The Hallucination Problem in LLMs: Why Scaling Parameters Isn't Enough

Abstract

Hallucinations in Large Language Models is becoming a growing concern in recent years. We can observe a prominent increase in LLMs usage across various domains, such as chatbots, business assistants, and medical diagnosis. Even though various studies have investigated hallucination in LLMs but there is still the need for further research in this area, as the current understanding of hallucinations in LLMs is limited. Therefore, this paper seeks to address the research questions: Why are Large Language Models (LLMs) susceptible to hallucination, and to what extent is hallucination in relationship with the model parameter? By doing so in an experiment with Gemma-3 models (1B-12B parameters) using SimpleQA benchmark to reach a conclusion the urgent need for architectural improvements beyond simple parameter scaling. We also aim to provide a literature review for the field of LLMs, investigating categorization, metric, and mitigation strategies.

1 Introduction:

The rapid development of language models can be said to completely revolutionized the world. Large language models show a great deal of promise in a variety of natural language generation and interpretation tasks as well as real-world applications, and they even suggest advancements in artificial general intelligence. Reaching onwards from the most popular application of LLMs in content generation and curation, programming code design, and virtual assistants. They open a wide range of application opportunities across numerous scientific fields, and high-stakes domains like law and healthcare.

However, the widespread adoption is undermined by an inherent characteristic of LLMs: hallucinations, when the model generates response that contains unverifiable content. Their errors range from subtle contradiction to outright falsification, undercutting reliability in high-risk deployment. A clinical LLM could hallucinate out-of-date treatments, and a legal assistant could invent fake precedents. Solving hallucinations requires a systematic categorization of their taxonomy, causality, and mitigation practices—which paper will include.

2 Categories of Hallucination

The classification of hallucinations is a heavily debated area, often characterized as a challenging area due to the lack of consensus on definitions. Despite this ongoing debate, the most widely accepted distinction in hallucination taxonomy remains the division between intrinsic and extrinsic hallucinations.

According to Ahadian, intrinsic hallucination refers to internal inconsistencies within the generated text, wherein the model contradicts itself [1]. In other words, generated texts that contradict the source query[2]. Alternatively, extrinsic hallucinations involve outputs that cannot be verified using the provided source context or external knowledge bases. This makes the outcome unverifiable and possibly deceptive because the generated text is neither directly contradicted by nor supported by the information that is now available. [3]

Intrinsic Hallucination (create table):
For example, a model summarizing an article might state that a certain species of fish was first found in the Pacific Ocean and then later claim these fish are fresh water fish, which clearly indicates an internally inconsistent response.

Extrinsic Hallucination:
Consider a news article stating, “Tesla today announced the release of its new Model Q electric car. The Model Q boasts a range of 400 miles on a single charge, thanks to its advanced solid-state battery technology.” If a language model generates the output, “Tesla has unveiled its Model Q, an electric vehicle

with a 500-mile range and self-driving capabilities," this demonstrates extrinsic hallucination. The claims of a '500-mile range' and 'self-driving capabilities' are not verifiable using the provided source context; the original article neither confirms nor denies these features.

However, the field of LLMs is rapidly evolving; current day large language models are applied in a variety of diverse fields and scenarios. Additionally, modern LLMs are placing significant emphasis on user-centric interaction and most hallucinations are surface level factual errors. Therefore, some researchers have proposed the building of a new taxonomy upon the foundations of intrinsic/extrinsic classification. From a frequently cited survey on LLM hallucination, Lei Huang's team introduces namely factuality hallucination and faithfulness hallucination. Huang defines factual contradiction as situations wherein the LLM's output contains facts that can be grounded in real-world information, but present contradictions. Factual fabrication refers to instances where the LLM's output contains facts that are unprovable against confirmed knowledge in the real world.

Similarly, another survey by Yue Zhang creates another categorization of hallucination outputs, separated into three sections: (i) input-conflicting where LLMs generate content that deviates from the source input provided by users; (ii)context-conflicting, where LLMs generate content that conflicts with previously generated information by itself; (iii)fact-conflicting, where LLMs generate content that is not faithful to established world knowledge. [4]

Nevertheless, some critics disagree with such categorization. Yejin Bang disapproves of these categories and highlights that both Zhang and Huang failed to capture inconsistencies with the training data of the model. Bang claims "that an answer that is consistent with the training data of the model but is factually wrong because e.g. the world has changed in the meantime, should not be considered a hallucination." [2]

3 Root Causes of Hallucination

Recognizably, LLMs hallucinate for a variety of reasons, some of which are complex and not fully understood. This section focuses on the root causes of hallucination in large language models, divided into these key points: (i) data related causes (ii) model architecture & training (iii) inference & deployment factors (iiii) emerging perspectives.

3.1 Data Related Causes

The process of creating a new family of LLMs often starts with the pre-training phrase, and similarly across many different LLMS publicly available online sources stand out as a key fraction. Machine learning algorithms are crucially dependent on these data to find inter-dependencies and patterns to "learn" and acquire their general capabilities and factual knowledge.

However, the sheer quantity of data over the internet makes it an almost impossible task to evaluate data word by word. Furthermore, the precise way data is distributed and utilized within an LLM often remains opaque – a "black box" – and determining the optimal amount of data needed for various tasks is still an area of uncertainty [5]. The reliance on massive amounts of data would be a double-edged sword for the LLMs, because they have an intrinsic tendency to memorize and copy down training data. [3]

Consequently, problematic sources within pre-training, such as imitative falsehood and societal biases data, may be amplified by the LLM's algorithm into hallucinated outputs. Misinformation such as a faulty news report or rumor will inevitably introduce generation of false statements leading to imitative false hood[5]. Highlighting this, Yue Zhang notes that LLMs sometimes misinterpret spurious correlations, such as positionally close or highly co-occurring associations, as factual knowledge. [4] Since training data often reflects societal biases, stereotypes, and prejudices, models can generate biased or discriminatory outputs, even if it's not explicitly instructed to do so.

The critical impact of data related to hallucination is further highlights by an investigation from Nick McKenna, who demonstrates that LLMs are prone to two biases originating from pretraining. LLMs tend to memorize the exact sentences from their training data and statistical patterns of usage learned at the level of corpora. [6] Training data lacking pertinent information therefore leads to hallucinations, because the LLMs might resort to fabricated responses based on its memorized, yet inadequate, knowledge. This highlights what is known as a "knowledge boundary", which encompasses limitations such as the need for Up-to-date knowledge and issues related to copyright knowledge.

Example up-to-date knowledge: LLMs trained on outdated medical literature may hallucinate current treatments

3.2 Model Architecture & Training

The transformer architecture is the fundamental building block of all Language Models with Transformers the reason the transformer architecture successfully outperforms other architecture like Recurrent Neural networks is due to its unique "attention mechanism" concept, which critically lets the model focus on different parts of the input sequence when making each output token.[7]
However, this concept isn't flawless. Research from Michael Hahn discusses the limitations of the self-attention in complex conditions, which can contribute to hallucinations. Specifically, the author shows that self-attention's inability to model hierarchical structures and periodic regular languages can lead to hallucinations.[8] Architecturally, some degree of hallucination is unavoidable under the current self-attention mechanism.

Fine training refers to the process of applying certain techniques to a pre-trained model whose weight have already been updated through prior pre-training. Fine training aims to align the model to desired behaviors, in the spite of that models inevitably will encounter new information, possibly extending beyond the knowledge it acquired during pre-training. Zorik Gekhman studied the impact of integrating new factual knowledge through fine-tuning with the following findings: (1) Acquiring new knowledge via supervised fine-tuning is correlated with hallucinations. (2) LLMs struggle to integrate new knowledge through fine-tuning and mostly learn to use their pre-existing knowledge, [9] concluding plausibly a source of hallucinations

3.3 Inference & Deployment Factors

Hallucinations could also arise from inference factors, chiefly the ability of a LLM to comprehend and encode input text into meaningful representations. Encoder are a seriously complex computational heavy structure that involves multiple layers of varying features. Being vital to LLMs, encoders allow the model to discern fine-grained differences in meaning, and clears the way to perform a range of tasks. [10]

With that in mind, a faulty encoder will wrongly understand relationships between section data, and could result in erroneous generation that diverges from the input. However, while encoder is still used in many LLMs, its traditional structure of encoder-decoder has been dispensed and replaced with decoder-only models like in GPT-3 and LLaMA. Similarly, the design of the decoding strategy itself can contribute to hallucinations, such as top-k sampling, is positively correlated with increased hallucination.[11]

Hallucination Evaluation & Detection

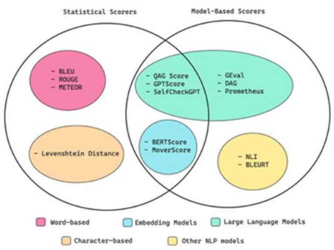
Before reaching into the actual specifications of hallucination benchmarks, it is important to acknowledge the differences between hallucination detection and hallucination evaluation. Although very similar hallucination detection is mainly directed at identifying if a hallucination exists at an instance level (one specific output). On the other hand, hallucination evaluation includes a system level scope and is more about the characteristics or system performance regarding hallucinations. While acknowledging the valuable distinctions between these concepts is crucial for a nuanced understanding, this essay will, for the sake of simplicity, treat them as a single, unified idea as hallucination evaluation.

Types of Hallucination Evaluation Methods
Various metrics have been developed to determine hallucination, this section mainly categorize them into statistical metrics, model-based metrics, and human evaluation.[12]

4.1 Statistical Metrics

One the simpler approaches to calculate hallucinations, statistical metrics are focuses on leveraging lexical(word-based) features to calculate information overlap and contradictions between generated and reference texts (ground truth). Dhingra's research works on this metrics, proposing PARENT (Precision And Recall of Entailed n-grams from the Table). Generated text is matched with the source table and target text using PARENT n-gram lexical entailment. The accuracy of the table-to-text task is reflected in the F1-score, which combines the entailment's precision and recall. Dhingra further shows that previous metrics such as ROUGE and BLEU are too reliant on the target text (refers to desired or reference output text), becoming problematic as the target text might not contain all the information from the source. [13]

TABLE 1 TYPES OF METRIC SCORERS [14]



First, let's define the terms based on the detection task:

True Positive (TP): The detector correctly identifies a hallucination as a hallucination.
True Negative (TN): The detector correctly identifies a non-hallucination (factual statement) as a non-hallucination.
False Positive (FP) / Type I Error: The detector incorrectly identifies a factual statement as a hallucination. (This is bad because you might discard correct information). *
False Negative (FN) / Type II Error: The detector incorrectly identifies a hallucination as a factual statement. (This is bad because you let an error slip through).

Metrics used to evaluate the performance of hallucination detection systems, including: TABLE 2

Metric	*Formula*	*Interpretation*	*Use Case*
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$	Overall correctness of the detector	Balanced datasets; general performance
Precision	$\frac{TP}{TP+FP}$	Percentage of flagged hallucinations that are actual hallucinations	When false positives are costly (e.g., medical diagnostics)
Recall(sensitivity)	$\frac{TP}{TP+FN}$	Percentage of actual hallucinations correctly detected	When false negatives are risky (e.g., legal advice)
F1-Score	$2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$	Harmonic mean of precision and recall	Balancing FP/FN trade-offs (default for PARENT [13])
Specificity	$\frac{TN}{TN+FP}$	Percentage of factual statements correctly identified	Avoiding over flagging (e.g., customer service chatbots)
False Positive Rate	$\frac{FP}{FN+TP}$	Percentage of factual statements wrongly flagged as hallucinations	Evaluating detector safety
False Negative Rate	$\frac{FN}{FN+TP}$	Percentage of hallucinations missed by the detector	Assessing coverage gaps
AUC-ROC	Area under ROC curve	Model's ability to distinguish classes across thresholds	Model's ability to distinguish classes among thresholds
AUC-PR	Area under PR curve	Precision-recall balance, especially for imbalanced data	Precision-recall balance, especially for imbalanced data

In addition to PARENT, BLEU, and ROUGE there are several more statistical methods including METEOR (Metric for Evaluation of Translation with Explicit Ordering) and Levenshtein Distance, which next I will quickly go through.

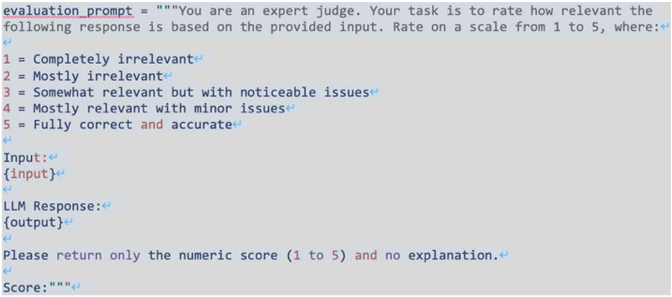
METEOR: An automatic metric for machine translation evaluation that is based on a generalized concept of unigram matching between the machine-produced translation and human-produced reference translations. [15]

Levenshtein Distance: A string metric for calculating the difference between two sequences. The smallest number of single-character modifications (insertions, deletions, or substitutions) needed to transform one word into another is known as the Levenshtein distance between two words. [16]

Nevertheless, statistical metrics have limited use in LLM hallucination detection because few reasoning mistakes are considered. Fundamentally it only handles lexical information and therefore doesn't take semantic into account. Resulting limited accuracy for evaluating LLM outputs that are often long and complex.

4.2 Model-based Metrics

Model-based metrics utilizes neural models, the opposite of statistical metrics. This method relying on NLP models are usually found to be more accurate and supposed to handle more complex and semantic variations. However, neural models can encounter errors that may propagate and negatively impact the precise measurement of hallucination.



```
evaluation_prompt = """You are an expert judge. Your task is to rate how relevant the
following response is based on the provided input. Rate on a scale from 1 to 5, where:

1 = Completely irrelevant
2 = Mostly irrelevant
3 = Somewhat relevant but with noticeable issues
4 = Mostly relevant with minor issues
5 = Fully correct and accurate

Input:
{input}

LLM Response:
{output}

Please return only the numeric score (1 to 5) and no explanation.

Score:"""
```

TABLE 3: SIMPLE SCORING FOR “OUTPUT RELEVANCE”

This method also includes what can be called as “LLM as a judge”, in other words using another LLM to evaluate LLM (system) outputs. These LLM judges utilizes LLM's innate reasoning abilities to evaluate other LLM generations. Not only is this cost effective compared to human based approaches, it is also a scalable detection method. Overall there is a diverse range of Model-based metrics, many with different domains or tasks. Below I'm going to list out in description some metrics in this category.

4.2.1 *BERTScore*

One of the earlier but most influential model-based detection metrics, BERTScore was first introduced in a research paper titled "BERTScore: Evaluating Text Generation with BERT". Fundamentally, BERTScore is based on a pre-trained BERT contextual embedding to evaluate the semantic similarity between pieces of text. [17] Importantly, BERT is a transformer-based machine learning model for nature language processing (NLP) introduced in 2018. At its time, it showed powerful ability to understand the meaning of words in its context and could also be easily fine-tuned for different tasks, this enable BERTScore to capture the nuances of word meanings in different contexts.

The following steps are needed to calculate BERTScore:

1. Token Representation: Mainly using the BERT model, the input text is tokenized into a sequence of word pieces to be than computed again with a Transformer encoder.

2. Similarity Measure: Making use of the vector representation, the cosine similarity of a reference token and candidate token is computed.

3.BERTScore: Every word in the reference sentence is used to find the most similar word in the candidate sentence, the average of the similarity score is the recall score. Oppositely, each word in the candidate sentence finds the most similar word in the reference sentence and is averaged to compute a precision score Finally, the precision and recall are calculated to assess and combined into a single F1 score which is the BERTScore.

$$R_{\text{BERT}} = \frac{1}{|x|} \sum_{x_i \in x} \max_{\hat{x}_j \in \hat{x}} \mathbf{x}_i^\top \hat{\mathbf{x}}_j \ , \quad P_{\text{BERT}} = \frac{1}{|\hat{x}|} \sum_{\hat{x}_j \in \hat{x}} \max_{x_i \in x} \mathbf{x}_i^\top \hat{\mathbf{x}}_j \ , \quad F_{\text{BERT}} = 2\frac{P_{\text{BERT}} \cdot R_{\text{BERT}}}{P_{\text{BERT}} + R_{\text{BERT}}} \ .$$

This metric has been shown to correlate better with human judgment on tasks like summarization than BLEU or ROUGE, since it captures meaning, not just exact wording. [18]

4.2.2 HalluCounter

This is a reference-free hallucination detection method (RFHD) that utilizes both response-response, query-response consistency and alignment patterns. HalluCounter can perform hallucination detection while obviating the need for an external knowledge base (KBs) or a correct reference text. This is particularly useful for closed-source LLMs where internal states like generation probabilities are inaccessible.

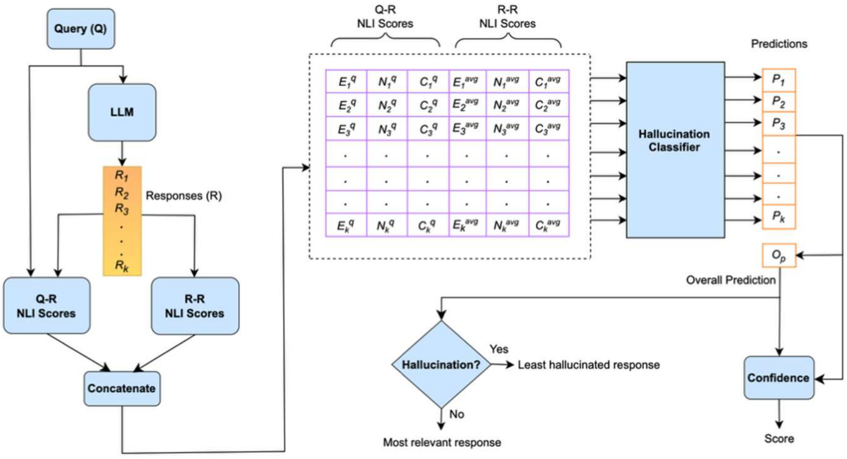

Query (Q)
LLM
R_1 R_2 R_3 . . . R_k
Responses (R)
Q-R NLI Scores
R-R NLI Scores
Concatenate
Q-R NLI Scores
R-R NLI Scores
E_1^q N_1^q C_1^q E_1^{avg} N_1^{avg} C_1^{avg}
E_2^q N_2^q C_2^q E_2^{avg} N_2^{avg} C_2^{avg}
E_3^q N_3^q C_3^q E_3^{avg} N_3^{avg} C_3^{avg}
E_k^q N_k^q C_k^q E_k^{avg} N_k^{avg} C_k^{avg}
Hallucination Classifier
Predictions
P_1 P_2 P_3 . . . P_k
O_p
Overall Prediction
Hallucination?
Yes
Least hallucinated response
No
Most relevant response
Confidence
Score

TABLE 4: HALLUCOUNTER[19] USES A PIPELINE WITH THREE MAIN STAGES

1. Extracting Natural Language Inference (NLI) Features:
    a) This stage generates multiple LLM responses to a query and then employs an NLI model to produce entailment, neutrality, and contradiction scores. These scores measure the relevance of each response to the original query and the consistency among the generated responses.
2. Hallucination Detection Classifier
    a) The NLI features are then fed into a classifier, a BERT-based model. This classifier's role is to predict whether each individual LLM-generated response is a hallucination or not.
3. Optimal Response Generation and Confidence Score Calculation (aggregation):
    a) Finally, this stage aggregates the individual response classifications into an overall prediction for the query using a majority vote. Based on this overall prediction, it selects an optimal response (prioritizing low contradiction if hallucinated, high entailment if not) and calculates a confidence score for its decision. [19]

4.2.3 Luna

Although Retrieval augmented generation (RAG) can effectively reduce LLM hallucinations in certain systems, LLMs can still often respond with nonfactual information that contradicts with the retrieved

The researchers at Luna addressed this issue with Luna, a lightweight RAG hallucination detection model that generalizes across multiple industry-specific domains and scales well for real-time deployment. Luna used a 440M parameter DeBERTa-large encoder and fine-tuned on real-world RAG data.

How Luna detects hallucinations: The core idea is NLI and Token-Level Support, Luna frames detection as an entailment problem, like a Nature Language Inference. Comparing and determining if the LLM response is supported by the RAG retrieved content. In addition, Luna at a granular level can provide predication at a token level which specific token in the response is supported by the context. [20]

4.2.4 G-EVAL

This framework uses LLMs, specifically GPT 4 and GPT 3.5 as the backbone of its evaluation. Proposing advantages such as higher correlation with human judgments and continuous scores that better reflect the difference in quality.

G-EVAL, a prompt-based evaluator, is divided into three main components:
1. Prompt: Defines the evaluation task and the desired evaluation criteria, in other words the specific aspects to be evaluated.

2. Auto Chain-of-Thought: Chain-of-thought is the sequence of intermediate reasoning steps LLM takes to arrive at the final answer. Using this, instead of manually designing detailed step-by-step instructions for every evaluation criterion, G-EVAL uses the LLM itself to generate these steps.

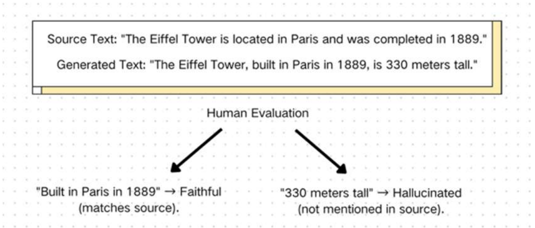
3.Scoring: After calling upon the scoring function, the LLM is provided with the complete prompt, automatically generated Cot, input content, target text. Then, G-EVAL directly performs the detection task by a form-filling paradigm. [21]

4.3 Human Hallucination Evaluation

Given the limitations of current automatic hallucination detection methods, human evaluation remains an essential approach. This typically involves two main strategies: (1) scoring, where human annotators assess the degree of hallucination on a graded scale, and (2) comparison**,** where annotators evaluate outputs against baseline references or ground-truth texts.

Comparison evaluations are simple and easy, which involves human annotators assessing whether generated text contains hallucinations compared to a trusted source. The annotators will need to mark each single claim as either faithful(supported), hallucinated(unsupported), or partially faithful.

TABLE 5: COMPARISON EVALUATION



Source Text: "The Eiffel Tower is located in Paris and was completed in 1889."

Generated Text: "The Eiffel Tower, built in Paris in 1889, is 330 meters tall."

Human Evaluation
"Built in Paris in 1889" → Faithful (matches source).
"330 meters tall" → Hallucinated (not mentioned in source).

5 Hallucination Mitigation Approaches

Current day large language models, with the unleashed power of nature language processing (NLP) are capable of analyzing inputs and generating the most sophisticated response. However, the inevitable phenomenon of hallucination remains one of most fundamentally challenges. Therefore, by mitigating this hallucination, especially in high-risk domains like medical-care, law, and enterprise environments, where inaccuracies can have several consequences. This work afterwards will focus on the strategies used to mitigate hallucination, categorized and targeted based on the root causes identified previously.

5.1 Mitigating Data Related Hallucinations

Data-related hallucinations can stem from low-quality, biased, or outdated pre-training data, leading LLMs to reproduce misinformation or fabricate responses. Mitigation strategies include rigorous data filtering/curation, dynamic model editing (e.g., ROME, MEMIT), and Retrieval-Augmented Generation (RAG) to supplement knowledge with verified external sources."

5.1.1 Data Filtering and Curation

An intuitive approach to reduce the presence of misinformation and bias is to using human labor to carefully select high quality sources one-by-one, validating the data during the process. Obviously, using such method we can ensure factual correctness of data while also minimizing the introduction of social biases.[3] However, this approach becomes problematic when dealing with massive quantities of data, as manual curation is inevitably inefficient for such large-scale tasks. Therefore, to construct information rich pretraining corpora, early approaches usually rely on heuristic filtering using hand-crafted rules.[23] An example of heuristic filtering is the removal of overly short or suspiciously long text. Additionally, deduplication methods help remove redundant content that would otherwise reduce dataset diversity and potentially bias model outcomes.

Nevertheless, these heuristic approaches cannot identify complex content noise and lead to suboptimal LLM performance.[24] Leading to the development of model-based data filtering, which gradually emerged as an optimal strategy for its ability to select high quality content. Model-based filtering successfully demonstrated its efficacy following preprocessing stages of datasets, showcasing substantial advancement in dataset quality and downstream LLM performance. Modern approaches like AutoPureData demonstrate the effectiveness of model-based filtering, combining AI-powered content flagging (achieving 91.5% F1 score with tools like LlamaGuard 2) and domain reliability checks to efficiently curate high-quality training data at scale.[25]

Wang's team outlined two main challenges with model-based approaches, stating that they: (1) resource heavy validation requires costly training, and (2) reliance on manually selected seed data that introduces human subjectivity. [24] They proposed their own strategy, which involves an efficient data filtering pipeline designed to tackle these specific issues directly. To counter the high cost and inefficiency of data validation, Wang et al. introduce an 'Efficient Verification Strategy' namely "Ultra-Fine web." This method uses a nearly-trained LLM, and then incorporating candidate corpora into the final training steps to rapidly assess data quality and its impact on LLM performance with significantly reduced computational expenditure.

5.1.2 Model Editing

The objective of model editing is to efficiently alter the behavior of LLMs within a specific domain without negatively impacting performance across other inputs. There are two main paradigm for model editing, integrating an auxiliary network with the original model without changing it (Preserve LLMs' Parameters), or adjusting the model parameters that contribute to the undesirable output(Modify LLMs' Parameters).[26]

a. Preserve LLMs’ Parameters

1. Memory Based Models: This methodology utilizes local model editors to update the behavior of base (pre-trained) models, allowing for the injection of updated knowledge or the correction of undesirable behaviors. Example including SERAC[27] and ROME[28], specifically SERAC avoids directly altering the base model’s weight. Instead, it uses an external memory to store edits and two auxiliary components, scope classifier and counterfactual model.
2. Additional Parameters: This paradigm introduces extra trainable parameters within the language models. The added parameters are trained on updated or corrected knowledge, allowing localized adjustments without disrupting the base model’s general capabilities. Meanwhile, the original model parameters remain static, frozen in place. T-Patcher is an example of such an approach, adding *task-specific patches* (small neural networks) to intermediate layers of the transformer, effectively fixing factual errors.[29]

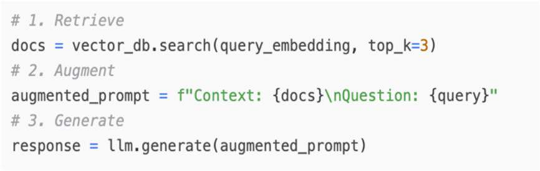
b. Modify LLMs’ Parameters

1. Locate-then edit: Mainly consist of two core stages, firstly it identifies the specific layers or parameters responsible for storing knowledge, afterwards modify them to correct hallucination behavior. For example, ROME [210] located the edits-related layer by destroying and subsequently restoring the activations and then updates the parameters of FFN in a direct manner to edit knowledge.
2. Meta-Learning: This approach uses an external hyper-network to learn the necessary parameter updates for modifying a mode’s behavior. However, showcasing key limitation in efficiency, due to its requirement of additional training overhead plus memory resources. In addition, contains the risk of unintended knowledge degradation when modifying parameters. [30]

5.1.3 Retrieval-Augmented Generation (RAG)

Retrieval-augmented generation is a technique for enhancing the accuracy and reliability of generative AI models with information fetched from specific and relevant data sources. [31] This capability significantly enhances transparency and lessen the probability of hallucination in AI-generated responses. In other words, this technique offers three key benefits: (i) Query Clarification (ii) Hallucination Reduction, minimizes the risk of models generating confident but factually wrong answers (iii) Implementation Efficiency, RAG is relatively easy to implement to such an extent that basic RAG functionality can be reach with as few as five lines of code. Making this method fast and a whole lot less expensive than additional datasets.

There are three main stages of RAG orderly they follow the steps of retrieval, augmentation, and generation. The retrieval phrase of RAG works by first converting the user input into a search query. This query is embedded into a vector and compared against a vector database (e.g. FAISS). Afterwards, the system will fetch the most relevant information based on semantic similarity. In the augmentation phase the content retrieved are added to the user’s prompt as context. Finally, during the generation phase the LLM will generate a response conditioned on both the query and the retrieved context.



```
# 1. Retrieve
docs = vector_db.search(query_embedding, top_k=3)
# 2. Augment
augmented_prompt = f"Context: {docs}\nQuestion: {query}"
# 3. Generate
response = llm.generate(augmented_prompt)
```

TABLE 6: EXAMPLE IMPLEMENT OF RAG

In terms of drawback, RAG approaches are crucially vulnerable to irrelevant, abnormal retrievals. Which will degrade output quality and increase the computational waste. Furthermore, the interaction between the retrieval and generation is often weak, leading to incomplete answers for multi-hop queries and redundant retrieval.

5.2 Mitigating Training-Related Hallucinations

Training-related hallucination in Large Language Models can stem from the transformer architecture and the deficiencies during the process of ore-training and fine-tuning. This following section provides a comprehensive overview of mitigation strategies organized to address these problems.

5.2.1 Addressing Pre-training Deficiencies

Mitigation approaches focus on architectural improvements and refinements to pretraining objectives. Recent advances in architectural focus on bidirectional context modeling, attention stabilization, and parallel inference frameworks to improve factual consistency. On the other hand, research on pe-training objective can roughly be divided into factuality-enhanced training, In-Context Pretraining, and Corruption-reconstruction objectives.

Bidirectional context modeling: Contrasting to traditional unidirectional autoregressive models, bidirectional context models is an innovative solution towards hallucinations. An example of such bidirectional architecture is BERT, showcasing the ability to allow attention to flow both in directions. [17] Yet, bidirectional context modeling may require more computational cost and increased memory usage. [32]

Attention Stabilization: The attention mechanism is fundamental to transformer-based LLMs like GPT, but meanwhile it also significantly contribute to hallucinations in some cases. Several strategies have been invented to promote higher attention. Such as Zeqiu Wu's team adopt inductive attention, which removes potentially uninformative attention links by injecting pre-established structural information to avoid hallucinations. [33]

Factuality-enhanced training: This method aims at training LLMs with factually rich plain-text corpora, primarily to enhance the factuality of the LLMs through exposure of data. Within this approach several strategies including knowledge graph integration, factuality-enhanced objectives, fact checking, and verification mechanisms are implemented. Nevertheless, the key component of factuality-enhanced training is TopicPrefix, which involves prepending a topic prefix to sentences in the factual corpus. Ideally, separating each sentence as a standalone fact; this will aid the "fragmentation" of information, an occurrence when wrong association of entities appear in independent documents with similar context. The introduction of TopicPrefix and sentence completion loss has been shown to significantly improve the factuality of LLMs. For instance, it reduces the named-entity factual error rate from 33.3% to 14.5%. [34]

5.2.3 Mitigating Misalignment During STF and RLHF

While supervised fine-tuning (SFT) and reinforcement learning from human feedback (RLHF) are powerful tools for reducing hallucinations in large language models (LLMs), improper implementation of these techniques can actually become a significant source of misalignment and subsequent hallucinations. This paradoxical situation arises because the fine-tuning process involves two critical phases that must be carefully managed: (1) the collection and annotation of high-quality instruction-following datasets, and (2) the actual fine-tuning of base LLMs using maximum likelihood estimation (MLE) [35].

Supervised Fine-tuning: In a way like pre-training, one approach to reduce hallucination during in supervised fine-tuning stage is curating the training data. Experimental results such ones made by Stephanie Lin suggest that LLMs fine- tuned on such curated instruction data demonstrate higher levels of truthfulness and factuality com- pared to LLMs fine-tuned on uncurated data. In addition, the volume of

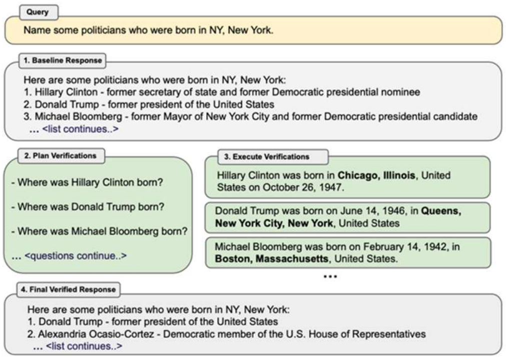
SFT is fairly acceptable and is within the scale capable for human marking. However, some previous work have point out that the SFT process may inadvertently introduce hallucination, because the LLMs are answering questions that is outside their knowledge sets.

Reinforcement Learning from Human Feedback (RLHF) : When supervised fine-tuned reach their limits RLHF appears as a approach that can further strengthen LLMs. The process consist of two main steps, **Reward Model Training**: A specialized reward model (RM) is developed to approximate human preference judgments. This model learns to assign calibrated reward scores to LLM-generated responses based on quality assessments. **Policy Optimization**: The supervised fine-tuned (SFT) model undergoes iterative refinement using reinforcement learning algorithms. The reward model's output guides this optimization process, with proximal policy optimization (PPO) being the most commonly employed algorithm for stable policy updates. [4] Although this method is useful in guiding LLMs in their knowledge understanding, the approach poses challenges in exhibiting over-conservatism in RL-tuned LLMs.

5.2.4 Mitigating Inference-Related Hallucinations

Inference-time mitigation strategies currently offer a complementary and flexible approach to de-hallucination. These methods address hallucinations during text generation, leveraging real-time adjustments without modifying the underlying model parameters. This section explores four key paradigms: (1) decoding strategies that steer generation toward factual consistency, (2) uncertainty estimation to identify and filter unreliable claims, and (3) auxiliary techniques like prompt engineering and human-in-the-loop verification. These inference-time miti/gation are effective in the sense that where they have great adaptability to new domains or knowledge updates. However, challenges persist—such as balancing computational overhead with accuracy and handling black-box model constraints.

Decoding Strategies:



Name some politicians who were born in NY, New York.

Here are some politicians who were born in NY, New York:
1. Hillary Clinton - former secretary of state and former Democratic presidential nominee
2. Donald Trump - former president of the United States
3. Michael Bloomberg - former Mayor of New York City and former Democratic presidential candidate
... <list continues..>

- Where was Hillary Clinton born?
- Where was Donald Trump born?
- Where was Michael Bloomberg born?
... <questions continue..>

Hillary Clinton was born in **Chicago, Illinois**, United States on October 26, 1947.

Donald Trump was born on June 14, 1946, in **Queens, New York City, New York**, United States

Michael Bloomberg was born on February 14, 1942, in **Boston, Massachusetts**, United States.

...

Here are some politicians who were born in NY, New York:
1. Donald Trump - former president of the United States
2. Alexandria Ocasio-Cortez - Democratic member of the U.S. House of Representatives
... <list continues..>

Figure 1: Chain-of-Verification (CoVe) method. Given a user query, a large language model generates a baseline response that may contain inaccuracies, e.g. factual hallucinations. We show a query here which failed for ChatGPT (see section 9 for more details). To improve this, CoVe first generates a plan of a set of verification questions to ask, and then executes that plan by answering them and hence checking for agreement. We find that individual verification questions are typically answered with higher accuracy than the original accuracy of the facts in the original longform generation. Finally, the revised response takes into account the verifications. The factored version of CoVe answers verification questions such that they cannot condition on the original response, avoiding repetition and improving performance.

TABLE 7: CoVe METHOD

As mentioned before, decoding strategies are curial in determining how we choose output tokens from the probability distribution generated by models. Shehazaad Dhuliawala from Meta AI studied and developed Chain-of-Verification (CoVe) method, following the line of research of encouraging language models to first generate internal thoughts or reasoning chains before responding. Their approach is to first generate a list of questions when given the initial draft response, then systematically answering those question for an improvised response. Generating separate verification questions helps get more accurate facts, improving the overall response compared to just using the original long answer. To prevent the model from repeating its own mistakes, they further improved this by separating the verification steps and carefully managing what context the model sees, leading to even better performance. Each of these steps is performed accessing the same base LLM, capable of being prompted with general instruction in either a few-shot or zero-shot fashion. They showed that CoVe significantly decrease the generation of factually incorrect information; furthermore, improve precision in longform generation (e.g. 28% from the few-shot baseline), with CoVe effective providing a guideline for a model's self-correction. Otherwise, it is not perfect, it is limited on directedly stated factual inaccuracies and cannot address other forms of hallucinations. Its complexity of implementations, especially with the factored variants and factor + revise steps combined with multiple steps also makes it computational expensive.

Uncertainty Estimate:

This approach discusses that the inherent uncertainty in Large Language Models outputs themselves can be a useful indicator for mitigating hallucination also during the inference process. The core idea is that, once the LLM's confidence in its generated output is accurately estimated to be highly uncertain, we can predict these responses be more likely hallucinated. The user or automated systems then can filter out this likely fabricated response and attempt to rectify them. Yue Zhang's paper mainly outlines three primary approaches to estimate the uncertainty in LLMs: logit-based estimation, verbalize-based estimation, and consistency-based estimation.

1. Logit-based Estimation:
   a) This method relies on accessing the internal state of the LLMs, specifically the logits (raw scores). In addition, also accessing the probabilities assigned to each potential token during generation. Next, the uncertainty is calculated at a token level (e.g. lower probability for the chosen token indicated higher uncertainty). Neeraj Vershney exclaimed this logit-based method to detect "false concepts" in LLM outputs, once detected the hallucinations are mitigated using auxiliary retrieval-augmented LLMs. [37]
2. Verbalize-based Estimation:
   a) The second approach relies on prompting the LLM to express its own uncertainty verbally. Therefore, the effectiveness of this method strongly depends on the LLM's instruction-following capabilities, and its own ability to introspect its confidence. An example prompt requesting the LLM to express their confidence may be "Please answer this question… and provide your confidence score (from 0 to 100)." In addition, research suggested the use of chain-of-thought prompts to potentially enhance the reliability of this method. [38] Nevertheless, a significant issue is that LLMs often will exhibit overconfidence when asked to present their confidence level, this fundamentally problem crucially makes this method limited.
3. Consistency-based Estimation:
   a) Consistency-based estimation proceeds on the basis that if an LLM is hallucinating or in doubt when responding to a question, then it would behave the same for similar questions. Approaches include sampling several responses (e.g., with temperature > 0) then measuring their consistency, with greater inconsistency indicating greater chance of hallucination. Numerous works have applied the method for hallucination detection. For instance, SELFCHECKGPT [39] was a pioneering model that employed consistency-based uncertainty estimation in a zero-resource, black-box setting, evaluating consistency of responses using many metrics. However, a accurate measurement for all the different responses' consistency remains a troubling issue for all.

Prompt engineering and other innovative strategies:

Prompt Engineering: Existing research has found that hallucination can vary based on the prompts given by the users. For instance, a user may receive different answers from a LLM with different prompts. Consequence, engineering more effective prompts could a workable method to mitigate hallucination.

Example Prompt: If you don't know, or don't fully understand the answer to a question, do not share the false information

Human-in-the-loop: The intervention of a human user is seen to clarify intent and guide the LLM towards more accurate and relevant information. This strategy notes that hallucinations that may arise from a misalignment between (retrieved) knowledge and the user's actual question or intent. It addresses such issue by involving a human user in the process to refine queries or clarify the relationship between LLMs query and available knowledge. The MixAlign framework of such kind is reportedly to no only reduce hallucination but also to enhance the overall quality of outputs. [40]

Multi-agent Interactions: This strategy's core idea is that instead of relying on a single model's output, multiple agents will independently generate responses and then collaborate—often through debate or discussion—to reach a consensus. Best case scenario, this will filter out individual hallucinations and improve overall factuality. Wanyu Du's benchmark for factual accuracy in LM-generated biographies, showed that multiple LLMs significantly reduced hallucinations compared to single-agent outputs. [41] Similarly, Zhenhailong Wang introduced a cost-effective approach for mitigation. Their method employs a single LLM self-collaborating with multiple personas (e.g., domain-specific "experts"), iteratively refining outputs while minimizing computational burdens. [42]

6 Experiment

6.1 Experiment Design

This experiment aims to study evaluates the relationship between LLM parameter count and hallucination rates. Proposing the hypothesis that a large model perimeter will exhibits a lower rate of hallucination in terms of factual knowledge, while keeping constant the model architecture and training protocols. Aiming with two major research questions: (i) To what extent is the hallucination rate effected by the model size (1B to 12B parameters) (ii) What patterns emerge in hallucination types (e.g., factual errors vs. fabrications) across scales.

6.2 Models Selected

For this experiment we selected three separate Gemma-3 models from Google, which are multimodal models with pretrained and instruction-tuned variants. Gemma-3 models can handle 128,000 token context window and can reply in JSON formats. In additions, Gemma-3 is computationally highly efficient, effectively making this model suitable for local installation with personal computers. [43] Gemma-3 models are available in three different four sizes: 1B, 4B, 12B, and 27B(billion) parameters. Unfittingly, the 27B model was hard to implement locally and had to be abandoned for the overall efficiency of the experiment.

6.3 Benchmark Dataset

SimpleQA focuses on evaluating the ability of language models to answer short factual questions, which contains 4,326 short, fact-seeking questions. SimpleQA features a diverse variety of questions created by AI trainer and validated again with another trainers. The questions in this dataset are designed with specific criteria: they require unique answers, are definitive regardless of the time period, and present a significant challenge. Visible in the figure, SimpleQA features a diverse range of topics, with Science & Technology leading the largest fractions (n=858). As for diversity in sources, wikipedia.com is by far the biggest source (one of the sources for 3.5k of 4.3k questions), followed by fandom.com (410 questions), ac.uk (154 questions), and imdb.com (121 questions). [44]

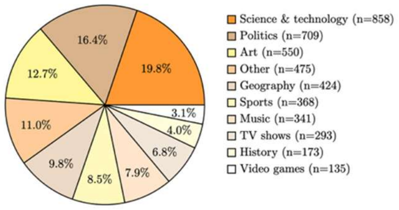


TABLE 8: TOPIC DISTRIBUTION [44]

6.4 Procedure

During this experiment, identical prompts were used as inputs for each model, with the questions extracted from the SimpleQA benchmark using Python coding and formatted accordingly. The API from the locally deployed models in LM Studio was utilized to generate outputs, with the answers saved to a JSON file. To evaluate the quality of these generated answers, the Python script evaluate_hallucinations.py was employed. This script assesses the model's tendency to hallucinate by comparing its answers against the known correct answers from the SimpleQA benchmark (https://github.com/Cheezecats/Hallucination-Evaulation). It calculates a hallucination rate based on whether the ground truth answer is present in the model's response, providing a quantitative measure of the model's reliability. The script saves the raw responses to a JSON file and prints the hallucination rate and number of correct answers. It is important to acknowledge that our hallucination detection method is rather basic, relying on a simple string match, and may not catch more nuanced or paraphrased correct answers. However, ideally this would not propose any significant differences to our evaluation because the SimpleQA benchmark is specifically design in such a way that there is only one answer, without doubt. Therefore, it is likely that any paraphrased answer will not meet those requirements and would be considered hallucinations. In addition, for the spite of experimental accuracy all models were ran on the same hardware to eliminate API variability, and we ran each trail at least three times to check for any major errors that might happen.

Component	**Specification**
Hardware	NVIDIA RTX 4080 (16GB VRAM), 64GB RAM
API	LM Studio (local inference, OpenAI-compatible endpoint at http://127.0.0.1:1234)
Generation Config	Temperature=0.7, Top-p=0.9, Max tokens=30000
Evaluation Metric	Hallucination rate = (Incorrect Answers) / (Total Answers) × 100

6.5 Results

Table 9: Performance of Gemma Models

Model	Parameters	Hallucination Rate	Correct Answers	Incorrect Answers
Gemma-1b	1	97.11	125	4201
Gemma-4b	4	95.19	208	4118
Gemma-12b	12	93.85	266	4060

We found that by a slight margin, larger models(without RAG) showed lower hallucination rate. For example Gemma-12b had a hallucination rate of 93.85% compared to the 1b model's 97.11% hallucination rate. A manual inspection of 50 of the models' incorrect responses revealed that 62% of hallucinations involved plausible but incorrect facts (e.g., "Leipzig 1877 honored Paul Morphy" vs. correct "Adolf Anderssen"). The rest of 38% percent are completely fabricated and are not backed out by any events.

Noticeably, the absolute improvements were modest, suggesting model size alone doesn't resolve hallucination. The 12B model reduced hallucinations by only 3.26 percentage points compared to the 1B variant, despite a 12x parameter increase. Arguably, LLM don't benefit significantly on factual context simply by scaling along; instead these LLMs will need architectural improvements or training data curation for hallucination mitigation.

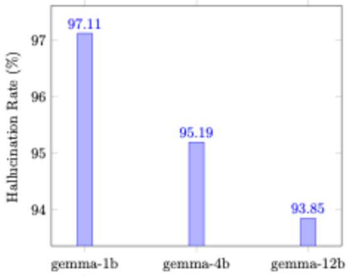


TABLE 10: HALLUCINATION RATES

6.6 Limitations

Admittedly, this current research has several limitations. There is limited resource to performance more evaluations, the current 4,326 QA from SimpleQA is not sufficient in vivificating the generalizability of such LLMs with other domains. Its limitation only to the field of English language also does not represent constant performance perhaps for others languages. In addition, due to the consideration of simplicity the current binary classification does not capture the degree of error severity. Adding on, a fixed model setting such as the temperature(0.7) might also be problematic because it may not reflect the optimal deployment settings in real life situations. The relatively small amounts of experiment repeats could also propose a problem as LLM do not perform the same all the time and differences in responses are unavoidable.

7 Discussion + Future Work

7.1 Discussion

Our research provides valuable practical implication for real life deployment. First, we outlined that current Gemma variants require verification mechanism (e.g. RAG) to improve performance on factual task. Secondly, the unworthy tradeoff between model size, computational requirements, and hallucination rate raises concerns against upsizing in resource-constrained scenarios.

Furthermore, the consistent high hallucination rate (>93% across all models) underscore the unreliability of unsupervised LLMs for critical applications like medical diagnostics or legal advice. Outlining that smaller, specialized fact checking models are a necessity to accurately validate the outputs of a larger less specific model.

7.2 Future Work

The limitations and insights uncovered in this study point to two major critical avenues for future research.

1. Hybrid Structures:
   a) Extending outwards, we can focus on increasing model parameters of hybrid LLMs, such as those with retrieval-augmented generation. By testing the differences between these models, we can observe effect when responses are anchored with verified sources.
2. Evaluation Paradigm Shifts
   a) Instead of a binary classification of response, in our future research we can work towards introducing tiers of correctness (partially correct, mostly correct) to capture a wider range of nuances. In addition, work towards evaluating both reasoning and factual ability, gaining a bigger overview of a model's ability. One possibly direction is to train the and command the models themselves to output a confidence score and degree of uncertainty.

Meanwhile, the ethical considerations of such research should also be seriously considered. RAG extracting sources from the internet should not come at the cost of introducing bias, even if it is a "trusted" source. It is also equally important to focus on the disturbing contradiction between performance vs. transparency. Larger models better represent a black box; on the other hand, they also produce more nuanced response, these two aspects will require a certain level of balance.

8 Conclusion

In recent years, LLMs have gained significant across every domain for their astonishing understanding and generations capabilities, creating revolution in healthcare, law, and education. However, hallucinations, responses where the model either misunderstood the user's prompt or fabricated against grounded real-world facts; this is a critical challenge we currently face in fully impeding LLMs in practical applications. Therefore, this article addresses such a problem offering a comprehensive review and an experiment diving into real world scenarios. The survey examines key aspects of LLMs, offering a well structure perspective on the current state of LLM research and the underlying problems awaiting to be solved.

We first examined the categorization of hallucinations; while also acknowledging differences in current understanding, we paid close attention to the separation of intrinsic vs. extrinsic hallucination. Afterwards, our article provides a general overview of the different approach and metrics of hallucination evaluation. Specifically, divided into three sections, statistical based, model based, and human based evaluations. Thirdly, our review of mitigation strategies provides a precise rundown of the current state of LLMs. Also bringing forth many examples of limitations and challenges faced by current methods, highlighting the continuous need for methodology improvements.

In the end we conducted an experiment demonstrating the current limitations of LLMs, exploring the correlation between model parameter size and hallucination. Though basic and limited in certain details, our investigation still provides insights into mitigating hallucinations. The results suggest that hybrid architectures, especially when combined with retrieval-augmented generation, hold promise for improving factual accuracy in question-answering scenarios.

In the end, it is important to remember that amide the complex landscape of large language models and hallucinations artificial intelligence should always serve for the good of human kind. To see the current challenges of hallucination as also an ethic imperative, as well as a technical one.

9 Acknowledgements

The author would like to wholeheartedly thank the entire team from Lumiere Individual Research Program for their constant support and motivation during the writing of this article. In addition, especially gratitude to Spencer Johnson for his mentorship throughout the program.

[1]Ahadian, P., & Guan, Q. (2025). A survey on hallucination in large language and foundation models. *Preprint*. https://doi.org/10.20944/preprints202504.1236.v1

[2]Bang, Y., Ji, Z., Schelten, A., Hartshorn, A., Fowler, T., Zhang, C., Cancedda, N., & Fung, P. (n.d.). HalluLens: LLM Hallucination Benchmark. arXiv.org. https://arxiv.org/abs/2504.17550

[3] Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., & Liu, T. (2024). A survey on hallucination in large language models: principles, taxonomy, challenges, and open questions. ACM Transactions on Office Information Systems. https://doi.org/10.1145/3703155

[4] Zhang, Yue et al. (2023) Siren's song in the AI Ocean: A survey on hallucination in large language models, arXiv.org. Available at: https://arxiv.org/abs/2309.01219 (Accessed: 02 May 2025).

[5] Hadi, M. U., Tashi, Q. A., Qureshi, R., Shah, A., Muneer, A., Irfan, M., Zafar, A., Shaikh, M. B., Akhtar, N., Wu, J., & Mirjalili, S. (n.d.). A Survey on Large Language Models: Applications, Challenges, Limitations, and Practical Usage. *TechRxiv*. https://doi.org/10.36227/techrxiv.23589741.v1

[6] McKenna, N., Li, T., Cheng, L., Hosseini, M., Johnson, M., & Steedman, M. (2023). Sources of hallucination by large language models on inference tasks. *arXiv*. https://doi.org/10.18653/v1/2023.findings-emnlp.182

[7] CryptoGPTo. (2023, March 10). Introduction to large language models and the Transformer architecture. *Medium*. https://rpradeepmenon.medium.com/introduction-to-large-language-models-and-the-transformer-architecture-534408ed7e61

[8] Hahn, M. (2020). Theoretical Limitations of Self-Attention in neural sequence models. *Transactions of the Association for Computational Linguistics*, *8*, 156–171. https://doi.org/10.1162/tacl_a_00306

[9] Gekhman, Z., Yona, G., Aharoni, R., Eyal, M., Feder, A., Reichart, R., & Herzig, J. (2024). Does Fine-Tuning LLMs on new knowledge encourage hallucinations? *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 7765–7784. https://doi.org/10.18653/v1/2024.emnlp-main.444

[10] Ghafforov, S. (2024, November 29). Understanding Encoders and embeddings in Large Language Models (LLMs). *Medium*. https://medium.com/@sharifghafforov00/understanding-encoders-and-embeddings-in-large-language-models-llms-1e81101b2f87

[11] Dziri, N., Madotto, A., Zaiane, O., & Bose, A. J. (2021). Neural Path Hunter: Reducing hallucination in dialogue systems via path grounding. arXiv preprint arXiv:2104.08455.

[12] Ji, Z. *et al.* (2024) *Survey of hallucination in natural language generation*, *arXiv.org*. Available at: https://arxiv.org/abs/2202.03629 (Accessed: 10 May 2025).

[13] Dhingra, B., Faruqui, M., Parikh, A., Chang, M., Das, D., & Cohen, W. (2019). Handling Divergent Reference Texts when Evaluating Table-to-Text Generation. *arXiv*. (*LLM Evaluation Metrics*, n.d.)https://doi.org/10.18653/v1/p19-1483

[14] *LLM Evaluation Metrics: The Ultimate LLM Evaluation Guide - Confident AI*. (n.d.-c). https://www.confident-ai.com/blog/llm-evaluation-metrics-everything-you-need-for-llm-evaluation

[15] Banerjee, S., & Lavie, A. (2005). METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. *Meeting of the Association for Computational Linguistics*, 65–72. https://www.cs.cmu.edu/~alavie/METEOR/pdf/Banerjee-Lavie-2005-METEOR.pdf

[16] Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady, 10(8), 707–710. https://doi.org/10.1007/BF00829252

[17] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 4171–4186).

[18] Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2020). BERTScore: Evaluating Text Generation with BERT. *arXiv (Cornell University)*. https://arxiv.org/pdf/1904.09675.pdf

[19] Urlana, A., Kanumolu, G., Kumar, C. V., Garlapati, B. M., & Mishra, R. (2025, March 6). *HalluCounter: Reference-free LLM hallucination detection in the wild!*. arXiv.org. https://arxiv.org/abs/2503.04615

[20] Belyi, M., Friel, R., Shao, S., & Sanyal, A. (2024, June 3). *Luna: An Evaluation Foundation Model to Catch Language Model Hallucinations with High Accuracy and Low Cost*. arXiv.org. https://arxiv.org/abs/2406.00975

[21] Liu, Y., Iter, D., Xu, Y., Wang, S., Xu, R., & Zhu, C. (2023). G-Eval: NLG Evaluation using Gpt-4 with Better Human Alignment. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. https://doi.org/10.18653/v1/2023.emnlp-main.153

[22] Sun, Y. (2010). Mining the Correlation between Human and Automatic Evaluation at Sentence Level. *Language Resources and Evaluation*. http://www.lrec-conf.org/proceedings/lrec2010/pdf/87_Paper.pdf

[23] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, *21*(140), 1–67. https://jmlr.org/papers/volume21/20-074/20-074.pdf

[24] Wang, Y., Fu, Z., Cai, J., Tang, P., Lyu, H., Fang, Y., Zheng, Z., Zhou, J., Zeng, G., Xiao, C., Han, X., & Liu, Z. (2025, May 8). *Ultra-FineWeb: Efficient Data Filtering and verification for high-quality LLM Training Data*. arXiv.org. https://arxiv.org/abs/2505.05427

[25] Vadlapati, P. (2024). AutoPureData: Automated filtering of undesirable web data to update LLM knowledge. *Journal of Mathematical &amp; Computer Applications*, 1–4. https://doi.org/10.47363/jmca/2024(3)e121

[26] Yao, Y., Wang, P., Tian, B., Cheng, S., Li, Z., Deng, S., Chen, H., & Zhang, N. (2023). Editing large language models: problems, methods, and opportunities. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 10222–10240. https://doi.org/10.18653/v1/2023.emnlp-main.632

[27] Mitchell, E., Lin, C., Bosselut, A., Manning, C. D., & Finn, C. (2022b, June 13). *Memory-based model editing at scale*. arXiv.org. https://arxiv.org/abs/2206.06520

[28] Meng, K., Bau, D., Andonian, A., & Belinkov, Y. (2023, January 13). *Locating and editing factual associations in GPT*. arXiv.org. https://arxiv.org/abs/2202.05262

[29] Huang, Z., Shen, Y., Zhang, X., Zhou, J., Rong, W., & Xiong, Z. (2023b, January 24). *Transformer-patcher: One mistake worth one neuron*. arXiv.org. https://arxiv.org/abs/2301.09785

[30] Hahn, M. (2020). Theoretical Limitations of Self-Attention in neural sequence models. *Transactions of the Association for Computational Linguistics*, *8*, 156–171. https://doi.org/10.1162/tacl_a_00306

[31] Merritt, R. (2025, January 31). *What Is Retrieval-Augmented Generation aka RAG | NVIDIA Blogs*. NVIDIA Blog. https://blogs.nvidia.com/blog/what-is-retrieval-augmented-generation/

[32] Li, Z., Yang, X., Gao, Z., Liu, J., Li, G., Liu, Z., Li, D., Peng, J., Tian, L., & Barsoum, E. (2024, June 19). *Amphista: Bi-directional Multi-head decoding for Accelerating LLM inference*. arXiv.org. https://arxiv.org/abs/2406.13170

[33] Wu, Z., Galley, M., Brockett, C., Zhang, Y., Gao, X., Quirk, C., Koncel-Kedziorski, R., Gao, J., Hajishirzi, H., Ostendorf, M., & Dolan, B. (2021). A controllable model of grounded response generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*(16), 14085–14093. https://doi.org/10.1609/aaai.v35i16.17658

[34] Lee, N., Ping, W., Xu, P., Patwary, M., Fung, P., Shoeybi, M., & Catanzaro, B. (2022, June 9). *Factuality Enhanced Language Models for Open-Ended Text Generation*. arXiv.org. https://arxiv.org/abs/2206.04624

[35] Wang, Y., Kordi, Y., Mishra, S., Liu, A., Smith, N. A., Khashabi, D., & Hajishirzi, H. (2023). *SELF-INSTRUCT: Aligning language models with self-generated instructions*. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, Volume 1: Long Papers (pp. 13484–13508). https://aclanthology.org/2023.acl-long.754.pdf

[36] Lin, S., Hilton, J., & Evans, O. (2022). TruthfulQA: Measuring How Models Mimic Human Falsehoods. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. https://doi.org/10.18653/v1/2022.acl-long.229

[37] Varshney, N., Yao, W., Zhang, H., Chen, J., & Yu, D. (2023b, August 12). *A stitch in time saves nine: Detecting and mitigating hallucinations of llms by validating low-confidence generation*. arXiv.org. https://arxiv.org/abs/2307.03987

[38] Xiong, M., Hu, Z., Lu, X., Li, Y., Fu, J., He, J., & Hooi, B. (2024a, March 17). *Can LLMS express their uncertainty? an empirical evaluation of confidence elicitation in llms*. arXiv.org. https://arxiv.org/abs/2306.13063

[39] Manakul, P., Liusie, A., & Gales, M. J. F. (2023b, October 11). *SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models*. arXiv.org. https://arxiv.org/abs/2303.08896

[40] Zhang, S., Pan, L., Zhao, J., & Wang, W. Y. (2024, June 13). *The knowledge alignment problem: Bridging human and external knowledge for large language models*. arXiv.org. https://arxiv.org/abs/2305.13669

[41] Du, W., Raheja, V., Kumar, D., Kim, Z. M., Lopez, M., & Kang, D. (2022, March 16). *Understanding iterative revision from human-written text*. arXiv.org. https://arxiv.org/abs/2203.03802

[42] Wang, Z., Mao, S., Wu, W., Ge, T., Wei, F., & Ji, H. (2024, March 26). *Unleashing the emergent cognitive synergy in large language models: A task-solving agent through multi-persona self-collaboration*. arXiv.org. https://arxiv.org/abs/2307.05300

[43] Gemma 3. (n.d.-a). https://huggingface.co/docs/transformers/model_doc/gemma3

[44] Wei, J., Karina, N., Chung, H. W., Jiao, Y. J., Papay, S., Glaese, A., Schulman, J., & Fedus, W. (2024, November 7). *Measuring short-form factuality in large language models*. arXiv.org. https://arxiv.org/abs/2411.04368