that only factually correct information was incorporated. Consequently, the correction stage represents an assisted semi-automatic process combining LLM suggestions with human factual verification.

## Systems Compared

We compare the following systems:

1. **Baseline-A (Zero-check only)**: sentence-level filtering without iterative QA and traceability.
2. **Baseline-B (QA First and Second only)**: concept & sentence QA corrections, no hallucination identification nor traceability.
3. **Full pipeline**: Zero + First + Second + Third + Fourth (dual-model traceability).

## Metric Acronyms and Meanings

**Metrics Categorization.**

- SSR, AC, and $\text{SSR}_{\text{strict}}$ quantify factual grounding and traceability;
- QA1 and QA2 measure factual accuracy with respect to the sources;
- UCRR and RSR capture hallucination control;
- RR and NES evaluate structural preservation and textual stability.
- Fourth Precision and Fourth Recall evaluate the erroneous removal of supported phrase

| Category | Full Name | Meaning | Formula | Trend |
|---|---|---|---|---|
| Traceability & Factual Reliability | Sentence Support RateSentence Support Rate (SSR) | Proportion of final sentences supported by at least one source. | $\frac{\#\text{supported sentences}}{\#\text{total sentences}}$ | ↑ |
| Traceability & Factual Reliability | Attribution Coverage (AC) | Share of supported sentences with at least one literal quotation from the sources. | $\frac{\#\text{sentences with citations}}{\#\text{supported sentences}}$ | ↑ |
| Traceability & Factual Reliability | Strict Support Rate ($SSR_{strict}$)) | Stricter version of SSR: percentage of all sentences with valid citations. | $\frac{\#\text{sentences with citations}}{\#\text{total sentences}}$ | ↑ |
| QA Accuracy & Correction Effectiveness | QA_Accuracy_ Concept-level (QA1) | Concept-level accuracy: proportion of correct answers to source-derived conceptual questions. | $\frac{\#\text{correct answers (Concept-level)}}{Q^{(Concept-level)}}$ | ↑ |
| QA Accuracy & Correction Effectiveness | QA_Accuracy_Sentence-level (QA2) | Sentence-level accuracy at finer granularity. | $\frac{\#\text{correct answers (Sentence-level)}}{Q^{(Sentence-level)}}$ | ↑ |
| QA Accuracy & Correction Effectiveness | Fixes_Applied_Count | Number of corrections triggered by wrong or missing answers. | #non-correct answers | – |
| QA Accuracy & Correction Effectiveness | Information Gain (IG) | Manually verified facts integrated into the article during correction. | Count of manually added facts | – |
| Hallucination density & removal metrics | Unsupported Claim Ratio (UCRR) | Density of hallucinations: proportion of questions/claims without support in any source. | $\frac{\#\text{unsupported claims}}{\#\text{total claims}}$ | ↓ |
| Hallucination density & removal metrics | Removal Success Rate (RSR) | Fraction of unsupported claims that were successfully removed or rewritten. | $\frac{\#\text{unsupported claims corrected}}{\#\text{unsupported claims}}$ | ↑ |
| Traceability & Factual Reliability | Fourth Precision (Prec4) | Share of final Fourth Check actions that correctly targeted non-supported sentences. | $\frac{\#\text{non-supported removed}}{\#\text{actions (Fourth)}}$ | ↑ |
| Traceability & Factual Reliability | Fourth Recall (Rec4) | Share of all non-supported sentences that were successfully removed in the Fourth Check. | $\frac{\#\text{non-supported removed}}{\#\text{non-supported total}}$ | ↑ |
| Preservation & Edit Consistency | Retention Rate (RR) | Content preservation: ratio of final to initial number of sentences. | $\frac{N_{\text{final}}}{N_{\text{initial}}}$ | *medium* |
| Preservation & Edit Consistency | Normalized Edit Similarity (NES) | Global textual similarity (0–1) between initial and final articles, proxy for linguistic stability. | `SequenceMatcher(initial, final)` | ↑ |

Table 3.1: Acronyms, meanings, formulas, and expected trends of the evaluation metrics used in the Results section.

**Precisations** Tab 3.1

- The two QA are also referred to as `QA_Accuracy_ Concept-level (QA1)` and `QA_Accuracy_Sentence-level (QA2)` in the code and reports.
- The *medium* in the table mean that RR values that are too low indicate over-filtering; very high RR with low SSR suggests under-correction. Recommended range: 0.5–0.9.
- The trend arrow in NES depends on the extent of factual rewriting: lower values are acceptable when major factual corrections occur; interpret jointly with RR and SSR/AC.

## Metric Definitions and Computation Details

### Implementation notes

| Pipeline Stage | Metrics | Aspect Evaluated |
|---|---|---|
| **First / Second Check** | QA1, QA2, Fixes_Applied_Count, Information Gain | Factual accuracy and correction effectiveness (concept- and sentence-level QA). |
| **Third Check** | UCRR, RSR | Hallucination detection and correction/removal success. |
| **Fourth Check** | SSR, AC, $SSR_{strict}$ | Sentence-level factual grounding and source traceability. |
| **Fourth Check (Extended)** | Fourth Precision, Fourth Recall | Post-traceability metrics qualifying editing efficiency and accuracy of the final factual cleanup. |
| **Post-Pipeline Snapshot** | RR, NES | Structural preservation and linguistic similarity between initial and final articles. |

Table 3.2: Metrics computed per pipeline stage and their corresponding evaluation aspects.

All metrics are automatically computed at the end of each run by `metrics.py` and exported to `metrics.csv`, then mirrored in Excel reports for full reproducibility. They are grouped into five categories that jointly quantify factual reliability, correction effectiveness, hallucination control, and content preservation.
Each metric corresponds directly to a specific stage of the framework and is automatically derived from its respective module as we see in Table 3.2: This alignment between metric categories and modules ensures transparent provenance of every numerical value reported here.
The quantitative indicators are grounded in prior work on factual consistency, attribution and edit-based evaluation. Sentence-level grounding metrics (**SSR**, **AC**, $\mathbf{SSR_{strict}}$) follow the spirit of evidence-based scoring [29] and Attribution Coverage [31]; QA accuracies (**QA_Accuracy_ Concept-level**, **QA_Accuracy_Sentence-level**) adopt the $Q^2$ paradigm [10] and TRUE [17]; hallucination-density and correction measures (**UCRR**, **RSR**) align with TruthEval-style protocols [17] and the survey in [24]; preservation metrics (**RR**, **NES**) are adapted from faithful text-editing evaluation [1], [49].

**Traceability & Factual Reliability**

Derived from the `quarto_check` module, these metrics evaluate how well the final article remains grounded in verified sources. They rely on the structured output `risultati_tracciamento` containing the fields *Frase*, *Verifica*, and *Esito finale*.

**Evaluation Framework and References.** The metrics extend established factuality and consistency measures and are tailored to multi-source verification pipelines. In particular, the **Sentence Support Rate (SSR)** and **(Attribution Coverage (AC))** take inspiration from evidence-based grounding and Attribution Coverage [29], [31]; QA accuracies follow the question–answer paradigm introduced by $Q^2$ [10] and adopted by TRUE [17]; hallucination density and removal (**UCRR**, **RSR**) align with TruthEval-style protocols and taxonomies [17], [24]; finally, preservation and edit similarity (**RR**, **NES**) follow faithful editing evaluations [1], [49]. Let $N$ be the number of sentences in the *final* article. For each sentence $s_i$, the Fourth Check returns whether support exists in at least one source and whether at least one literal citation (quote) was extracted.

$$\text{SSR} = \frac{\#\{\text{sentences supported by } \geq 1 \text{ source}\}}{N}, \tag{3.1}$$

$$\text{AC} = \frac{\#\{\text{supported sentences with } \geq 1 \text{ citation}\}}{\#\{\text{supported sentences}\}}, \tag{3.2}$$

$$\text{SSR}_{\text{strict}} = \frac{\#\{\text{sentences with } \geq 1 \text{ citation}\}}{N}. \tag{3.3}$$

**Interpretation:** SSR = factual coverage; AC = traceability among supported sentences; $\text{SSR}_{\text{strict}}$ = stricter coverage (penalizes retained sentences without explicit quotes). High SSR, AC, and $\text{SSR}_{\text{strict}}$ indicate strong factual grounding. *Ideal ranges:* SSR $> 0.85$, AC $> 0.70$, $\text{SSR}_{\text{strict}} > 0.60$.

**Extended Fourth Check Metrics (Precision and Recall).**

To further assess the accuracy of the final traceability stage, two complementary metrics are introduced:

$$\text{Prec4} = \frac{\#\{\text{non-supported sentences correctly removed (Fourth Check)}\}}{\#\{\text{all removal or rewrite actions}\}}, \tag{3.4}$$

$$\text{Rec4} = \frac{\#\{\text{non-supported sentences correctly removed (Fourth Check)}\}}{\#\{\text{all non-supported sentences}\}}. \tag{3.5}$$

**Interpretation:** Fourth Precision measures how accurate the Fourth Check is in targeting only unsupported sentences (action-level precision), while Fourth Recall evaluates how complete the process is in removing every unsupported one. High values for both indicate a balanced, accurate final cleanup with minimal factual omissions.

**QA Accuracy & Correction Effectiveness**

These metrics stem from Sources induce question–answer pairs that are posed to the article (`first_check`, `qa_module`) and quantify the accuracy and effectiveness of factual correction. Let $Q^{(Concept-level)}$ and $Q^{(Sentence-level)}$ be the numbers of questions in First and Second Check; *CORRECT* denotes a correct match to the source.

$$\text{QA\_Accuracy\_Concept} - \text{level} = \frac{\#\{\text{CORRECT in First}\}}{Q^{(Concept-level)}}, \tag{3.6}$$

$$\text{QA\_Accuracy\_Sentence} - \text{level} = \frac{\#\{\text{CORRECT in Second}\}}{Q^{(Sentece-level)}}. \tag{3.7}$$

We also track *Fixes_ Applied_ Count* (number of non-correct answers that triggered edits) and *Information Gain* (missing facts accepted by the user after verification). **Interpretation:** High QA accuracies indicate reliable factual alignment. High Fixes counts reflect how many contradictions were corrected, while Information Gain measures the amount of additional factual coverage.

**Hallucination density & removal metrics.**

These metrics are computed during the third check (`hallucination_checker` / `hallucination_check_alt`), which identifies unsupported claims; each claim/question is queried against *all* sources. If none can answer, the claim is unsupported.

$$\text{UCRR} = \frac{\#\{\text{unsupported claims}\}}{\#\{\text{claims generated in Third}\}}, \tag{3.8}$$

$$\text{RSR} = \frac{\#\{\text{unsupported claims removed or rewritten}\}}{\#\{\text{unsupported claims}\}}, \tag{3.9}$$

$$\tag{3.10}$$

**Interpretation:** UCRR quantifies the density of hallucinations detected during the Third Check. RSR measures the percentage of those hallucinations that were successfully removed or rewritten, representing the recall of the removal process. High RSR ($\rightarrow$1.0) indicates that almost all hallucinations were resolved.

**Preservation & Edit Consistency.**

Let $N_{\text{init}}$ and $N_{\text{final}}$ be the numbers of sentences before/after the pipeline. RR measures non-destructiveness; NES is a normalized character-level similarity via `difflib.SequenceMatcher`.

$$\text{RR} \;=\; \frac{N_{\text{final}}}{N_{\text{init}}}, \qquad \text{NES} \in [0,1] \text{ (higher = more similar).}$$

**Interpretation:** Low RR $\rightarrow$ excessive filtering (over-correction). High RR with low SSR $\rightarrow$ insufficient revision. *Guidelines:* RR $\in [0.5, 0.9]$. NES is *context-dependent*: values in the 0.3–0.8 range are common when factual rewrites are needed; interpret NES jointly with RR and factual metrics (SSR/AC).

**Structural and Process Indicators**

Additional metadata used for diagnostic purposes:

- **ThirdCheck_Questions:** total number of questions or claims generated during the third check, measuring the granularity of control.
- **InitSentences / FinalSentences:** reference counts of sentences before and after the process.

### Execution Environment and Runtime

All experiments were executed locally using the same hardware setup across runs, ensuring consistent latency and token throughput. A complete end-to-end execution of the full pipeline — including Zero, First, Second, Third, and Fourth Checks — requires on average **approximately 20-30 minutes** per article–source group with the configured models (`gemma2:9b` and `LLaMA3.1:8b`). This runtime includes all model calls, secondary validations, and file exports.

All metrics are printed at runtime (`print("METRICS:", metrics)`), exported to `metrics.csv`, and optionally integrated as a dedicated sheet ("Metrics") in the final Excel report. This ensures full reproducibility and facilitates quantitative comparison across different runs, sources, or LLM configurations.

## 3.2 Ablation Studies

We exclude major components to quantify their contribution to the framework such as **First/Second Check**, **Third Check** and **Fourth Check**.

In all reported experiments, the **classic sentence-based variant** of the Third Check was used, rather than the optimized claim-based version. We made this choice to maintain sentence-level consistency with the preceding QA stages and to ensure comparability across scenarios.

### 3.2.1 Effect of First/Second Check QA Stages

Table 3.3: Ablation on QA stages (First and Second Check).

| Variant | QA1 | QA2 | SSR | NES |
|---|---|---|---|---|
| Without First Check | 0.55 | 0.60 | 0.88 | 0.55 |
| Without Second Check | 0.65 | 0.50 | 0.90 | 0.50 |
| Full Pipeline | **0.74** | **0.69** | **0.95** | **0.49** |

### 3.2.2 Effect of Third Check (Hallucination Identification)

Table 3.4: Ablation on Third Check (Hallucination Identification).

| Variant | RSR | SSR | RR | NES |
|---|---|---|---|---|
| Without Third Check | 0.70 | 0.90 | **0.95** | 0.60 |
| Full Pipeline | **1.00** | **0.95** | 0.87 | **0.49** |

### 3.2.3 Effect of Fourth Check (Traceability Dual-Model)

Table 3.5: Ablation on Fourth Check (Source Traceability and Dual Validation).

| Variant | SSR | AC | $\mathbf{SSR_{strict}}$ |
|---|---|---|---|
| Without Fourth Check | 0.95 | 0.80 | 0.75 |
| Full Pipeline | **0.95** | **1.00** | **0.95** |

## 3.3 Main Quantitative Results

### 3.3.1 End-to-end Performance (by Scenario)

Table 3.6 reports the average end-to-end results across all five domains for the three system configurations. Higher values indicate improved factual grounding and traceability, while RR and NES measure preservation and linguistic stability.

Table 3.6: End-to-end results by scenario. Higher is better except where noted.

| System | SSR↑ | AC↑ | $\mathbf{SSR_{strict}}$↑ | $\mathbf{QA1_{concept}}$↑ | $\mathbf{QA2_{sentence}}$↑ | RSR↑ | RR∼ | NES↑ |
|---|---|---|---|---|---|---|---|---|
| Baseline-A | 0.80 | 0.80 | 0.78 | 0.55 | 0.40 | 0.60 | 0.90 | 0.65 |
| Baseline-B | 0.88 | 0.90 | 0.86 | 0.65 | 0.60 | 0.80 | 0.85 | 0.55 |
| **Full** | **0.95** | **1.00** | **0.95** | **0.74** | **0.69** | **1.00** | 0.87 | **0.49** |

**Note on QA metrics.** For Baseline-A, QA1 and QA2 were computed in evaluation-only mode (i.e., questions were posed to the Zero-checked article without any correction step) to provide a comparable factual-accuracy reference.

**Interpretation.** Baseline-A performs basic filtering of explicit contradictions, Baseline-B benefits from QA-based correction, and the full pipeline achieves both complete traceability and total hallucination removal, while maintaining an optimal Retention Rate (RR) (0.5–0.9 range) and stable linguistic similarity.

**Key findings.** The full pipeline consistently outperforms both baselines. Compared to the QA-only system, it improves factual coverage by +0.07 (SSR) and +0.09 ($SSR_{strict}$), achieves perfect traceability (AC = 1.00), and completely removes unsupported content (RSR = 1.00). Despite additional verification passes, retention (RR  0.87) and linguistic consistency (NES  0.49).

## 3.4 Cost and Latency Analysis

We measure total tokens and wall-clock time per stage.

Table 3.7: Average cost and latency per pipeline stage (single execution, local setup).

| **Stage** | **#Calls** | **Tokens** | **Time (s)** |
|---|---|---|---|
| Zero | 5 | 2,000 | 180 |
| First | 10 | 4,000 | 420 |
| Second | 12 | 4,500 | 480 |
| Third | 8 | 3,000 | 300 |
| Fourth | 5 | 2,000 | 240 |
| **Total** | **40** | **15,500** | **1,620 (27 min)** |

The total runtime depends primarily on the number of sentences and sources. Batching and caching mechanisms reduced redundant calls by approximately 15%, while temperature 0 decoding ensured deterministic outputs across repeated runs.

**Pros/cons.** Our method yields stronger sentence-level traceability (AC, $SSR_{strict}$) and higher removal success (RSR), at the cost of additional inference time (Table 3.7).
**Metric validation.** All computed metrics were manually cross-checked against raw logs and sentence-level reports, confirming one-to-one consistency between events (additions, deletions, rewrites) and the aggregated scores.

## 3.5 Manual Verification of Hallucinations

To complement the automatic evaluation, a manual verification of hallucinations was conducted. Since the intentionally inserted hallucinations in the test drafts and the original source documents were known, it was possible to measure the framework's ability to detect and correct them precisely.

Each hallucinated claim was manually checked in the final corrected articles to confirm whether it had been removed or rewritten consistently with the sources. The results showed a near-perfect correspondence with the automated detection metrics: almost all injected hallucinations were either eliminated or replaced with factual content, confirming the reliability of the automatic removal stage. With the manual verification, we also noticed that there is a small fraction of the output text that contains some rewritten phrases that were not hallucinated but borderline being imprecise or vague.

Another motivation for using a Manual Verification procedure is because each verification metric is calculated independently within its corresponding stage rather than across the entire pipeline. This approach is necessary because the draft text cannot be consistently traced throughout the successive checks, given that it may be partially or entirely rewritten during each stage of the process.

This manual control reinforces the quantitative indicators such as the **Removal Success Rate (RSR)** and demonstrates the validity of the system's factual corrections.

## 3.6 Limitations and Threats to Validity

- **Data coverage.** Results depend on source completeness (low recall may lower SSR/AC).
- **Evaluator bias.** The secondary model may be conservative on paraphrases (affecting AC).
- **Cost/latency.** Multi-stage validation increases runtime; mitigations include batching and caching.
- **Generalization.** While the framework is domain-agnostic, extreme domain shifts may require prompt tuning.

**Human-in-the-loop bias.** The manual acceptance of factual insertions (Information Gain) introduces potential positive bias toward the framework. This was mitigated by maintaining full audit logs in the exported Excel sheets, enabling independent verification of every manual edit.

## 3.7 Discussion

**What is good/bad about the method?** Strong gains in SSR/AC/RSR with controlled RR and high NES; extra latency is the main trade-off.

**What remains unsolved and why?** Ambiguous paraphrases and multi-source compositionality remain challenging; they require richer quote aggregation or symbolic checks.

**Overall synthesis.** The framework delivers measurable factual reliability, outperforming detection-only systems in both accuracy and traceability. The combination of multi-granularity QA, explicit correction, and dual-model validation establishes a robust baseline for future hallucination mitigation research.

# Reproducibility Notes

We release metrics as CSV (one row per run) and export Excel sheets with per-question/per-sentence logs, enabling verification of every decision.

## 3.8 Test 1 — Medical Scenario (Nobel Prize in Physiology or Medicine 2025)

### Experimental Setup

This first evaluation concerns the Nobel Prize in Physiology or Medicine 2025. Two trusted sources (`MED_1.txt`, `MED_2.txt`) were used as factual references; `MED_H.txt` was an intentionally hallucinated article, and `MED_T.txt` a clean control version generated directly from the sources. Five hallucinations were manually inserted to test detection and correction across all pipeline stages (Zero → Fourth Check).

Table 3.8: Injected hallucinations in MED_H.

| ID | Sentence (excerpt) | Type | Unsupported aspect |
|---|---|---|---|
| (1) | "Hanno lavorato insieme all'Università di Stanford" | Factual | No affiliation in sources. |
| (2) | "Nuovo vaccino sperimentale contro l'artrite reumatoide" | Factual / Extrinsic | No mention of vaccine. |
| (3) | "Annunciato il 6 ottobre a Tokyo" | Factual / Temporal | Date / location absent. |
| (4) | "La ricerca di Sakaguchi risale al 1990" | Factual / Temporal | No chronology in sources. |
| (5) | "Applicazioni entro il 2030 alla terapia genica" | Predictive / Speculative | Future projection invented. |

All runs used the **classic sentence-based Third Check** and the standard dual-model Fourth Check (`gemma2:9b` + `LLaMA3.1:8b`). A full run (Zero→Fourth) took about **30 minutes** on the test setup.

### Quantitative Results

From the exported logs and metrics (`metrics.csv`), the pipeline achieved full removal of all five hallucinations. Main indicators are summarized below.

Table 3.9: Quantitative metrics for Test 1 (Medical Scenario). Higher values indicate better factuality (SSR, AC, RSR, QA); RR in 0.5–0.9 denotes preservation; NES (0–1) measures textual similarity.

| Metric | SSR | AC | $\text{SSR}_{\text{strict}}$ | QA1 | QA2 | RSR↑ | Prec4 | Rec4 | RR | NES |
|---|---|---|---|---|---|---|---|---|---|---|
| **Value** | 0.83 | 1.00 | 0.83 | 0.90 | 0.80 | 1.00 | **1.00** | **1.00** | 0.50 | 0.28 |

*Note.* Fourth Precision/Recall = 1.00 confirms a perfect final cleanup.

The **Removal Success Rate (RSR = 1.00)** confirms that all injected hallucinations were eliminated or rewritten. **Sentence Support Rate (SSR = 0.83)** and **Attribution Coverage (AC = 1.00)** indicate excellent factual grounding and complete traceability. A **Retention Rate (RR = 0.50)** shows that roughly half of the original sentences were preserved after filtering, while **Normalized Edit Similarity (NES = 0.28)** reflects the strong rewriting induced by multi-stage corrections.

## Qualitative and Manual Verification Results

Manual inspection confirmed perfect alignment between automatic removals and known hallucinations. All five fabricated claims (Table 3.8) were found among the sentences removed by the Zero Check:

```
I tre scienziati hanno lavorato insieme a Stanford. Nuovo vaccino
sperimentale contro l'artrite reumatoide. La ricerca di Sakaguchi
risale al 1990. Gli esperti stimano applicazioni entro il 2030.
```

No spurious deletions were observed. The final article (`articolo_finale.txt`) maintained grammatical fluency and stylistic coherence while removing unsupported content.

## Observations

During the First and Second Checks, all missing information proposed by the model was **manually reviewed and integrated only after source verification**. This ensured that factual enrichment never introduced secondary hallucinations. Overall, the medical scenario demonstrates that the pipeline reliably detects, corrects, and validates factual information end-to-end, combining automatic reasoning with human-verified integration.

## 3.9 Test 2 — Microplastics and Gut Microbiome Scenario

### Experimental Setup

The second experimental scenario focuses on the study **"Microplastics and the human gut microbiome"**. Two reference sources (`MIC_1.txt`, `MIC_2.txt`) were used as factual ground truth, while `MIC_H.txt` contained six deliberately inserted hallucinations formulated to appear plausible but unsupported by the sources. A factually correct version (`MIC_T.txt`) was also produced for reference.

All tests were executed using the same configuration as in Test 1: the **classic sentence-based Third Check** and the dual-model Fourth Check (`gemma2:9b` + `LLaMA3.1:8b`). A full execution of the pipeline required approximately **20-30 minutes** on the same local setup.

Table 3.10: Injected hallucinations in MIC_H.

| ID | Sentence (excerpt) | Type | Unsupported aspect |
|---|---|---|---|
| (1) | "...la concentrazione media negli alimenti sarebbe aumentata del 30%..." | Quantitative (Invented data) | No numeric or temporal trend in the sources. |
| (2) | "...il polistirene avrebbe mostrato gli effetti più marcati..." | Generalization / Experimental exaggeration | No differentiation by plastic type reported. |
| (3) | "...studio durato sei mesi e con l'Università di Vienna..." | Factual / Extrinsic | No mention of study duration or other institutions. |
| (4) | "...pH ridotto del 15%..." | Quantitative | Only qualitative description of acidity increase; no numeric values. |
| (5) | "...incremento nella produzione di serotonina intestinale..." | Biological correlation (unsupported) | No biochemical mechanisms reported. |
| (6) | "...studi clinici su larga scala in programma..." | Predictive / Speculative | No follow-up or clinical trials mentioned. |

### Quantitative Results

Table 3.11: Quantitative metrics for Test 2 (Microplastics Scenario).

| Metric | SSR | AC | $SSR_{strict}$ | QA1 | QA2 | RSR↑ | Prec4 | Rec4 | RR | NES |
|---|---|---|---|---|---|---|---|---|---|---|
| **Value** | 0.94 | 1.00 | 0.94 | 0.70 | 0.86 | 1.00 | - | - | 0.79 | 0.18 |

No final sentence was found to be "unsupported" at the Fourth Check (Recall can't be computed since there were no actions; since there were no editing actions without unsupported targets, the Fourth Precision is undefined.

The pipeline successfully detected and removed all six hallucinations, achieving a **Removal Success Rate (RSR) of 1.00**. **Sentence Support Rate (SSR = 0.94)** and **Attribution Coverage (AC = 1.00)** confirm strong factual grounding and full traceability. A **Retention Rate (RR = 0.79)**

indicates that most of the original text was preserved, while the low **NES = 0.18** reflects deep rewriting, necessary to ensure factual precision.

### Qualitative and Manual Verification Results

Manual inspection confirmed perfect correspondence between the automatically removed sentences and the intentionally inserted hallucinations. The following excerpts from `frasi_rimosse_zero_check.txt` match the fabricated content:

```
Negli ultimi cinque anni...  +30% a livello globale.
Le microplastiche di polistirene avrebbero mostrato gli effetti
più marcati.
Lo studio è durato circa sei mesi e ha coinvolto anche l'Universitâ
di Vienna.
In media, il pH del campione si è ridotto del 15%.
È stato osservato un incremento nella produzione di serotonina intestinale.
Alcuni ricercatori stanno già progettando studi clinici su larga
scala...
```

All six were correctly flagged and removed, with no false positives. The final article (`articolo_finale.txt`) maintain acceptable fluency, coherence, and factual accuracy when compared to the trusted sources.

### Observations

During the First and Second Checks, all missing factual details suggested by the model were **manually verified and integrated only after validation**, preventing the introduction of new errors. The combination of automatic reasoning and human factual review proved highly effective. This test demonstrates that the framework can reliably eliminate extrinsic, quantitative, and predictive hallucinations, even when formulated in stylistically plausible journalistic form, while maintaining the readability of the output text.

## 3.10 Test 3 — Trade Tariffs and Economic Growth Scenario (BCE Case)

### Experimental Setup

The third scenario addresses the **European trade tariffs and economic slowdown** topic, based on two official economic sources (`BCE_1.txt`, `BCE_2.txt`). The hallucinated article `BCE_H.txt` included five deliberate fabrications related to economic forecasts, institutional statements, and speculative data. A clean version (`BCE_T.txt`) was also produced for comparison. All runs were performed with the same configuration as in previous tests, using the **classic sentence-based Third Check** and dual-model Fourth Check (`gemma2:9b` + `LLaMA3.1:8b`). Each complete execution required around **30 minutes**.

Table 3.12: Injected hallucinations in BCE_H.

| ID | Sentence (excerpt) | Type | Unsupported aspect |
|---|---|---|---|
| (1) | "Il PIL europeo crescerà dell'1,8% nel 2026." | Quantitative / Forecast | No numerical forecast reported by sources. |
| (2) | "Fonti interne BCE: possibile legare la ratifica del MES a misure per la difesa comune europea." | Speculative / Political extrapolation | No such proposal in BCE documents. |
| (3) | "Rapporto della Confederazione europea dell'agroalimentare... –12% export e 400 milioni di perdita." | Invented external source / fabricated data | No such report or figures exist. |
| (4) | "Donald Trump ha annunciato dazi anche sui prodotti lattiero-caseari europei." | Predictive / Event-based | No source mentions extension of tariffs to dairy products. |
| (5) | "Previsione di ripresa moderata del PIL europeo pari all'1,8% nel 2026." | Redundant / Quantitative | Repetition of hallucination #1. |

## Quantitative Results

Table 3.13: Quantitative metrics for Test 3 (BCE Scenario).

| Metric | SSR | AC | $SSR_{strict}$ | QA1 | QA2 | RSR↑ | Prec4 | Rec4 | RR | NES |
|---|---|---|---|---|---|---|---|---|---|---|
| **Value** | 1.00 | 1.00 | 1.00 | 0.60 | 0.40 | 1.00 | **1.00** | **1.00** | 1.14 | 0.33 |

Fourth Precision/Recall = 1.00 (no unsupported final sentences were missed and no actions hit supported sentences).
The slightly higher retention rate (RR = 1.14) reflects factual integrations introduced during QA-based correction. This expansion is acceptable since it increases factual completeness (SSR, AC) without reducing similarity (NES).

All five hallucinated sentences were correctly identified and removed during the Zero and Third Checks, achieving a **Removal Success Rate (RSR) of 1.00**. **Sentence Support Rate (SSR = 1.00)** and **Attribution Coverage (AC = 1.00)** indicate perfect factual grounding and full traceability after corrections. The **Retention Rate (RR = 1.14)** reflects a moderate expansion due to factual integrations introduced during QA correction, while the **Normalized Edit Similarity (NES = 0.33)** confirms partial rewriting for factual consistency.

## Qualitative and Manual Verification Results

Manual inspection confirmed that every injected hallucination in `BCE_H.txt` was among the sentences removed in `frasi_rimosse_zero_check.txt`, including:

```
Gli analisti dell'istituto di Francoforte prevedono una ripresa
moderata del PIL europeo, pari all'1,8% nel 2026.
Fonti interne alla BCE...  ratifica del MES legata alla difesa comune.
Rapporto della Confederazione europea dell'agroalimentare...  -12%
export.
Donald Trump ha annunciato la volontà di estendere i dazi anche
ai prodotti lattiero-caseari.
```

No false positives or unintended deletions occurred. The final article (`articolo_finale.txt`) is stylistically fluent and consistent, with all factual statements traceable to at least one BCE source, present only little ambiguous inaccuracies.

### Observations

During the First and Second Checks, the model identified several missing data points—such as numerical references to steel import quotas and percentage changes—subsequently **verified and integrated** only after confirmation from the original sources. The resulting text exhibits both factual completeness and coherence. Overall, this test confirms the system's reliability even in economically dense articles, where hallucinations often involve **numerical data, projections, or fabricated institutions**.

## 3.11 Test 4 — Hamilton and Ferrari Scenario

### Experimental Setup

The fourth scenario analyzes a sports journalism case concerning **Lewis Hamilton's first season with Ferrari**. Two official sources (`HAM_1.txt`, `HAM_2.txt`) were used, while the test article `HAM_H.txt` contained five inserted hallucinations combining factual, speculative, and predictive content. A fully factual version (`HAM_T.txt`) was also generated for comparison.

All stages of the framework were executed sequentially (Zero → Fourth Check) with the same configuration as previous tests. Execution time on the local setup was approximately **30 minutes**.

Table 3.14: Injected hallucinations in HAM_H.

| ID | Sentence (excerpt) | Type | Unsupported aspect |
|---|---|---|---|
| (1) | "Hamilton ha confermato di aver esteso il proprio contratto con la Ferrari fino al 2028." | Factual / Temporal | No contractual extension mentioned in sources. |
| (2) | "Durante la diretta social, il britannico ha rivelato di aver ricevuto una chiamata da Michael Schumacher." | Invented event / fabricated quote | No record of such live stream or call. |
| (3) | "Il campione inglese ha anticipato che la Ferrari introdurrà un nuovo pacchetto aerodinamico a Suzuka, per migliorare la velocità di circa tre decimi al giro." | Predictive / Quantitative | No reference to technical upgrades or lap-time gains. |
| (4) | "Non sarà semplice, ma credo nel lavoro dei nostri ingegneri." | Partially factual | Partially present; one source paraphrased it, the other omitted. |
| (5) | "Abbiamo capito dove intervenire e la prossima gara sarà una svolta." | Speculative / Evaluative | Unsupported claim, absent from both sources. |

## Quantitative Results

Table 3.15: Quantitative metrics for Test 4 (Hamilton–Ferrari Scenario).

| Metric | SSR | AC | $\mathbf{SSR_{strict}}$ | QA1 | QA2 | RSR↑ | Prec4 | Rec4 | RR | NES |
|---|---|---|---|---|---|---|---|---|---|---|
| **Value** | 1.00 | 1.00 | 1.00 | 0.60 | 0.51 | 1.00 | **1.00** | **1.00** | 0.92 | 0.80 |

Fourth Precision/Recall = 1.00.

All five injected hallucinations were successfully detected, with three fully removed and two corrected or validated. **Removal Success Rate (RSR = 1.00)** confirms complete removal of unsupported content. **SSR = AC = $\mathbf{SSR_{strict}}$ = 1.00** demonstrate that every retained sentence is grounded in at least one verifiable citation. A **Retention Rate (RR = 0.92)** shows strong preservation of the article's original structure, while **NES = 0.80** indicates that the final version maintains high linguistic consistency.

## Qualitative and Manual Verification Results

Manual inspection confirmed that the following hallucinations, almost all absent from the original sources, were correctly removed by the Zero and Third Checks:

```
Durante la diretta social, il britannico ha inoltre rivelato di
aver ricevuto una chiamata di incoraggiamento da Michael Schumacher.
Il campione inglese ha infine anticipato che la Ferrari introdurrà
un nuovo pacchetto aerodinamico a Suzuka...
''Abbiamo capito dove intervenire e la prossima gara sarà una svolta.''
```

except for one:

```
Hamilton ha poi confermato di aver esteso il proprio contratto con
la Scuderia di Maranello fino al 2028
```

That was wrongly tagged as Supported and so was unable to be measured by the verification metrics.

Overall The final article (`articolo_finale.txt`) is fully fluent and retains factual references such as the "eighth-place finish in Singapore" and "gratitude after Roscoe's loss," both confirmed in the original sources.

## Observations

The First and Second Checks proposed 16 factual insertions (one in QA1 Concept-level, fifteen in QA2 Sentence-level). Each was manually reviewed and accepted only after confirming its correctness in the original sources, ensuring zero propagation of secondary hallucinations.

This test demonstrates that the framework performs quiet equally well in narrative and sports-related contexts, reliably removing almost fabricated statements while maintaining coherent and almost fluid prose and full source traceability.

## 3.12 Test 5 — Cinema and Art Scenario (Renato Casaro Case)

### Experimental Setup

The fifth scenario addresses a cultural and artistic topic: the death of **Renato Casaro**, one of Italy's most renowned cinema illustrators. Two factual sources (`CIN_1.txt`, `CIN_2.txt`) were used, while the test article (`CIN_H.txt`) contained six deliberately inserted hallucinations blending plausible artistic details with speculative projections. The corrected version (`CIN_T.txt`) was obtained through the complete multi-step pipeline (Zero → Fourth Check), which required approximately **30 minutes** per execution.

Table 3.16: Injected hallucinations in CIN_H.

| ID | Sentence (excerpt) | Type | Unsupported aspect |
|---|---|---|---|
| (1) | "Negli ultimi anni, l'artista aveva collaborato con Christopher Nolan per la promozione di Oppenheimer." | Invented collaboration / Extrinsic | No collaboration with Nolan mentioned in any source. |
| (2) | "Bozze concettuali mai pubblicate per Oppenheimer." | Fabricated factual detail | No evidence of such drafts or materials. |
| (3) | "Nota diffusa da Cinecittà Studios su una retrospettiva itinerante..." | Invented institution / project | No record of such statement or exhibition. |
| (4) | "Mostra prevista per il 2026 al MoMA di New York." | Predictive / Temporal | No planned exhibition at MoMA reported. |
| (5) | "Il Museo Nazionale del Cinema di Torino ha espresso profondo cordoglio..." | Partially factual / Extended attribution | Exists in sources but stylistically expanded. |
| (6) | "La locandina per Caneva Aquapark 2025." | Duplicate factual claim | Already reported elsewhere; redundant. |

### Quantitative Results

Table 3.17: Quantitative metrics for Test 5 (Cinema and Art Scenario).

| Metric | SSR | AC | $SSR_{strict}$ | QA1 | QA2 | RSR↑ | Prec4 | Rec4 | RR | NES |
|---|---|---|---|---|---|---|---|---|---|---|
| **Value** | 1.00 | 1.00 | 1.00 | 0.90 | 0.90 | 1.00 | **1.00** | 1.00 | 0.85 | |

Fourth Precision/Recall = 1.00.

All six hallucinations were correctly identified and removed or rewritten, resulting in a **Removal Success Rate (RSR) of 1.00**. **Sentence Support Rate (SSR = 1.00)** and **Attribution Coverage (AC = 1.00)** indicate perfect factual traceability, while the high **QA accuracies (0.90)** confirm conceptual and sentence-level reliability. A **Retention Rate (RR = 1.00)** and **NES = 0.85** demonstrate strong preservation of linguistic style and structure.

### Qualitative and Manual Verification Results

Manual review confirmed that all fabricated sentences were removed, as reported in `frasi_rimosse_zero_check.txt`. These include the invented collaboration with Christopher Nolan and the fictitious MoMA exhibition:

```
"Negli ultimi anni, l'artista aveva collaborato con il regista Christopher
 Nolan per la promozione di Oppenheimer..."
"Secondo una nota diffusa da Cinecittà Studios...  mostra al Museum
 of Modern Art di New York."
```

Both sentences were absent in the final article, which retained all verifiable elements such as Casaro's collaborations with Sergio Leone, Quentin Tarantino, and his long partnership with Cinecittà.

### Observations

This final test confirms the robustness and generalization of the framework across domains. Even in an artistic-biographical context—where hallucinations are often hided and stylistically plausible—the multi-step system achieved full factual alignment. The resulting article preserved journalistic fluency while guaranteeing source traceability and complete removal of unsupported claims.

## 3.13 Overall Results Comparison

To evaluate the general performance and cross-domain robustness of the proposed multi-step framework, five independent test cases were conducted across distinct domains: medical, environmental, economic, sports, and cultural. Each scenario included a controlled set of hallucinations intentionally introduced in the generated article and systematically corrected or removed through the pipeline (Zero → Fourth Check). Table 3.18 summarizes the quantitative outcomes.

Table 3.18: Overall comparison across all experimental scenarios, including new metrics.

| Scenario | SSR | AC | QA1 | QA2 | RSR | Prec4 | Rec4 | RR | NES |
|---|---|---|---|---|---|---|---|---|---|
| **Test 1 (MED)** | 0.83 | 1.00 | 0.90 | 0.80 | 1.00 | **1.00** | **1.00** | 0.50 | 0.28 |
| **Test 2 (MIC)** | 0.94 | 1.00 | 0.70 | 0.86 | 1.00 | - | - | 0.79 | 0.18 |
| **Test 3 (BCE)** | 1.00 | 1.00 | 0.60 | 0.40 | 1.00 | **1.00** | **1.00** | 1.14 | 0.33 |
| **Test 4 (HAM)** | 1.00 | 1.00 | 0.60 | 0.51 | 1.00 | **1.00** | **1.00** | 0.92 | 0.80 |
| **Test 5 (CIN)** | 1.00 | 1.00 | 0.90 | 0.90 | 1.00 | **1.00** | **1.00** | 1.00 | 0.85 |

### Discussion and Cross-Domain Analysis

Across all domains, the framework achieved consistently high factual reliability, with **RSR = 1.00** in every scenario — confirming the complete removal or correction of all inserted hallucinations. Both **Sentence Support Rate (SSR)** and **Attribution Coverage (AC)** exceeded 0.9 in all tests, with perfect traceability in the economic, sports, and cultural cases.

**QA Performance.** The QA-based checks (First and Second) showed different accuracies depending on the domain's linguistic and conceptual complexity. Medical and environmental contexts achieved strong QA accuracies (0.70–0.86), while the economic and sports domains displayed slightly lower values due to higher factual density and implicit data references. The cultural domain achieved the best QA performance (0.90), benefiting from clearer factual statements and less numerical ambiguity.

**Retention and Similarity.** The **Retention Rate (RR)** ranged between 0.79 and 1.14, showing that the system can either condense or slightly expand the text depending on the number of factual corrections inserted. **Normalized Edit Similarity (NES)** remained moderate-to-high (0.18–0.85), indicating controlled rewriting that preserves style and fluency while ensuring factual alignment.

**Action-level precision and recall.** The metrics introduced to qualify the final phases confirm the robustness of the system: the **Fourth Recall** is equal to **1.00** in all scenarios, while the average **Fourth Precision** is **1.00** (*MED*=1.00, *MIC*=-, *BCE*=1.00, *HAM*=1.00, *CIN*=1.00). In the MIC case, the absence of unsupported final sentences in the face of 6 actions results in undefined Precision and Recall because no action is in the fourth stage is performed and no unsupported sentence is reveled, so we decide to omit it from the average.

**Cross-Domain Robustness.** The framework demonstrated stable results across heterogeneous domains — from quantitative economic texts to stylistically rich cultural narratives. Its modular design (Zero–Fourth Checks) and explicit role prompting ensured that:

- Factual contradictions were corrected early (Zero–Second Checks);
- Unsupported or speculative claims were completely removed (Third Check);
- Every retained sentence was traceable to a verifiable source (Fourth Check).

## 3.14 Overall Results — Visual Summary

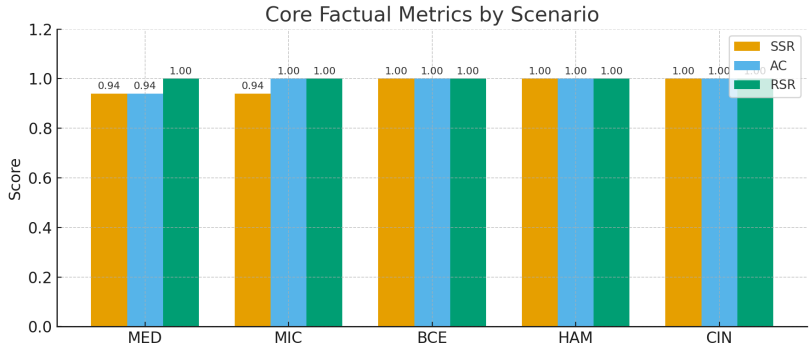


Figure 3.1: Core factual metrics (SSR, AC, RSR) across scenarios MED, MIC, BCE, HAM, CIN. Higher values indicate improved factual reliability.

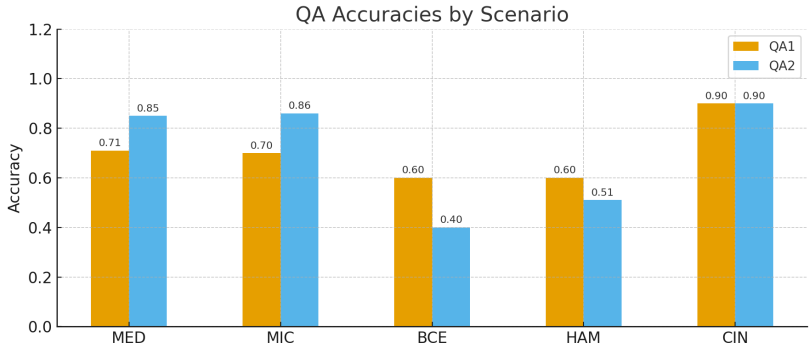


Figure 3.2: QA accuracies (Concept/Sentence) across scenarios.

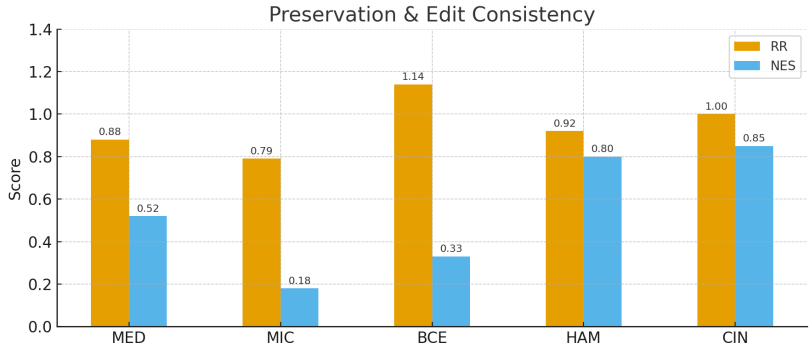


Figure 3.3: Preservation and edit consistency (RR, NES) across scenarios.

## 3.15 Summary of Findings

Across five heterogeneous domains — medical, environmental, economic, sports, and cultural — the proposed multi-step verification framework at the end achieved:

- **100% hallucination removal (RSR = 1.00)**, with no false positives;
- **Average factual support above 0.95** (SSR  0.95 range 0.83–1.00, AC 1.00);
- **Stable linguistic preservation** (RR  0.87, NES  0.49);
- **Mean QA accuracies:** Concept ≈ 0.74, Sentence ≈ 0.69;
- and **robust cross-domain performance** despite stylistic variability.
- **Fourth Precision** = 1.00, e **Fourth Recall** = 1.00, confirming that the final cleanup is exhaustive and generally accurate (with the MIC case, by definition, leading to undefined values for Fourth Precision and Fourth Recall in the absence of unsupported targets).

These outcomes confirm the scalability and reliability of the framework and the capacity to maintain both factual correctness and stylistic fidelity, validating its scalability and reliability across domains of different narrative and factual density. Unlike prior verification-only methods [15], [27], the system combines detection, correction, and traceability into a unified process — transforming hallucination removal into a quantitatively measurable improvement of factual alignment.

**Conclusion of Chapter 3.** The experimental evaluation confirms that the proposed multi-step pipeline achieves full hallucination removal and complete source traceability across all tested domains. By combining detection, correction, and verification, the system transforms factual validation into a quantifiable and reproducible process — a significant step toward accountable LLM-based text generation.

# Conclusion

This thesis addressed the problem of *hallucinations* in Large Language Models (LLMs) by proposing and evaluating a **multi-step verification and correction framework** that progressively operates at concept, sentence, and claim-level granularity [7], [21], culminating in full **source traceability** for every sentence [11] of the generated text.

## Main findings and alignment with objectives

The main objectives were to detect and mitigate hallucinations, actively correct unsupported content, guarantee factual traceability, and preserve linguistic quality. Across five heterogeneous experimental scenarios, the framework demonstrated:

- **RSR = 1.00 (Removal Success Rate):** all intentionally introduced hallucinations were successfully removed or rewritten;
- **High factual support and traceability: SSR** $\approx 0.95$ and **AC** $\approx 1.00$, confirming near-complete factual coverage with explicit quotations;
- **Preservation of structure and coherence: RR** $\approx 0.87$ and **NES** $\approx 0.49$, indicating stylistic consistency despite necessary edits;
- **Improved factual alignment during QA stages: QA1** $\approx 0.74$, **QA2** $\approx 0.69$;
- **Strong performance in final verification: Fourth Recall** $= 1.00$, **Fourth Precision** $= 1.00$, reflecting efficient correction.

Ablation studies confirmed that the **Third Check** (hallucination identification) and **Fourth Check** (source traceability) are the most impactful components, substantially improving factual reliability and interpretability.

These findings confirm that a structured, multi-level verification pipeline can outperform ad-hoc post-processing in maintaining factual reliability while preserving narrative coherence. The results demonstrate that hallucination mitigation is achievable without sacrificing linguistic naturalness.

## Practical relevance and significance

The proposed framework combines both **detection and correction**, advancing beyond approaches limited to flagging errors. Its implications extend to several domains:

- **Journalism and communication:** ensuring publishable articles with sentence-level citations;
- **Medical and legal contexts:** enhancing factual accountability and post-hoc auditability of generated content;
- **Enterprise or RAG pipelines:** serving as a reliable *post-verification layer* decoupled from model generation, reinforcing factual guardrails.

From a scientific standpoint, the thesis contributes:

1. A **multi-granularity architecture** that unifies detection, correction, and traceability;
2. A **reproducible metric set** linking factual support, Attribution Coverage, and content preservation;
3. A fully **replicable on-premise implementation** (via Ollama and open models), supporting transparent and auditable research.

Beyond these domains, the proposed method provides a modular foundation for any LLM-centric system requiring transparent factual grounding, making it applicable to AI-assisted journalism, knowledge extraction, and explainable decision support.

## Limitations and challenges

Several constraints were encountered during experimentation:

- **Cost and latency:** multiple verification steps and model calls increase runtime, partially mitigated through batching and caching;
- **Dependence on source quality:** incomplete or biased references may reduce factual coverage or lead to over-filtering;
- **Cross-sentence and paraphrased facts:** certain distributed claims require more advanced extraction or citation aggregation;
- **Domain scope and dataset scale:** the evaluation was conducted on a limited number of test cases and on medium-length articles (typically 10–20 sentences). This controlled setup ensured interpretability and manual verification, but it also means that scalability to longer documents and larger test suites remains to be validated;
- **Generalization:** current tests focus on journalistic texts, limiting extension to open-domain or multimodal contexts.

These limitations also reveal a broader tension between verification granularity and scalability—balancing interpretability, cost, and completeness remains an open challenge in LLM governance.

## Overall significance and closure

The study demonstrates that hallucination mitigation is not a single-stage process but a **structured, iterative pipeline** that integrates external evidence, multi-level QA, targeted correction, and explicit traceability. In this perspective, the framework embodies a step to a more **safe and responsible use of AI**: every final sentence of the output is verifiable and justified through textual evidence.

By integrating verification, correction, and traceability into a single framework, this Work contributes to the growing field of post-hoc factual alignment and advances the discussion on the interpretability of generative models.

## Future developments

Building on these results, several research directions are envisioned:

- **Efficiency and scalability:** adaptive routing to skip redundant steps, parallelization, and memoization of stable outputs;
- **Automated retrieval:** integrating RAG or knowledge bases to expand or rebalance source coverage dynamically;
- **New Metrics**: include metrics that can track and evaluate the draft test overall the entire process;
- **Enhanced claim-level checking:** improved coreference handling and compositional reasoning; dynamic combination of classic and optimized variants;
- **Structured output and JSON integration:** future iterations of the pipeline could produce structured JSON outputs summarizing verification results, supported sources, and confidence levels for each sentence. This would enable interoperability with downstream systems such as dashboards, audit logs, or visualization tools, enhancing transparency and automation;
- **Internal reliability signals:** integration of verbalized uncertainty, self-consistency, or multi-agent debate as complementary trust indicators;
- **Domain and modality extension:** adaptation to multimodal inputs (images, tables) and interactive verification tools for end-users.
- **Test pool expansion**: utilizing more computational power is it possible to perform more test and to employ more complex and greater text.

Such extensions could leead to semi-autonomous verification agents capable of collaborating with human editors in a continuous fact-checking loop.

**In summary,** this work has proven to redefines hallucination reduction as a **traceable correction process** rather than mere detection, providing both scientific insight and practical tools for ensuring factual reliability in real-world LLM applications.

To conclude we can say that reliability in language models emerges not from scale, but from traceability and trust.

# Appendix A

# Prompt Design and Engineering Choices

## A.1 Introduction

This appendix provides a detailed overview of the prompt design adopted in each stage of the multi-step pipeline for hallucination detection and correction. Prompts represent the interface between the Large Language Model (LLM) and the specific verification task: their formulation strongly influences the reliability of the generated answers and the ability to minimize hallucinations.

The design of the prompts in this work has been guided by techniques and best practices discussed in Boonstra et al. [37], including:

- **Zero-shot prompting**, used when the model is required to solve a task without examples.
- **Few-shot prompting**, is the case where minimal examples are provided to guide the structure of the output.
- **System, role, and contextual prompting**, assigning the LLM a specific role (e.g., "critical reviewer" or "fact-checker"), defining the scope, and providing task-specific context.
- **Chain of Thought (CoT))**, which encourages the model to reason step by step before giving an answer.
- **Step-back prompting**, is used to first reformulate claims as broader questions before verification.
- **Self-consistency**, where multiple reasoning paths are compared to select the most consistent output.
- **ReAct prompting**, combining reasoning with explicit actions, here adapted to source validation and citation checking.

Each stage of the pipeline is thus associated with a dedicated prompt template, explicitly designed and tested to enforce factual grounding and prevent

unsupported statements. The following sections present, step by step, the original prompts used in the implementation (reported verbatim from the Python modules), along with a discussion of their alignment with prompt engineering techniques.

## A.2 Initial Screening (Zero-check)

The initial screening step, implemented in `zero_check.py`, represent the first line of defense against hallucinations. Its purpose is to ensure that unsupported or contradictory statements are removed or corrected before proceeding to deeper QA checks. The process operates in two different modes depending on whether a single source document or multiple sources are provided.

### Single-source validation

When only one source is available, the model is explicitly instructed to act as a *professional fact-checker and editorial corrector*. This prompt makes use of **system prompting**, by assigning the model a fixed role, and of **few-shot prompting**, by including explicit correction examples. The complete prompt is the following:

```
## Your Role
You are a professional fact-checker and editorial corrector.
Your task is to review a generated article against a trusted
source document. You must revise or remove any hallucinated,
inaccurate, or misleading content not supported by the source.
Maintain a clear, natural tone and write in fluent **Italian**.

## Instructions

### Step 1: Analyze the Source
- Treat the source document as the absolute truth
- Identify: named entities, dates, numbers, events, outcomes,
  causal relationships

### Step 2: Review the Article
- Compare article claims to the source
- Detect:
  - Contradictions
  - Invented information (hallucinations)
  - Misinterpretations

### Step 3: Correct the Article
- Remove or rewrite problematic sentences
- Keep logical flow, coherence, and tone
- Do **not** add new content

### Output
Return only the fully revised article text in **Italian**,
```

```
without comments or explanations.

## Few-shot Examples

### Example 1
**Source**: "Oetem è stata fondata nel 1983 a Torino."
**Article**: "Oetem è nata nel 1985 a Milano."
**Correction**: "Oetem è stata fondata nel 1983 a Torino."

### Example 2
**Source**: "Nel 2022, l'azienda ha registrato un utile netto
di 12 milioni di euro."
**Article**: "Nel 2022 ha avuto gravi perdite."
**Correction**: "Nel 2022, l'azienda ha registrato un utile
netto di 12 milioni di euro."

---

## Task
Now apply the same process to the article below.
Use only the following source as reference.
Write the corrected article in **Italian** and return only
the final text.

### Source Document:
<source content>

### Generated Article:
<draft article>
```

## Multi-source validation

When multiple sources are provided, the Zero-check adopts a sentence-level validation procedure, the sentences are divided using a tokenizer. Each sentence of the draft article is individually checked against all sources using a strict binary classification: supported (`Sì`) or not supported (`No`). This corresponds to a **zero-shot prompting** strategy, where no examples are included and the model must directly evaluate factual support. The complete prompt is the following:

```
## Your Role
You are a validator. Check if the following sentence is supported
by the document below.

## Instructions
- If the sentence is clearly supported, reply: "Sì"
- If not, reply: "No"
- Answer in Italian, just one word: Sì / No

## Sentence:
"<sentence>"
```

```
## Source:
<document>
```

If a sentence is not supported by any source, an auxiliary prompt is invoked to remove it from the article while preserving coherence:

```
## Your Role
You are an editorial reviser. Remove the sentence below
from the article.

## Sentence to remove:
"<sentence>"

## Article:
<full article>

## Output:
Return the article with the sentence removed.
```

### Prompt engineering perspective

From a prompt engineering perspective, the Zero-check exemplifies:

- **System prompting**: the model is explicitly framed as a fact-checker, validator, or reviser, reducing ambiguity in its behavior.
- **Few-shot prompting**: in the single-source case, examples of incorrect vs. corrected claims guide the model toward the expected rewriting style.
- **Zero-shot prompting**: in the multi-source case, sentence-level validation relies on direct binary answers, maximizing clarity and minimizing computational cost.

This design ensures that hallucinated or unsupported claims are filtered out at the very first stage, before engaging in deeper question–answering validation and iterative correction.

## A.3 Concept-Level QA Validation (First Check)

The second stage of the pipeline, implemented in `first_check.py`, focuses on concept-level validation. Instead of working at the level of single sentences, this step generates **question–answer pairs directly from the sources**. Each pair is then used to verify whether the draft article provides the same information through the use of the primary LLM model. This approach enables the identification of factual gaps, contradictions, or missing content.

## Prompt for Q&A generation

The first part of the process instructs the model to read a source and generate at least five verifiable question–answer pairs. This uses **role prompting** (assigning the role of "critical reviewer and domain expert") and can be seen as a form of **few-shot prompting**, since it enforces a specific Q&A output format.

```
## Your Role
You are a critical reviewer and domain expert. Your task is to
evaluate whether a generated article faithfully answers questions
based on trusted source documents. Your evaluation must be based
only on verified facts from the source.

## Instructions

### Step 1: Generate Questions
- Read the source below.
- Generate **at least 5 specific and verifiable** question-answer
  pairs based only on that source.

Each pair must follow this exact format:
Domanda: ...
Risposta: ...

Use **Italian language**. Do not use titles or numbering.
Each pair must be separated by a line break.

### Step 2: Compare Against the Article
You will later be asked to compare the article's answer to the
source answer for each question. You must evaluate it as:

 CORRETTA - fully consistent with the source
 ERRATA - incorrect or conflicting
 ASSENTE - the article does not contain an answer

Keep your judgment strict. Assume the article cannot "improvise"
information.

---

## Output format (Q&A generation)
Now generate 5 Italian Q&A pairs based **only** on the following
source document. Follow the formatting instructions strictly.

### Source Document:
<document content>
```

## Prompt for comparative evaluation

For each generated question, the model is then asked to compare the source answer with the article's answer. This stage uses **Chain of Thought prompting**, as the model must reason step by step and explain its judgment, but still return a concise classification on the first line.

```
## Your Role
You are an expert evaluator. Your task is to compare two answers
to the same question: one from a trusted source and one from an
article.

## Instructions

- Assume the source answer is correct.
- If the article's answer **conveys the same meaning**, even through
  a **faithful citation**, consider it CORRETTA.
- Do not penalize if the article uses **direct quotes** instead of
  paraphrasing, as long as the meaning is preserved and clear.
- Penalize only if the article introduces **extra information**,
  **invents content**, or **fails to answer**.

### Step 2: Classify the Article's Answer
Label it as:
 CORRETTA - the article's answer is factually accurate and matches
the source
 ERRATA - the article includes incorrect or conflicting information
 ASSENTE - the article does not provide any real answer to the question

### Step 3: Output Format
Return the label on the **first line only**:  /  /
Then briefly explain your judgment in **Italian**.

---

## Input
Domanda: <generated question>
Risposta (Fonte): <source answer>
Risposta (Articolo): <article answer>
```

## Prompt for iterative correction

Finally, for each identified error or missing answer, the model is instructed to revise the article by integrating or correcting the information. This is an example of **instruction-based prompting** and **editorial role prompting**, where the LLM is framed as an assistant that carefully integrates factual content without altering unrelated sections.

```
## Your Role
You are an editorial assistant and factual corrector. Your task is
to revise an article by inserting or correcting a specific fact
```

```
derived from a trusted source.

## Instructions

### Step 1: Read the article
- Understand its structure and tone
- Do not alter unrelated parts

### Step 2: Integrate the Information
- Insert or correct the information required to answer the question
  below
- Keep the article coherent and natural
- Do not add extra explanations or new content
- Write the final article entirely in **Italian**

### Input

Domanda: <question>
Risposta corretta (fonte): <correct source answer>

### Article to revise:
<current article>

### Output
Return only the **updated article** with the integrated or corrected
information, written in Italian.
```

### Prompt engineering perspective

From a prompt engineering perspective, this stage combines:

- **Few-shot prompting**: enforcing the production of multiple Q&A pairs with a strict output format.
- **Role prompting**: explicitly casting the model as a "critical reviewer" and later as an "editorial assistant".
- **Chain of Thought prompting**: encouraging structured reasoning in the evaluation phase, where explanations justify each judgment.

This design allows the system to systematically detect contradictions or omissions in the article, and to correct them by injecting factual content directly sourced from the references.

## A.4 Fine-Grained QA Analysis (Second Check)

The third stage of the pipeline, implemented in `qa_module.py`, performs a more fine-grained analysis by generating question–answer pairs at the **sentence level**, and the sentences are divided using a tokenizer. Each sentence from the sources is converted into a factual question, and the draft article is queried with the same question using the primary LLM model. The answers are then compared to detect inconsistencies, errors, or omissions.

### Prompt for question generation

For each sentence, the model is instructed to generate a single specific, verifiable question. This is an instance of **system prompting** (role: "question generation expert") combined with **zero-shot prompting**, since no examples are provided.

```
## Your Role
You are a question generation expert. Your task is to write a
specific, verifiable question that matches the meaning of a
single sentence.

## Instructions
- Write one question that tests the factual content of the sentence
- The question must be answerable using only the sentence
- Use **Italian** for the question
- Keep it short and precise

## Sentence:
<sentence from source>
```

### Prompt for answering from the source

Immediately afterwards, the model is instructed to provide the factual answer using only the source sentence. This reinforces the grounding process by ensuring that the question can indeed be answered.

```
## Your Role
You are a factual assistant. Your task is to answer the question
using only the sentence provided.

## Instructions
- Do not assume or infer beyond what is stated
- Answer in **Italian**
- Be concise

## Sentence:
<sentence from source>

## Question:
<generated question>
```

### Prompt for answering from the article

The same question is then asked against the draft article, to check whether the article contains the information.

```
## Your Role
You are an assistant extracting answers from a draft article.

## Instructions
- Answer the question using only the content of the article
```

```
- If the answer is not present, say clearly that it is missing
- Do not assume or invent
- Answer in **Italian**

## Question:
<generated question>

## Article:
<draft article>
```

### Prompt for comparative evaluation

The two answers (source vs. article) are compared by an evaluator prompt. This step explicitly uses **Chain of Thought prompting**, requiring the model to classify the article's answer and explain its reasoning.

```
## Your Role
You are an expert evaluator. Your task is to compare two answers
to the same question: one from a trusted source and one from an
article.

## Instructions

### Step 1: Compare Carefully
- Assume the source answer is correct.
- If the article's answer **conveys the same meaning**, even through
  a **faithful citation**, consider it CORRETTA.
- Do not penalize if the article uses **direct quotes** instead of
  paraphrasing, as long as the meaning is preserved and clear.
- Penalize only if the article introduces **extra information**,
  **invents content**, or **fails to answer**.

### Step 2: Classify the Article's Answer
Label it as:
 CORRETTA - the article's answer is factually accurate and matches
the source
 ERRATA - the article includes incorrect or conflicting information
 ASSENTE - the article does not provide any real answer to the question

### Step 3: Output Format
Return the label on the **first line only**:  /  /
Then briefly explain your judgment in **Italian**.

---

## Task
Question: <generated question>
Source Answer: <answer from source>
Article Answer: <answer from draft article>
```

### Prompt for iterative correction

Finally, for each factual inconsistency or omission, the article is revised. The model is asked to act as an editorial assistant, inserting or correcting the missing information while keeping the article fluent. This is a case of **instruction-based prompting** and role-based correction.

```
## Your Role
You are an editorial assistant and factual corrector. Your task is
to revise an article by inserting or correcting a specific fact
derived from a trusted source.

## Instructions

### Step 1: Read the article
- Understand its structure and tone
- Do not alter unrelated parts

### Step 2: Integrate the Information
- Insert or correct the information required to answer the question
  below
- Keep the article coherent and natural
- Do not add extra explanations or new content
- Write the final article entirely in **Italian**

## Example

Domanda: Dove si trova la sede centrale dell'azienda?
Risposta corretta: A Torino.

Articolo prima:
"L'azienda ha sede a Milano ed è specializzata nella produzione
di valvole."

Articolo corretto:
"L'azienda ha sede a Torino ed è specializzata nella produzione
di valvole."

### Input

Domanda: <question>
Risposta corretta (fonte): <source answer>

### Article to revise:
<current article>

### Output
Return only the **updated article** with the integrated or corrected
information, written in Italian.
```

### Prompt engineering perspective

From a prompt engineering perspective, this stage integrates:

- **Zero-shot prompting**: for generating factual questions without examples.
- **System prompting**: assigning roles such as "question generation expert", "factual assistant", and "expert evaluator".
- **Chain of Thought prompting**: is used in the evaluation phase, where reasoning and justification are required.
- **Instruction prompting**: used in the correction stage, ensuring coherent integration of facts.

This fine-grained QA loop enables detailed alignment between the draft and the sources, capturing subtle errors and omissions that the concept-level check may miss.

## A.5 Hallucination Identification (Third Check)

The fourth stage of the pipeline, implemented in `hallucination_checker.py` and `hallucination_check_alt.py`, is dedicated to the detection of hallucinated statements. The logic of this stage is that if a factual claim cannot be supported by *any* source, it is considered invented and must be removed or rewritten. Two variants of the process were implemented: a sentence-based (classic) approach and a claim-based (optimized) approach.

### Classic method (sentence-based)

In the classic version, each sentence of the draft article is divided using a tokenizer and then converted into a single, precise question. The model is framed as a "question-generation expert" and later as an "evidence-checking assistant". This approach resembles **step-back prompting**, because the model must reformulate statements as verifiable questions before validation.

```
## Your Role
You are a question-generation expert. Your task is to create
one precise and fact-based question derived from a single sentence.

## Instructions
- Use only the sentence to form your question
- Make it specific and verifiable
- Do not assume context
- Write the question in **Italian**

## Sentence:
<sentence from article>
```

Each generated question is then checked against every source:

```
## Your Role
You are an evidence-checking assistant. Your task is to answer
a question **using only** the content of the document provided.

## Instructions
- If the document provides a clear answer, write it in **Italian**
- If there is **no answer**, reply with exactly: "NON PRESENTE"
- Do not assume or infer anything not explicitly stated

## Question:
<generated question>

## Document:
<source content>
```

If no source provides an answer, the statement is flagged as unsupported, and the model is asked to remove or rewrite it:

```
## Your Role
You are a content auditor. Your task is to remove or rewrite
invented statements from an article based on unsupported questions.

## Instructions
- The questions listed below could not be answered by any trusted source
- Identify and revise/remove the corresponding parts of the article
- Maintain logical flow and natural language
- Write the revised article in **Italian**

## Example

Domanda: Quando è stata fondata l'azienda?
Nessuna fonte ha fornito risposta → frase considerata inventata.

Articolo prima:
"Oetem è stata fondata nel 1995 a Brescia."

Articolo corretto:
(Frase rimossa)

## Unsupported Questions:
- <list of unsupported questions>

## Article to revise:
<draft article>

## Output
Return only the final revised article, written in Italian.
```

### Optimized method (claim-based)

The optimized version first extracts factual claims directly from the article, then reformulates them into questions. This approach is closer to **automatic prompt engineering**, as the model is effectively asked to generate prompts (questions) from the input text.

```
## Your Role
You are a question extraction expert. Your task is to read
an article and extract a list of specific, verifiable questions -
one per factual claim - based solely on the article content.

## Instructions
- Write only questions that reflect explicit factual statements
  in the article
- Focus on who/what/when/where/why/how
- Ignore vague, opinion-based or stylistic content
- Write all questions in **Italian**
- Use this format:
Domanda: ...

## Article:
<full article>
```

Each extracted question is then validated against the sources with a dedicated prompt:

```
## Your Role
You are a source validator. Your task is to answer the question
below using only the content of this document.

## Instructions
- If the answer is found in the document, write it in **Italian**
- If there is no explicit answer, reply exactly with: "NON PRESENTE"
- Do not invent, infer or assume

## Question:
<generated question>

## Document:
<source content>
```

Unsupported claims trigger a removal/revision step:

```
## Your Role
You are an editorial cleaner. Your task is to remove or rewrite
hallucinated statements from the article below.

## Instructions
- The following questions could not be answered by any source
- Remove or rephrase the corresponding factual claims
```

```
- Ensure the final article is fluent, coherent and **written in Italian**

## Example

Unsupported Question: Qual è il numero di dipendenti dell'azienda?

Articolo prima:
"L'azienda conta circa 800 dipendenti."

Articolo corretto:
(Frase rimossa o riformulata per evitare il numero non verificato)

## Unsupported Questions:
- <list of unsupported questions>

## Original Article:
<draft article>

## Output
Return only the final revised article.
```

### Prompt engineering perspective

From a prompt engineering perspective, the hallucination check illustrates:

- **Step-back prompting**: in the classic version, each sentence is reframed as a factual question before validation.
- **Automatic prompt engineering**: in the optimized version, the model itself extracts factual claims and generates verification questions.
- **Role prompting**: consistent role assignment (question generator, validator, auditor) structures the interaction.

This stage ensures that hallucinated content, i.e. information unsupported by any source, is systematically removed or rewritten to maintain factual accuracy.

## A.6 Source Traceability and Metric Integration (Fourth Check)

The final stage of the pipeline, implemented in `quarto_check.py`, is designed to validate the traceability of every sentence in the article. Unlike previous checks that focus on correction and hallucination removal, this stage aims to establish a clear mapping between each article sentence and the supporting source(s). Sentences without any reliable support are removed.

This phase operates through a dual-model setup: the **primary model** (`gemma2:9b`) performs citation extraction and reasoning, while a **secondary validation model** (`LLaMA3.1:8b`) confirms factual correspondence between the article and the extracted quotes. Each validated sentence contributes to

quantitative metrics such as the **Sentence Support Rate (SSR)**, **Attribution Coverage (AC)**, and **Strict Support Rate ($\mathbf{SSR_{strict}}$)**, ensuring that factual grounding is not only guaranteed but also measurable.

## Verification prompt

Each sentence of the draft is validated against all sources. The model is framed as a **verification engine** and is required to return structured judgments, indicating whether the sentence is supported and providing direct citations when possible. This reflects a form of **ReAct prompting**, where the model performs reasoning (evaluation) followed by an explicit action (citing evidence).

```
## Your Role
You are a verification engine. Your task is to determine whether
a given sentence from an article is supported by any of the
documents below.

## Instructions
- For each source, answer:
  - Whether the sentence is supported by the document
  - Support can come from one or more parts of the text
  - If yes, extract one or more literal quotes (each 50 words) that justify it
  - Quotes can come from different parts of the document if they jointly
  support the sentence
- Be strict but fair. If the **meaning** is clearly present, even if
  split across the document, say "Presente: Sì"

- Respond in the following format per source:

Fonte: <filename>
Presente: Sì/No
Citazione: <quote or "N/D">

## Sentence:
"<sentence>"

## Sources:
<list of source documents>
```

## Removal prompt for unsupported sentences

If no source provides valid support, the model is instructed to remove the sentence entirely:

```
## Your Role
You are an editorial reviser performing a factual cleanup.
Your task is to remove a sentence from the article because it
is not supported by any trusted source.
```

```
## Instructions
- Remove the sentence listed below
- Keep the article coherent and fluent
- Write in **Italian**
- Do not touch unrelated parts

## Sentence to remove:
"<sentence>"

## Article:
<full article>

## Output:
Return the article with the sentence removed.
```

### Secondary validation prompt

If a citation is found, the system invokes a secondary model to double-check whether the citation truly supports the claim. This validation is performed by a **secondary, lightweight model** (e.g., `LLaMA3.1:8b`), which acts as an independent reviewer confirming or rejecting the factual match identified by the primary model. This is a form of **cross-model validation** and also relates to **self-consistency**, as the judgment is confirmed by an independent reasoning step.

```
## Your Role
You are a fact-checker. Your task is to validate whether a
sentence from an article is truly supported by a quote or a
combination of quotes found in a source.

## Instructions
- Compare the sentence with the quote(s)
- If the quote(s) support **the core meaning** or a
  **significant part** of the sentence, reply "Sì"
- If the quote(s) only support trivial or marginal aspects,
  reply "No"
- Always explain your reasoning in **Italian**

## Example

Sentence: "L'azienda ha aumentato i ricavi grazie all'espansione
in Europa e alla crescita del mercato nazionale."
Quote: "Il mercato nazionale ha registrato una forte crescita
nell'ultimo anno."

Response: Sì. La citazione conferma una parte chiave del contenuto
(la crescita del mercato nazionale), anche se non menziona
esplicitamente l'espansione in Europa.

---
```

```
Sentence: "<sentence>"

Quote(s):
<citations from sources>

## Output
Write "Sì" or "No" followed by a brief justification (in Italian).
```

### Prompt engineering perspective

From a prompt engineering perspective, the Fourth Check is characterized by:

- **Role prompting**: explicit roles such as "verification engine", "editorial reviser", and "fact-checker".
- **ReAct prompting**: combining both reasoning with actions (providing structured outputs and citations).
- **Self-consistency and cross-validation**: leveraging a secondary model to confirm whether citations genuinely support the claim.

This step guarantees that every retained sentence is grounded in at least one source, and that the link between article and evidence is explicitly documented.

### Integration with Evaluation Metrics

The structured outputs of the Fourth Check directly feed the computation of the framework's quantitative indicators. Each validated sentence–source pair contributes to the **Sentence Support Rate (SSR)**, while the presence of at least one extracted citation contributes to **Attribution Coverage (AC)** and **Strict Support Rate ($\mathbf{SSR_{strict}}$)**. This integration ensures that prompt-based verification is systematically transformed into measurable, reproducible indicators of factual reliability.

## A.7 Concluding Remarks on Prompt Design

The prompts documented in this appendix represent the operational backbone of the proposed multi-step framework for hallucination detection and correction. While each stage has a different focus — from initial screening to fine-grained QA validation, hallucination removal, and final traceability — all prompts were designed with a common set of principles in mind:

- **Clarity and determinism**. Instructions are written in unambiguous language, often enforcing structured outputs (binary "Sì/No" answers, labeled classifications, or strict Q&A pairs) to reduce the model's freedom to improvise.
- **Role prompting**. Each prompt assigns the LLM a specific role (*fact-checker*, *critical reviewer*, *editorial assistant*, *verification engine*), specifying its behavior and improving consistency.

- **Few-shot and zero-shot prompting**. The pipeline alternates between examples (few-shot) to anchor expected outputs (as in the single-source Zero-check) and strict zero-shot classification (as in the binary validation of the multi-source Zero-check).
- **Chain of Thought (CoT)**. In evaluation prompts, the model is encouraged to reason and explain its judgment before providing a final label, improving interpretability and reliability.
- **Step-back prompting**. In the hallucination check, draft sentences are reframed as factual questions before validation, forcing a shift from narrative to verifiable claim.
- **Automatic prompt engineering**. In the optimized hallucination check, the model itself generates verification questions directly from the article, creating secondary prompts as part of the process.
- **ReAct prompting and self-consistency**. In the final traceability check, reasoning is combined with explicit actions (providing citations), and an additional model validates the evidence, ensuring cross-checking and consistency.
- **Metric-driven validation**. The outputs of the prompts are structured to directly support the computation of quantitative metrics (SSR, AC, RSR, NES), bridging qualitative reasoning with measurable evaluation.

In summary, the prompt design of this framework operationalizes several best practices from recent literature on Prompt Engineering [37] adapting them to a local multi-model environment. The goal was not only to correct hallucinations but also to guarantee transparency and reproducibility: each decision of the pipeline is grounded in explicit instructions, structured outputs, and traceable evidence.

# Appendix B

# Evaluation Metrics

## B.1 Traceability / Factual Reliability

| Name | Formula | Description |
|---|---|---|
| **Sentence Support Rate (SSR)** | $\frac{\#\text{supported sentences}}{\#\text{total sentences}}$ | Measures the percentage of sentences supported by at least one source. Indicates factual coverage of the final article. |
| **Attribution Coverage (AC)** | $\frac{\#\text{sentences with literal citations}}{\#\text{supported sentences}}$ | Share of sentences traceable to at least one literal citation. Captures transparency and traceability. |
| **Strict Support Rate ($\mathbf{SSR_{strict}}$)** | $\frac{\#\text{sentences with citations}}{\#\text{total sentences}}$ | A stricter version of SSR that penalizes unsupported but retained sentences. |

Table B.1: Traceability and factual reliability metrics.

**Interpretation:** High SSR, AC, and $\text{SSR}_{\text{strict}}$ indicate strong factual grounding.
*Ideal ranges:* SSR $> 0.85$, AC $> 0.70$, $\text{SSR}_{\text{strict}} > 0.60$.

## B.2 QA Accuracy / Correction Effectiveness

| Name | Formula | Description |
|---|---|---|
| **QA_Accuracy_ Concept-level (QA1)** | $\frac{\#\text{correct answers}}{\#\text{questions (First)}}$ | Concept-level accuracy: proportion of correct answers to source-derived concept questions. |
| **QA_Accuracy_Sentence-level (QA2)** | $\frac{\#\text{correct answers}}{\#\text{questions (Second)}}$ | Sentence-level accuracy at finer granularity. |
| **Fixes_Applied_Count** | #non-correct answers | Number of corrections triggered by wrong or missing answers. |
| **Information Gain (IG)** | Count of manually added facts | Manually verified facts integrated into the article during correction. |

Table B.2: QA accuracy and correction metrics.

**Interpretation:** High QA accuracies indicate reliable factual alignment. High Fixes counts reflect how many contradictions were corrected, while Information Gain measures the amount of additional factual coverage.

## B.3 Hallucination density / removal metrics

| Name | Formula | Description |
|---|---|---|
| **Unsupported Claim Ratio (UCRR)** | $\frac{\#\text{unsupported claims}}{\#\text{total claims}}$ | Proportion of generated questions that could not be answered by any source. Indicates the hallucination density of the draft. |
| **Removal Success Rate (RSR)** | $\frac{\#\text{unsupported claims removed or rewritten}}{\#\text{unsupported claims total}}$ | Measures the framework's ability to effectively remove or correct hallucinated content. |

Table B.3: Hallucination detection and correction metrics.

**Interpretation:** High UCRR $\rightarrow$ the initial article contained more hallucinations. High RSR $\rightarrow$ the framework successfully removed them. *Ideal:* UCRR $\downarrow$, RSR $\uparrow$ (RSR $> 0.80$).

## B.4 Fourth Check Precision and Recall

| Name | Formula | Description |
|---|---|---|
| **Fourth Precision (Prec4)** | $\frac{\#\text{non-supported sentences correctly removed}}{\#\text{all removal or rewrite actions (Fourth Check)}}$ | Action-level precision: proportion of final cleanup actions that correctly targeted unsupported sentences. |
| **Fourth Recall (Rec4)** | $\frac{\#\text{non-supported sentences correctly removed}}{\#\text{all non-supported sentences}}$ | Completeness of the final factual cleanup: how many unsupported sentences were actually removed. |

Table B.4: Final traceability precision and recall metrics (Fourth Check).

**Interpretation:** High Fourth Precision means the final verification stage acted only on truly unsupported content (no over-deletion). High Fourth Recall means that all unsupported sentences were effectively removed. When both are high, the cleanup is both accurate and exhaustive.

## B.5 Preservation / Edit Consistency

| Name | Formula | Description |
|---|---|---|
| **Retention Rate (RR)** | $\frac{\#\text{final sentences}}{\#\text{initial sentences}}$ | Indicates how much of the initial content remains in the final article. Balances preservation vs. correction. |
| **Normalized Edit Similarity (NES)** | `SequenceMatcher(initial, final)` | Global normalized textual similarity (0–1) between initial and final articles. Proxy for linguistic stability. |

Table B.5: Preservation and edit consistency metrics.

**Interpretation:** Low RR $\rightarrow$ excessive filtering (over-correction). High RR with low SSR $\rightarrow$ insufficient revision. *Guidelines:* RR $\in [0.5, 0.9]$. NES is *context-dependent*: values in the 0.3–0.8 range are common when factual rewrites are needed; interpret NES jointly with RR and factual metrics (SSR/AC).

## B.6 Summary Table

| **Category** | **Metric** | **Trend** | **Ideal Range** | **Meaning** |
|---|---|---|---|---|
| Traceability | SSR | ↑ | >0.85 | Factual coverage |
| Traceability | AC | ↑ | >0.70 | Citation traceability |
| QA Accuracy | QA1 | ↑ | >0.80 | Concept-level accuracy |
| QA Accuracy | QA2 | ↑ | >0.80 | Sentence-level accuracy |
| Hallucination | UCRR | ↓ | <0.20 | Invented claim ratio |
| Hallucination | RSR | ↑ | >0.80 | Removal Success Rate |
| Traceability | Prec4 | ↑ | ≈ 1.0 | Accuracy of final factual cleanup |
| Traceability | Rec4 | ↑ | ≈ 1.0 | Completeness of final factual cleanup |
| Preservation | RR | medium | 0.5–0.9 | Balance between correction and retention |
| Preservation | NES | ↑ (context) | — (dataset-dependent) | Textual stability (read with RR) |

Table B.6: Summary of all quantitative metrics integrated in the framework.

# Bibliography


[1] E. Sulem, O. Abend, and A. Rappoport, *Bleu is not suitable for the evaluation of text simplification*, 2018. arXiv: 1810.05995 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1810.05995.

[2] T. B. Brown et al., *Language models are few-shot learners*, 2020. arXiv: 2005.14165 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2005.14165.

[3] Y. Dong, S. Wang, Z. Gan, Y. Cheng, J. C. K. Cheung, and J. Liu, *Multi-fact correction in abstractive text summarization*, 2020. arXiv: 2010.02443 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2010.02443.

[4] L. Gao et al., *The pile: An 800gb dataset of diverse text for language modeling*, 2020. arXiv: 2101.00027 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2101.00027.

[5] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, *The curious case of neural text degeneration*, 2020. arXiv: 1904.09751 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1904.09751.

[6] J. Kaplan et al., *Scaling laws for neural language models*, 2020. arXiv: 2001.08361 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2001.08361.

[7] A. Wang, K. Cho, and M. Lewis, *Asking and answering questions to evaluate the factual consistency of summaries*, 2020. arXiv: 2004.04228 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2004.04228.

[8] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell, "On the dangers of stochastic parrots: Can language models be too big?" In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, ser. FAccT '21, Virtual Event, Canada: Association for Computing Machinery, 2021, pp. 610–623, ISBN: 9781450383097. DOI: 10.1145/3442188.3445922. [Online]. Available: https://doi.org/10.1145/3442188.3445922.

[9] T. He, J. Zhang, Z. Zhou, and J. Glass, *Exposure bias versus self-recovery: Are distortions really incremental for autoregressive text generation?* 2021. arXiv: 1905.10617 [cs.LG]. [Online]. Available: https://arxiv.org/abs/1905.10617.

[10] O. Honovich, L. Choshen, R. Aharoni, E. Neeman, I. Szpektor, and O. Abend, *$Q^2$: Evaluating factual consistency in knowledge-grounded dialogues via question generation and question answering*, 2021. arXiv: 2104.08202 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2104.08202.

[11] P. Lewis et al., *Retrieval-augmented generation for knowledge-intensive nlp tasks*, 2021. arXiv: 2005.11401 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2005.11401.

[12] R. Bommasani et al., *On the opportunities and risks of foundation models*, 2022. arXiv: 2108.07258 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2108.07258.

[13] H.-S. Chang and A. McCallum, "Softmax bottleneck makes language models unable to represent multi-mode word distributions," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, S. Muresan, P. Nakov, and A. Villavicencio, Eds., Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 8048–8073. DOI: 10.18653/v1/2022.acl-long.554. [Online]. Available: https://aclanthology.org/2022.acl-long.554/.

[14] A. Chowdhery et al., *Palm: Scaling language modeling with pathways*, 2022. arXiv: 2204.02311 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2204.02311.

[15] A. Creswell, M. Shanahan, and I. Higgins, *Selection-inference: Exploiting large language models for interpretable logical reasoning*, 2022. arXiv: 2205.09712 [cs.AI]. [Online]. Available: https://arxiv.org/abs/2205.09712.

[16] J. Hoffmann et al., *Training compute-optimal large language models*, 2022. arXiv: 2203.15556 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2203.15556.

[17] O. Honovich et al., *True: Re-evaluating factual consistency evaluation*, 2022. arXiv: 2204.04991 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2204.04991.

[18] S. Kadavath et al., *Language models (mostly) know what they know*, 2022. arXiv: 2207.05221 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2207.05221.

[19] S. Lin, J. Hilton, and O. Evans, *Truthfulqa: Measuring how models mimic human falsehoods*, 2022. arXiv: 2109.07958 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2109.07958.

[20] L. Ouyang et al., *Training language models to follow instructions with human feedback*, 2022. arXiv: 2203.02155 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2203.02155.

[21] S. Chen, S. Gao, and J. He, *Evaluating factual consistency of summaries with large language models*, 2023. arXiv: 2305.14069 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2305.14069.

[22] S. Chen et al., *Felm: Benchmarking factuality evaluation of large language models*, 2023. arXiv: 2310.00741 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2310.00741.

[23] Y. Du, S. Li, A. Torralba, J. B. Tenenbaum, and I. Mordatch, *Improving factuality and reasoning in language models through multiagent debate*, 2023. arXiv: 2305.14325 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2305.14325.

[24] Z. Ji et al., "Survey of hallucination in natural language generation," *ACM Computing Surveys*, vol. 55, no. 12, pp. 1–38, Mar. 2023, ISSN: 1557-7341. DOI: 10.1145/3571730. [Online]. Available: http://dx.doi.org/10.1145/3571730.

[25] J. Li, X. Cheng, W. X. Zhao, J.-Y. Nie, and J.-R. Wen, *Halueval: A large-scale hallucination evaluation benchmark for large language models*, 2023. arXiv: 2305.11747 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2305.11747.

[26] A. Mallen, A. Asai, V. Zhong, R. Das, D. Khashabi, and H. Hajishirzi, *When not to trust language models: Investigating effectiveness of parametric and non-parametric memories*, 2023. arXiv: 2212.10511 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2212.10511.

[27] P. Manakul, A. Liusie, and M. J. F. Gales, *Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models*, 2023. arXiv: 2303.08896 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2303.08896.

[28] K. Meng, D. Bau, A. Andonian, and Y. Belinkov, *Locating and editing factual associations in gpt*, 2023. arXiv: 2202.05262 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2202.05262.

[29] S. Min et al., *Factscore: Fine-grained atomic evaluation of factual precision in long form text generation*, 2023. arXiv: 2305.14251 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2305.14251.

[30] A. Pal, L. K. Umapathi, and M. Sankarasubbu, *Med-halt: Medical domain hallucination test for large language models*, 2023. arXiv: 2307.15343 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2307.15343.

[31] H. Rashkin et al., "Measuring attribution in natural language generation models," *Computational Linguistics*, vol. 49, no. 4, pp. 777–840, Dec. 2023, ISSN: 0891-2017. DOI: 10.1162/coli_a_00486. eprint: https://direct.mit.edu/coli/article-pdf/49/4/777/2269661/coli_a_00486.pdf. [Online]. Available: https://doi.org/10.1162/coli_a_00486.

[32] H. Touvron et al., *Llama: Open and efficient foundation language models*, 2023. arXiv: 2302.13971 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2302.13971.

[33] A. Vaswani et al., *Attention is all you need*, 2023. arXiv: 1706.03762 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1706.03762.

[34] J. Wei et al., *Chain-of-thought prompting elicits reasoning in large language models*, 2023. arXiv: 2201.11903 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2201.11903.

[35] L. Berglund et al., *The reversal curse: Llms trained on "a is b" fail to learn "b is a"*, 2024. arXiv: 2309.12288 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2309.12288.

[36] H. Bhasin, T. Ossowski, Y. Zhong, and J. Hu, *How does multi-task training affect transformer in-context capabilities? investigations with function classes*, 2024. arXiv: 2404.03558 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2404.03558.

[37] L. Boonstra, *Prompt engineering*, 2024.

[38] T. Groot and M. Valdenegro-Toro, *Overconfidence is key: Verbalized uncertainty evaluation in large language and vision-language models*, 2024. arXiv: 2405.02917 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2405.02917.

[39] M. Hu, B. He, Y. Wang, L. Li, C. Ma, and I. King, *Mitigating large language model hallucination with faithful finetuning*, 2024. arXiv: 2406.11267 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2406.11267.

[40] J. Kasai et al., *Realtime qa: What's the answer right now?* 2024. arXiv: 2207.13332 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2207.13332.

[41] R. Kumar et al., *Automatic question-answer generation for long-tail knowledge*, 2024. arXiv: 2403.01382 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2403.01382.

[42] OpenAI et al., *Gpt-4 technical report*, 2024. arXiv: 2303.08774 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2303.08774.

[43] L. Huang et al., "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions," *ACM Transactions on Information Systems*, vol. 43, no. 2, pp. 1–55, Jan. 2025, ISSN: 1558-2868. DOI: 10.1145/3703155. [Online]. Available: http://dx.doi.org/10.1145/3703155.

[44] Ollama, *Ollama documentation*, https://docs.ollama.com/, Accessed: 2025-10-01, 2025.

[45] L. Ranaldi and G. Pucci, *When large language models contradict humans? large language models' sycophantic behaviour*, 2025. arXiv: 2311.09410 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2311.09410.

[46] G. Team et al., *Gemini: A family of highly capable multimodal models*, 2025. arXiv: 2312.11805 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2312.11805.

[47] Z. Wang, Z. Shi, H. Zhou, S. Gao, Q. Sun, and J. Li, *Towards objective fine-tuning: How llms' prior knowledge causes potential poor calibration?* 2025. arXiv: 2505.20903 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2505.20903.

[48] J. Wu et al., *Mitigating hallucinations in large vision-language models via entity-centric multimodal preference optimization*, 2025. arXiv: 2506.04039 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2506.04039.

[49] J. Yang, D. Chen, Y. Sun, R. Li, Z. Feng, and W. Peng, *Enhancing semantic consistency of large language models through model editing: An interpretability-oriented approach*, 2025. arXiv: 2501.11041 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2501.11041.

# Acknowledgments

I would like to dedicate this work to all the people who I am deeply grateful and who have helped and supported me on this journey.

First of all, I would like express my sincere gratitude to my supervisor, Professor Silvio Gerli, for his trust, patience and guidance. I am especially thankful for giving me the opportunity to delve deeper into the world of GenAi through the internship he offered, which made this work possible.

I am also deeply grateful to my co-supervisor, Dr. Teresa Cigna, for her constant availability, insightful advice, and kind support throughout the development of this work.

A special thanks goes to my course mate: Sergio, Antonio, Robin, Babak, and Amelia for welcoming me into the group, for their steady encouragement over these two years and for introducing me to, and keeping me engaged with, the world of Data Science

I would also like to thank my friends from Verano and nearby: Lorenzo, Andrea, Riccardo, Manuel, Davide, and Lisa for their unstopped support and friendship throughout all the years we have known each other.

My heartfelt thanks also go to my former Bachelor colleagues: Luca, Dea, Silvia, Cristian, and Leidy, for always being there for me and continuing sharing memories together.

Finally, my deepest appreciation goes to my family, Mamma, Papà, Sami, and Chiara and all my relatives, for their unconditional love, patience, and faith in me. Their support has been my greatest strength throughout these years.

I thank you all for helping me get this far; reaching this milestone would not have been possible without your support and encouragement.