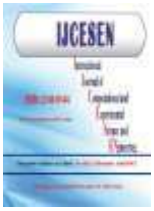
International Journal of Computational and Experimental Science and ENgineering (IJCESEN)

Vol. 11-No.3 (2025) pp. 4126-4137
http://www.ijcesen.com



ISSN: 2149-9144

Research Article

Understanding and Mitigating Strategies for Large Language Model (LLMs) Hallucinations in HR Chatbots

Rishab Bansal[1]*, Reena Chandra[2], Karan Lulla[3]

[1]Independent researcher, Fremont, CA, USA
*** Corresponding Author Email:** connect.rishabbansal@gmail.com - **ORCID:** 0009-0002-5348-0872

[2]Independent Researcher, San Francisco, CA, USA
Email: reenachandra11@gmail.com - **ORCID:** 0009-0001-8061-1084

[2]Independent Researcher, San Francisco, CA, USA
Email: kvlulla16@gmail.com - **ORCID:** 0009-0007-7491-4138

Article Info:

DOI: 10.22399/ijcesen.2471
Received: 22 March 2025
Accepted: 20 May 2025

Keywords

Large Language Models, LLM, Hallucination, RAG, Retrieval Augmented Generation, Prompt Engineering, Fine Tuning, Output Verification

Abstract:

Large language models are widely used in enterprise workflows, particularly in human resources and internal communication using chatbots. Although they provide efficiency and shorter turnaround times, their tendency to hallucinate—generating plausible but factually incorrect information—is a significant concern. This paper provides a comprehensive review of the problem statement and the solutions studied. It starts with defining and evaluating the causes and types of hallucinations particular to HR applications. The research also explores industry use cases and implements mitigating measures such as retrieval-augmented generation (RAG), confidence rating, abstention mechanisms, prompt engineering, domain-specific fine-tuning, and post-generation fact-checking. Using accessible empirical data, the research assesses the limitations, scalability, and effectiveness of various methods. Important research gaps are found, including the absence of HR-specific hallucination benchmarks, difficulties in uncertainty estimates, and the necessity of ongoing domain knowledge integration. Aiming to create reliable and grounded AI systems for HR and corporate support, the article ends by suggesting practical directions for future research and development.

.

1. Introduction

Large language models (LLMs) are a major technological development that quickly moved from research labs to useful applications in corporate settings [1]. Aiming to increase productivity, automate operations, and improve information access, companies are increasingly using LLMs to drive a range of internal tools and systems. Among the notable uses are internal chatbots for staff support, virtual assistants for information sharing, tools for document or meeting summaries, and even systems meant to help with HR operations, including hiring or employee onboarding. The possible advantages are significant, offering simplified procedures, quick access to data, and less labor for human employees. Though LLMs have great power, a fundamental vulnerability—the phenomenon known as hallucinations—tethers their general acceptance.

Broadly stated, hallucinations are the tendency of LLMs to provide responses that seem plausible, logical, and confident yet are factually inaccurate, nonsensical, incongruous with the underlying material, or completely invented. This problem seriously compromises the dependability and integrity of systems driven by LLM. This work addresses the problem of LLM hallucinations in relation to internal company chatbots and human resources (HR). We chose this area due to the significant risks associated with the generated and managed data. From hiring to termination, HR operations manage sensitive employee data, distribute important corporate policies, and guarantee legal compliance. They share important corporate rules and impact key facets of the employment lifecycle—from recruiting to termination—while ensuring legal compliance. For instance, an HR chatbot providing false information regarding leave entitlements, benefits eligibility, or workplace conduct regulations could cause major

employee confusion, legal non-compliance, and loss of faith in HR systems. Likewise, prejudices ingrained or imagined by LLMs, helping with performance assessment or recruiting, could support structural inequality.

A thorough overview of LLM hallucinations with an emphasis on manifestations, hazards, and mitigating techniques in HR and corporate internal chatbot applications is presented in this work We consolidate results from recent scholarly publications on LLM hallucinations, investigate how these hallucinations show in an HR environment, and go over methods to uncover and reduce them. Analyzed for their efficacy and fit in business environments, key techniques include retrieval-augmented generation, confidence estimate, prompt engineering, fine-tuning, and post-hoc fact-checking. We also draw attention to research shortcomings, including the need for domain-specific standards and the difficulty of guaranteeing current, company-specific information; we then suggest future avenues of research. The aim is to educate engineers and practitioners deploying internal chatbots on current knowledge on this topic and useful approaches to reduce hallucinations, therefore enhancing the dependability and credibility of HR artificial intelligence assistants.

2. Definitions of Hallucinations

We should realize that the word "hallucination" is used figuratively. LLM hallucinations are artifacts of their probabilistic character, training data, design constraints, and other elements, unlike human hallucinations, which are perceptual events usually connected to specific conditions. [2] LLMs lack consciousness or subjective experience; their "hallucinations" explain traits of the produced output instead of a cognitive process. Although the name captures the idea of generating apparently realistic but fake information, the anthropomorphic term can be misleading. [2]

Literature offers various overlapping definitions, describing hallucinations as

- Outputs deviating from user input or the model's training data. [3]
- The content generated is plausible-sounding but factually incorrect, inconsistent, irrelevant, nonsensical, or entirely fabricated. [2]
- Outputs ungrounded in actual data, external reality, or provided source content. [4]
- Inconsistencies between the LLM's output and a computable ground truth function. [5]

3. Reasons for Hallucination

3.1 Knowledge Gaps and Long-Tail Data

LLMs are trained on vast but finite corpora. If a question deviates from the observed distribution, say a specialized or company-specific question, the model may "fill in" missing information with its best guess. [6] According to a recent poll, LLMs struggle with domain-specific or rare information but perform well on popular topics (where training data is plentiful), often resorting to falsification for those long-tail searches. [7] In business environments, a significant amount of HR knowledge, such as internal policies and proprietary procedures, is not commonly found in the public internet material that mainstream LLMs were trained on. Therefore, a broad model may lack expertise in this domain-specific information. [7] When asked about something it never learned—say, a company policy—the model often cannot acknowledge ignorance; instead, it generates a plausible-sounding response from broad patterns, most likely a hallucination.

3.2 Lack of Access to Authoritative Data

Usually for legal and privacy considerations, public LLMs are not trained on private or confidential files. [7] This implies that the training of the base model cannot access internal corporate regulations or modern HR guidelines. This knowledge difference drives the model to create solutions in fields lacking or inaccessible training data. [7] An HR bot might be questioned about a confidential policy, for example; if the model's training has no record of that policy, a hallucinated response is most likely if no mitigating action is in place.

3.3 Outdated Knowledge and Temporal Issues

LLMs also have a defined knowledge cut; they cannot know information beyond the time of their training data collection. An unaugmented model would provide an outdated response (essentially a hallucination in the current context) or respond with something that sounds sensible if corporate policies have changed or regulations have been updated since the knowledge cutoff of the model. Researchers find that when LLMs are asked questions beyond their chronological knowledge, they often fabricate facts or provide answers that were once correct but are now outdated. [7] For HR, this is a major issue since policies are modified, and the bot must provide current data (such as the length of the parental leave, which can have changed from last year).

3.4 Training on Conflicting or Unreliable Data

LLMs consume material from various sources, even inconsistent or erroneous ones. If an LLM encounters contradicting claims about an issue, it may internally "average" them or choose one at random to respond. [6] In broad fields, such behavior results in sporadic errors. In an internal chatbot situation, the model may hallucinate by combining contradicting information if it was trained with certain corporate data that has inconsistencies (or if it blends company data with general knowledge). Conflicting definitions of a benefit program, for instance, can lead the model to generate a confused or inaccurate response.

3.5 Imperfect Fine-Tuning or Alignment:

While it can help to reduce some problems, fine-tuning an LLM using instruction-following data—akin to OpenAI's RLHF process—won't completely prevent hallucinations. Studies have discovered that, in fact, fine-tuning sometimes causes hallucinations when the model attempts to incorporate new data inconsistently, even if it brings knowledge beyond what the model knew. [7] Furthermore, alignment tweaking usually gives the useful model top priority, not rejecting responses. As a side effect, the model might be inclined to answer every query—even when it lacks the knowledge—rather than respond with “I don’t know.” This tendency to always produce an answer can amplify hallucinations in an HR bot unless explicitly controlled.

4. Taxonomies of Hallucination

Several taxonomies have been proposed to categorize the diverse manifestations of LLM hallucinations. A prominent framework, particularly relevant for open-ended LLMs, distinguishes between factuality and faithfulness. [3] Figure 2 shows graphically the different taxonomies that are discussed.

4.1 Factuality Hallucination: The generated content conflicts with verifiable real-world facts. [1] This includes:

- Factual Contradiction: Generating incorrect information about verifiable entities or their relationships (e.g., stating the wrong inventor for a device). [1]
- Factual Fabrication: Generating information that is unverifiable or non-existent in the real world (e.g., inventing a species, making unsubstantiated broad claims). [1]

4.2 Faithfulness Hallucination: The generated content deviates from the user's input or instructions, or lacks internal consistency. This includes:

- Instruction Inconsistency: Failing to follow the user's explicit directive (e.g., answering instead of translating). [1]
- Context Inconsistency: Contradicting information provided in the prompt or source context (e.g., generating details not present in or conflicting with an input document). [1]
- Logical Inconsistency: Exhibiting flaws in reasoning or internal contradictions within the generated output (e.g., making calculation errors in a step-by-step solution). [1]

Other relevant classifications exist, often overlapping with the above:

- Intrinsic vs. Extrinsic: Intrinsic hallucinations contradict the source content or conversational history, while extrinsic hallucinations introduce new, unverifiable information. [1] Intrinsic relates closely to context inconsistency, while extrinsic relates to factual fabrication.
- Input-Conflicting: Responses diverging from user input [2], like instruction or context inconsistency.
- Alternative Categories: Some researchers propose categories like Factual Incorrectness, Misinterpretation (Corpus or Prompt), and Needle in a Haystack (difficulty retrieving specific facts). [8]

5. Mitigation Strategies for Hallucinations

Developers and academics have proposed numerous strategies to reduce hallucinations in LLMs. Mitigating techniques in the framework of an internal HR chatbot must guarantee that the bot's responses are based on accurate, company-approved knowledge, or else the bot knows when to refrain. Below we go over some important strategies and assess their relevance for business HR situations.

5.1 Retrieval-Augmented Generation (RAG)

Retrieval-Augmented Generation (RAG) is one of the most often used and successful methods to fight hallucinations. [19] Under a RAG system, the LLM is expected to base its response on relevant material obtained from a trustworthy knowledge source (such as a corporate HR policy database or document repository) rather than rely just on its internal memory. [6] Grounding generation on

genuine documents helps reduce the model's tendency to create false information. RAG essentially turns an open-ended generating job into an open-book question-answering activity where the book serves as the knowledge base for the model. Figure 1 shows the RAG architecture.

Effectiveness: RAG has proved to drastically lower hallucinations in certain environments. Shuster et al. (2021) showed that "retrieval augmentation reduces hallucination in conversation"—their "conversational agent created fewer incorrect assertions when it could get information on demand. [9, 21] The developers of the SAP HR chatbot project observed in the enterprise area that the RAG method was "ideal for this use case as it allows the model to produce more grounded answers, hence reducing hallucinations." Providing the real corporate wiki or policy text helped to reduce the model's requirement for guessing. [9] Empirically, the resulting answer is typically accurate (barring the model misreading the reference) if the retrieval component identifies the proper reference for a question.

Challenges: RAG is not perfect even if it is a powerful method. Its success depends on the retrieval part locating the pertinent and accurate data. If the retrieval fails—that is, if no relevant document is discovered or the top papers are unrelated—the LLM may still hallucinate unless it is intended to refrain. [10] RAG systems are known to generate mistakes because the model attempts to "fill in gaps" between retrieved text segments. Another problem is that several documents may need to be synthesized, or files may include information in indirect forms. A naive RAG that merely dumps material might not be able to completely answer difficult questions. According to the Microsoft study on domain-specific QA, the model's effort to answer could still be erroneous because retrieved chunks typically provided only incomplete information for difficult multi-hop questions. [10] This finding suggests that RAG reduces hallucinations, but it does not guarantee accurate policy interpretation beyond its stated boundaries.

5.2. Confidence Estimation and Answer Abstention

Having the system recognize its doubt and refrain from responding when confidence is low helps prevent hallucinations. The concept is straightforward: rather than risk a confident but incorrect response, the AI should declare "I don't know" or escalate to a human if it isn't sure it's correct. This approach calls for some kind of gauge of the model's confidence or likelihood of hallucinations.

A. Techniques for Confidence Measurement:
Several approaches have been explored:

5.2.1 Model logits and probabilities

Examining the probability distribution over potential outputs as the model responds is one of the methods of measuring confidence. If the model gives an extremely high probability to a particular word or phrase, thereby producing a highly peaked distribution, the model may seem more "confident" than when probabilities are more equally distributed. These raw probabilities, meanwhile, do not consistently point to factual correctness. Language models often give claims that seem reasonable but are factually untrue with great probability. This is so regardless of truthfulness, as, often based on patterns they have regularly observed throughout training, they make their predictions. Consequently, an LLM like GPT may boldly create misleading information just because it has been trained on such phrasing often. Thus, when trying to identify hallucinations, relying solely on the probability estimates of a model may lead to errors. [11]

5.2.2 Self-consistency / Multi-sampling

Generating several answers (or other pathways of reasoning) for the same question and observing their consistency is a more solid approach. Should the model produce the same response (or very similar replies) over many samples, such behavior indicates that the probability mass of the model is concentrated, and it "believes" that answer strongly, which might correspond with correctness should the model know the truth. Variations in the replies suggest doubt. Entropy-based techniques and the SelfCheckGPT methodology have their roots in this. Ask the model five times, "What is the maximum carryover PTO days?" Clearly, if it produces five distinct numbers in every attempt, it does not really know; any one of those is probably a guess. If it says "10 days" frequently, chances are higher that "10 days" is either accurate or at least the model strongly believes it to be. Practically, Farquhar et al. (2024) discovered they could efficiently identify a subset of hallucinations (those resulting from uncertainty) by grouping several outputs and evaluating semantic entropy. [12] The drawback is that the approach entails additional computation—running the model several times—and may still overlook scenarios in which the model is regularly wrong—that is, continuously hallucinating the same incorrect answer.

5.2.3 Calibration via a separate classifier

Another approach is teaching a lightweight classifier to ascertain whether a model-generated response is accurate. This classifier can use various internal signals, output embeddings, or hidden states of the LLM. One can, for instance, adjust a model using a dataset of question-answer (QA) pairs tagged as either factually correct or hallucinations. This approach forecasts response correctness by using logistic regression on the final hidden embeddings of the model. This method struggled to generalize to new or unseen topics even when it exhibited favorable results when evaluated on data close to its training distribution—that is, it could separate true from false when examples were familiar. [12] This finding exposes a more general restriction: classifiers evaluate new claims poorly without access to external grounding knowledge, although they can learn to identify known facts.

5.2.4 LLM self-evaluation prompts

One further approach is to ask the model itself to consider the response. For example, one may add, "Is the above answer based on factual company policy?" once the model responds. A true or false response is required. The model then performs essentially a second-pass assessment. This is like using a verifier to check the model's response. This type of verification is the P(True) technique, according to one research study (Varshney et al., 2023), in which the model is questioned if it is true or false once the response is given to it. [12] Under few-shot prompting, the model may occasionally recognize its own erroneous responses. The model can say, for example, "Actually, I am not totally sure." However, if the model remains confident in its hallucination, it may incorrectly validate it as accurate. Promising research uses models such as GPT-4 as a critic or checker for outputs of smaller models, which can often improve accuracy.

B. Abstention Mechanisms:

Once a system has some awareness of uncertainty, it can decide to refrain—that is, to fail to respond—when confidence falls short of a threshold. Conformal Prediction-based Abstention (Abbasi-Yadkori et al., 2024) is a recently advanced method that sets a threshold on a confidence metric such that the probability of a mistake falls below a target level with statistical guarantees. [13] They calibrate a threshold so that, say, with 95% probability, the answer is true whenever the model does not abstain using self-consistency—that is, by comparing several sampled replies. [13] This approach lets the model state, "I don't know," in a principled manner rapidly. Their tests revealed that, despite avoiding too conservative behavior, "conformal abstention" consistently limited the hallucination (error) rate on open-domain QA datasets. [13] Stated differently, it did not reject too often when it knew the solution; rather, it did reject in many circumstances where an unmitigated model would hallucinate.

An abstention for an HR chatbot might be an answer like, "I'm sorry, I'm not convinced I have the correct information on that. Let me point you to HR. Alternatively, "I don't have that information right now." This approach is better than answering boldly and incorrectly. Still, creating the user experience around refusals is delicate. If the bot turns away too often or without providing a road forward—such as linking to a human or reinterpreting the question—users may become annoyed. Confidence estimates are thus commonly combined with other techniques (such as retrieval). The HR chatbot should ideally only refrain when it can't really come up with a grounded response.

Efficiency: Correct tuning of confidence-based abstention can greatly reduce the frequency of hallucinations of answers. Conformal techniques, for instance, can set a maximum hallucination rate of 1% at the expense of, say, refraining from 10% of searches (hypothetically); these figures can be changed depending on risk tolerance. In key fields (medical, legal), high abstention is permissible; in HR, one might allow a little more risk if the questions are primarily low stakes, but for significant searches, it might be best to abstain. The important factor is that a well-calibrated system understands when it doesn't know, unlike an unmanaged LLM, which will answer anything to any prompt (including gibberish). For dependability, this is a major advance.

5.3 Prompt Engineering and Instruction Design

Prompt engineering is designing the input instructions and format to lead the LLM away from hallucinations. The presentation of the challenge can greatly affect the behavior of generative models. [21] Given that corporate chatbots rely solely on a thorough design of the conversation and system instructions, prompt engineering often serves as the primary defense, as it eliminates the need for model retraining. Figure 1 shows how we can use RAG with prompt engineering.

Effective prompt strategies to reduce hallucination include

- Explicit Guidelines Do not guess. Clearly tell the model, in the system or prompt, that if it is unsure or the information is inaccessible, it should react with a fallback—that is, indicate uncertainty or ask for clarification—instead of fabricating anything. For example, don't answer if you're unsure or if it doesn't fit the context.

Say instead you cannot locate the material. Repeating this activity helps us match the inclination of the model toward securely rejecting rather than hallucinating. The procedure works to some degree, notably for models who have been taught to obey directions; they may comply and provide more frequent "I'm not sure." Still, certain models—especially older GPT-3 variants—did not produce anything reasonable until their RLHF training included rejection. More recent ChatGPT/GPT-4 models are better at following such directions.

- Few-Shot Models of Factual Responses: Including a few demos in the Q&A prompt where the response fits given facts or states "I don't know" when something isn't in the text helps the model to replicate that behavior. Showing an example, for instance, helps: The assistant responds, "This refers to a policy quote," when the user asks, "What is our company's policy on X?" Our policy states... and even another illustration: Assistant: "I'm sorry, I cannot find information on policy Y." User: "What's the policy on Y?" This feedback loop forms the standard for the approach. The P(True) self-check approach also made use of few-shot prompting, so it guided the model in determining truth. [12] Regarding direct QA, this type of practice can inspire integrity and caution.
- Controlled Style and Formatting: Ask the model to respond using specific forms to help prevent free-form hallucinations. Tell the model, for example, to always quote the source or incorporate a reference from the knowledge base in her response. If the model fails to identify a reference, it may reduce its likelihood of expressing a fact. Always respond by citing the relevant section of the staff handbook for a real-world example. If you are unable to say, "I have to check HR." It makes use of the learned behavior of the model to conform with the format; it understands it should include a citation, so it might follow the latter's advice if it lacks one.
- Limit Open-Endedness: The question might restrict the response range. The model can drift if we let it produce very long responses or mix several subjects. Keeping searches and answers targeted helps reduce hallucinations. One suggested habit is dissecting difficult questions. This veers into the realm of chain-of-thought or agent approaches, where the model might first be prompted to create a plan or search query, etc., rather than directly answering. Instead of letting the model handle a multifarious question in one go, the system could prompt it step by step.
- Persona and Tone to Indicate Uncertainty: It can be important how the assistant's "persona" is formed from the prompt. If you direct the assistant to be "an accurate and cautious HR assistant" rather than an overly eager people-pleaser, accuracy may take precedence. For example, "You assist HR. You could find the corporate policies here. You should be truthful and acknowledge when your knowledge is insufficient. This helps offset any acquired behavior from general RLHF that would force the model to always answer.

5.3.1 **Limitations**: Prompt engineering by itself cannot address hallucinations. Even if the model lacks information and is motivated to be useful, it may still hallucinate within the prompt's limits. For example, it may start with "I am not sure, but..." and then still offer an estimate (some models do this unless explicitly restricted). Furthermore, too much prompting can make the model useless or too refutative, therefore affecting usability. Good design strikes a careful mix between caution and utility.

5.3.2 **Scalability**: The excellent thing about prompt-based solutions is they scale readily; you simply update the prompt for every query instead of requiring fresh model training. It's also domain-agnostic: the approach can be applied across different domains without needing customization for specific company data. That said, one must validate prompts carefully and maybe change them as the model develops (what works on GPT-4 might not work on a different model in the same way). Maintaining such a system usually depends on constant, rapid testing and improvement.

To summarize, prompt engineering is an inexpensive, rapid fix for hallucinations. It performs best when combined with other techniques: for RAG, a prompt might specifically instruct the model to only use the obtained text. This approach greatly reduced hallucination rates taken together from a zero-shot, no-retrieval baseline. By simply restricting style and encouraging brevity can reduce irrelevant or unrelated content, essentially stopping the model from meandering off-topic, which is where hallucinations usually start.

5.4 Fine-Tuning and Domain-Specific Adaptation

Fine-tuning is additional training of the base LLM either with goals promoting factual truth or on domain-specific data. Within the corporate HR setting, fine-tuning can have several uses: This process involves aligning the model's style and fallback behavior, such as learning to say "I don't know" correctly and educating the model about the company's knowledge to prevent it from experiencing hallucinations.

5.4.1 What is Domain Fine-Tuning on HR Data:
One simple method is to fine-tune the LLM on a corpus of internal HR documents, Q&A pairs, and policy texts, allowing the model to efficiently absorb ground truth data into its weights. [20] Then, rather than guessing, it can recall a policy detail accurately without needing to retrieve it. If we focus on the employee handbook, for example, the model might absorb important information (e.g., the number of vacation days and policies for certain requests) and be more accurate in its responses. One can also use fine-tuning to incorporate pairs of sample questions and right responses. The model is less likely to hallucinate something different if it has seen during training that the question "What is the dress code?" maps to "Our dress code is business casual, according to policy section X." Yang et al. (2023) took a fresh strategy whereby they used a smaller LLM fine-tuned on domain documents as an intermediary knowledge source [14, 18]. By optimizing LLaMA on their domain (cloud support) data, they developed a domain-specific model capable of producing pertinent knowledge upon demand, hence substituting a generative retrieval for a conventional one. [14] From an enterprise standpoint, one may picture training a modest-sized model using the entire HR policy archive. At query time, we ask the model to generate a targeted summary or pertinent fact piece, which the bigger model—like GPT-4—then uses to respond. Though benefiting from the excellent model's fluency and logic, this is a sort of knowledge distillation into a smaller model that is simpler to update than a large model. Since the smaller model was essentially a trained retriever that was effective at bridging gaps, the scientists found this paradigm improved performance on domain-specific QA and could help minimize problems, including missing context. [14]

5.4.2 Effectiveness:
When the inquiry belongs in the domain of the fine-tuned knowledge, fine-tuning can significantly reduce hallucinations. It is essentially addressing the root cause by bridging the knowledge gap. A refined internal model, for example, would never dream of the number of vacation days; it knows exactly from training. However, fine-tuning is not a foolproof solution for all hallucinations.

- No model can possess unlimited knowledge. There will be questions outside that data (or that combine internal and external knowledge) even after perfecting HR data. The model might yet hallucinate on those.
- Sometimes models overfit or overreach. Should fine-tuning not be done with extreme care, the model may become less fluent or overly eager to use specific bits of knowledge even when not relevant. Additionally, fine-tuning on a narrow corpus like the regulations of one corporation runs the danger of the model losing part of its general language skills or absorbing prejudices from that data.
- For most companies, fine-tuning big models like GPT-4 is not practical (closed and somewhat costly). Smaller organizations may not have the resources to modify a 70B model, even though open-source models like LLaMA-2 can be modified. They might rely on retrieval with a larger model or use a smaller model refined with some quality trade-off.

5.4.3 Scalability:
From the standpoint of scalability and maintenance, fine-tuning adds a load: you would have to update the model with every policy change. Retrieving is thus usually preferred since it separates knowledge updates from model parameters. One might combine retrieval for specific fresh information or details with fine-tuning to have the model broadly familiar with the topic.
On a technical QA dataset, a carefully customized method (with their interaction paradigm) outperformed a normal LLM + retrieval pipeline, according to the work "Empower LLMs to perform better on industrial QA." [14] They let a fine-tuned model provide the required data, thus addressing various limitations of retrieval, such as missing context. The result suggests that in certain scenarios, intelligent fine-tuning can either match or exceed retrieval. A finely adjusted model might be more coherent and integrated in responses for an HR chatbot (less of a copy-paste feel than a pure retrieval answer and possibly able to handle when info is implicit or needs merging).

5.4.4 Cost and Feasibility:
Data privacy (you might not want to submit your internal data to a third party to fine-tune their model) and cost make perfecting a big model impractical for business HR. Still, there are new

ideas that might be used, including smaller, fine-tuned models or on-site LLMs. Certain businesses refine open models based on data and then apply them behind their firewall.

In general, fine-tuning can help the model be more domain-aware and cautious, hence lowering hallucinations. It helps especially to guarantee that the model has fewer knowledge gaps. One must manage fresh questions and keep the model intact as knowledge evolves. While fine-tuning by itself can help to lower hallucination frequency, integrating it with retrieval guarantees the model can always obtain current data and cross-check itself.

5.5 Output Verification and Fact-Checking Pipelines

Using a post-processing stage to validate and fix the output of the model is a key step in ensuring accuracy. The idea is to add a second mechanism, like another model or algorithm, to check the result's accuracy and fix or flag issues instead of relying solely on the model to "get it right" the first time.

5.5.1 Methodologies for Output Verification and Fact-Checking Pipelines

A. Fact-Checking with External Knowledge:

One method is to use the response of the model and then confirm each claim by means of a search—internal or even web search, if relevant. If the chatbot responds, for instance, "Our company offers 12 paid holidays per year," the verification module would search the HR policy documentation or database to validate that figure. Should it discover that the policy indeed specifies ten holidays, it identifies an inconsistency. This method seeks supporting data and essentially treats the model's response as a hypothesis. It is like how a human might fact-check a claim by searching it up. Natural Language Inference (NLI) is another approach in which one treats the answer of the model as a hypothesis and the obtained context as premises and uses an NLI model to observe if the hypothesis is necessitated by the premises, contradicted, or unrelated. [12] If the information is contradictory or irrelevant, it may indicate a possible hallucination, as the answer isn't fully supported by facts. Some studies using DeBERTa models looked for consistency between responses and sources. [12, 26] This feature could be integrated such that the system either rejects the response or attempts another method if none of the known sources entail the answer.

B. Human in the Loop:

Under high stakes, the best backup is to have a human check the AI's responses before they are published (at least for some searches). [22] This clearly does not scale to all searches, but maybe for very sensitive or complicated ones (such as inquiries regarding legal compliance, harassment concerns, terminations, etc.), the algorithm may highlight that and call for an HR specialist to review. To assess results, the SAP chatbot project did, in fact, include a human in the loop during development and testing stages. [9] One may envision a situation in live deployment in which the bot responds initially, and should it fall into a risky area, an HR team member rapidly sanity-checks it.

C. Multi-agent or Iterative Checking:

We might also pit one model against another. Have a second instance of the LLM or another LLM criticize the response of the first LLM, for example? This process is like a chain of thinking in which the model is expected to consider accuracy: "The assistant answered X. Examine the query against corporate policy and note any mistakes. The second pass might catch errors that the first one missed. One develops and another verifies using the concept of an "adversarial assistant and checker." Using OpenAI's "Toolformer," or others, a model can contact Outside tools, such as a calculator or search engine, can be used while producing an answer, enabling instant fact-checking on demand.

5.5.2 Performance

At the expense of some recall or latency, fact-checking pipelines can significantly raise accuracy (correctness). By double-checking everything, a system might get near 99% factual accuracy; however, these improvements might result in a few seconds delay in response time and an increased frequency of abstentions or requests for clarification. For an internal tool used by an organization, some latency is usually acceptable; for HR responses, accuracy comes first over speed. One clear outcome: Using a checker model could help a lot of hallucinated responses from a conversation system be detected in research by Liu et al. (2022) on factual correctness in dialogue; hence, user trust considerably increased when responses only allowed factual ones. [15, 16]

6. Future Work and Research Direction

Understanding and reducing LLM hallucinations has advanced significantly, but there are still major potential and limitations, particularly when it comes to workplace and HR applications. We list several unresolved issues and potential study avenues below:

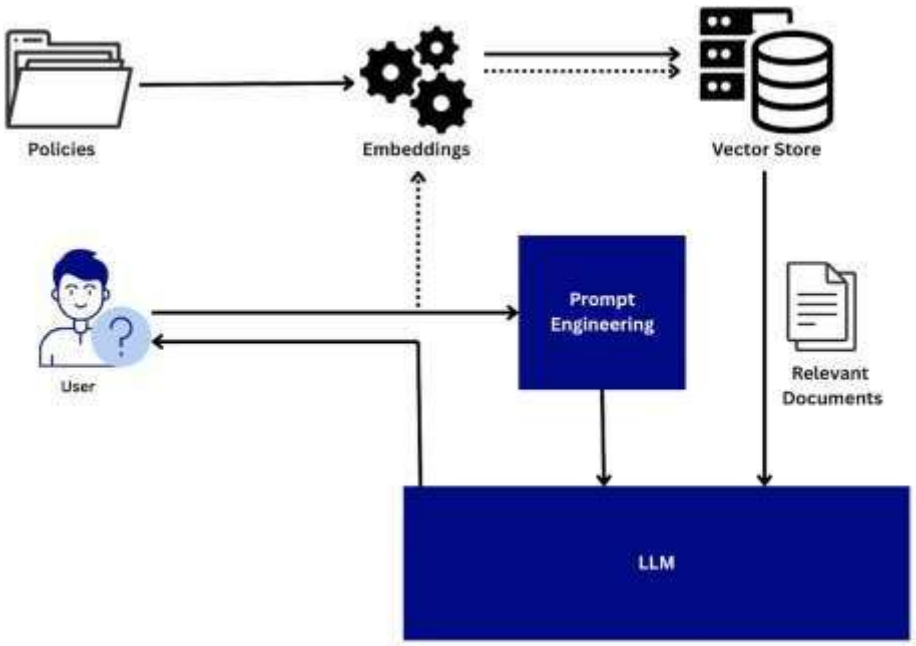


Figure 1. *Rag Architecture with Prompt Engineering*

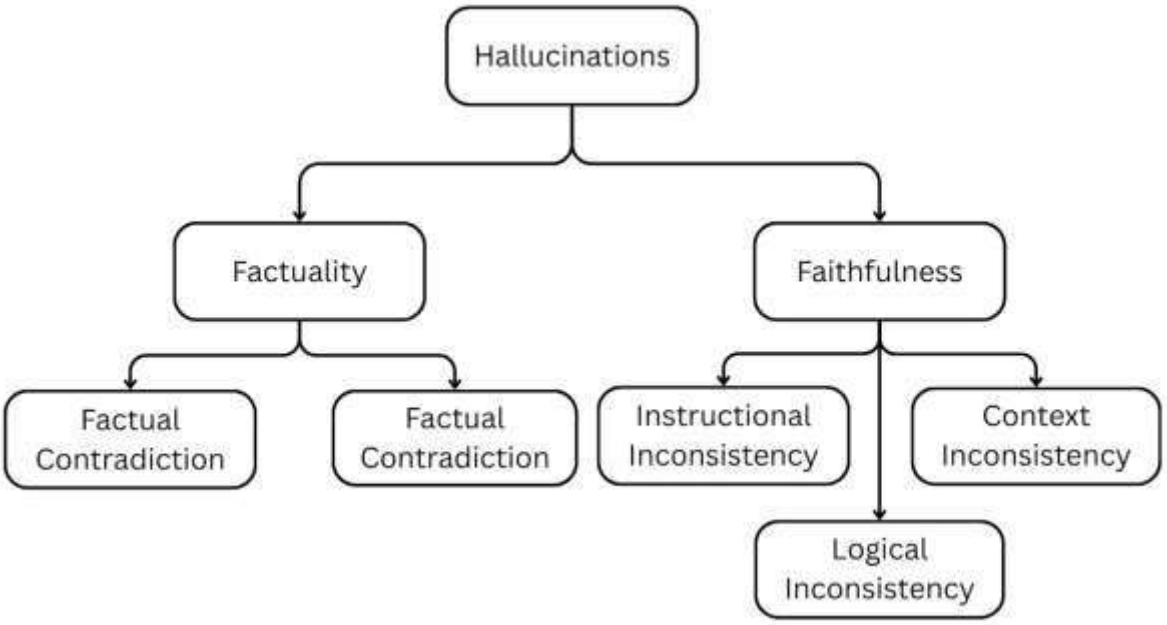


Figure 2. *Taxonomies of Hallucinations*

6.1 Domain-Specific Hallucination Benchmarks:

Evaluation frameworks and benchmark datasets specific to internal enterprise scenarios (such as HR) are required. Many benchmarks (HaluEval [24], Truthfulqa [25], etc.) focus on general or publicly available information. [1] Research in this area would be fueled by establishing a standard of realistic HR queries with ground-truth responses from corporate policy and tracking how frequently models hallucinate. The compilation of an FAQ dataset by Afzal et al. (SAP HR chatbot) is a beneficial place to start, but improvements may be sparked by more publicly available data—possibly synthetic but realistic HR circumstances. [9] Since not all mistakes are created equal, these

benchmarks should also assess consequentiality, or errors that would truly matter in an HR context rather than merely factual errors.

6.2 Better Evaluation Metrics for Factuality:

New measurements that automatically measure hallucinations in a response might also be helpful. Factual correctness cannot be captured by metrics such as BLEU or ROUGE. [9, 27,28] Factual consistency scores that match human perceptions of reality could be the subject of future research. While some approaches use Natural Language Inference (NLI), others employ question-answering techniques, which involve posing questions to the model about its responses and verifying the answers. It would be beneficial to create a trustworthy metric to enter training (for RL or for model selection). For instance, the model would be encouraged to reduce fake content if it had a "hallucination penalty" score to optimize against.

6.3 Knowledge Boundary Detection:

Huang et al. (2023) point out that one unanswered question is how to inform models about their knowledge gaps. Future research might concentrate on training techniques or architectures that give the model a clear representation of its knowledge and limitations. [1] Internally, this approach could include a two-step procedure: first, the model determines if it knows the answer with confidence; if not, it initiates a different response (such as retrieval or denial). To achieve this, we now use multiple sampling methods or heuristics. A unique token or latent that indicates "I know this" rather than "I'm guessing" may be used to train a model. This issue has to do with calibrating. One suggestion is to make models signal uncertainty when they cannot provide an explanation or supporting data for their responses. This approach would operate as an inherent truth check and be consistent with the human practice of explaining why a response is accurate.

6.4 Reducing High-Confidence Hallucinations:

One particularly problematic condition is the CHOKE phenomenon, which refers to certain high-confidence hallucinations. [11] Future studies should examine the reasons for models' occasionally high confidence in incorrect information, such as the frequency of training data or some misleading associations that are memorized. Knowing such details could aid in improving training data (e.g., by locating and eliminating examples that are inconsistent or misleading) or creating decoders that more effectively account for uncertainty. The likelihood that all models will incorrectly agree on a falsity may be decreased by using strategies like an ensemble of models or consensus among various model designs.

6.5 Advanced Retrieval and Fusion Techniques:

More effort is required on the retrieval side in situations where the answers are not contained in a single document. An ongoing field is multi-hop retrieval, which can collect fragments from several sources and allow the model to assemble them without experiencing hallucinations. Dispersed HR policies may include eligibility in one document and benefit specifics in another. "I found X here and Y there, so combining them, the answer is Z" is what we need models that can state. To enhance retrieval for complex queries, some advancements have been made in question decomposition (dividing a question into sub-questions) and inverted index matching of questions (QuIM-RAG [29]). It can be difficult to apply these to internal corpora while maintaining the faithfulness of the final response because it is simple for a model to draw a logical conclusion that isn't explicitly stated in the text. To accurately link facts, future systems may incorporate a more symbolic reasoning component (for instance, employing a tiny logical engine to check relationships described in some ontology to confirm that a stepsibling matches the concept of immediate family).

6.6 Robustness Against Prompt Attacks and Misuse:

A related failure mode of hallucination occurs when users purposefully or inadvertently cause nonsense or circumvent security measures (also known as prompt injection or jailbreak prompts). An internal user may attempt to elicit speculation from the bot by asking, "If the policy isn't clear, what do you think it might be?" We must make sure the bot remains steadfast and avoids speculation. It is crucial for security that future studies focus on aligning models to remain within their knowledge bounds even when confronted with hostile cues. A cunning user may attempt to socially engineer the bot into giving a response that contains estimates about sensitive information in a business context that the bot shouldn't even know or reveal. Preventing hallucinations also guards against unintentional information leaks.

6. Conclusion

In high-stakes business sectors like human resources, LLM hallucinations provide a major obstacle for the deployment of large language models for practical uses. We started by characterizing hallucinations as cases in which an LLM generates erroneous yet fluent information, and we investigated why this happens. In internal HR chatbot scenarios, hallucinations might manifest as false policies, erroneous advice, or other false information compromising the system's dependability and, hence, creating real organizational hazards. By means of this literature review, we examined current studies clarifying the nature of LLM hallucinations and developments in their management. Important results are:

- LLMs are prone to hallucinating in certain domains since they lack specialized domain knowledge (e.g., proprietary HR regulations), thereby highlighting the need to add enterprise data to models.
- The best mitigating results come from combining methods. One especially effective approach is retrieval-augmented generation, which grounds model outputs on factual records and greatly reduces hallucinations. Concurrently, abstention and confidence estimates offer a safety net by identifying ambiguity and thereby avoiding confident mistakes.
- By teaching models to avoid unsupported answers and guiding their responses to contain verifiable evidence, prompt engineering and careful system design can help further reduce hallucination rates. Still, prompts are not perfect and must be combined with other strategies.
- By means of domain data, fine-tuning can equip the model with authoritative information, hence lessening its predisposition to generate responses, and specialized fine-tuning techniques—e.g., smaller domain expert models feeding a bigger model—show promise for corporate deployment.
- Whether by retrieval + NLI checks or multi-step LLM reasoning, automated verification employing external knowledge can catch many hallucinations that pass through, therefore guaranteeing a high degree of factual accuracy in the result.

For an engineer creating an internal HR chatbot, the consequences are obvious: no one metric will ensure veracity; instead, a layered approach will greatly reduce hallucinations. One can build a system that employees can trust for correct information by including retrieval of up-to-date HR information, ensuring the model adheres to established guidelines, and adding procedures to check or reject doubtful answers. Human supervision on selected cases and ongoing model refinement via feedback help control the small residual error rate.

The research community is actively closing the still existing gaps. Future work on knowledge border detection, better standards, and dynamic learning will help LLMs to get even more dependable. Techniques that let models learn from corrections—each time a delusion is found and corrected—by a human or a tool—should be especially important since they will help algorithms or systems to avoid repeating that mistake. In each domain, this repeated learning loop can bring the hallucination rate toward zero in ever closer proximity.

Large language models are ultimately great tools for HR and business assistance, but their effective adoption depends on their factual accuracy being the priority. If uncontrolled, hallucinations can undermine confidence and cause actual damage; nevertheless, with the techniques covered—from RAG to thorough post-processing— we have a strong arsenal to control hallucinations. The rapid advancements in technology allow engineers to design HR chatbots that are not only grounded and dependable but also clever and engaging. Continuously staying updated with recent studies and validating systems will help practitioners apply LLM solutions that improve HR operations and protect against the dangers of artificial intelligence-generated incorrect information. Trustworthy artificial intelligence in businesses is being guided by exactly this mix of innovative research and intelligent engineering.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

References

[1] Huang, Lei, et al. (2025): A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems* 43(2);1-55.

[2] Berberette, Elijah, Jack Hutchins, and Amir Sadovnik. (2024) Redefining" Hallucination" in LLMs: Towards a psychology-informed framework for mitigating misinformation." *arXiv preprint arXiv*:2402.01769.

[3] Bang, Yejin, et al. (2025): HalluLens: LLM Hallucination Benchmark. *arXiv preprint* arXiv:2504.17550.

[4] Tonmoy, S. M., et al. (2024) A comprehensive survey of hallucination mitigation techniques in large language models.″ *arXiv preprint* arXiv:2401.01313 6 (2024).

[5] Xu, Ziwei, Sanjay Jain, and Mohan Kankanhalli. (2024) "Hallucination is inevitable: An innate limitation of large language models." *arXiv preprint* arXiv:2401.11817.

[6] Li, Johnny, et al. (2024) Banishing LLM hallucinations requires rethinking generalization." *arXiv preprint* arXiv:2406.17642.

[7] Huang, Lei, et al. (2025) A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions." *ACM Transactions on Information Systems* 43(2) 1-55.

[8] Banerjee, Sourav, Ayushi Agarwal, and Saloni Singla. (2024) Llms will always hallucinate, and we need to live with this. *arXiv preprint* arXiv:2409.05746.

[9] Afzal, Anum, et al. (2024) Towards Optimizing and Evaluating a Retrieval Augmented QA Chatbot using LLMs with Human in the Loop. *arXiv preprint* arXiv:2407.05925.

[10] Yang, Fangkai, et al. (2023) Empower large language model to perform better on industrial domain-specific question answering. *arXiv preprint* arXiv:2305.11541 (2023).

[11] Simhi, Adi, et al. (2025): Trust Me, I'm Wrong: High-Certainty Hallucinations in LLMs. *arXiv preprint* arXiv:2502.12964 (2025).

[12] Farquhar, Sebastian, et al. (2024) Detecting hallucinations in large language models using semantic entropy. *Nature* 630.8017;625-630.

[13] Yadkori, Yasin Abbasi, et al. (2024) Mitigating llm hallucinations via conformal abstention. *arXiv preprint* arXiv:2405.01563.

[14] Yang, Fangkai, et al. (2023) Empower large language model to perform better on industrial domain-specific question answering. *arXiv preprint* arXiv:2305.11541.

[15] Yang, Borui, et al. (2025) Hallucination Detection in Large Language Models with Metamorphic Relations. *arXiv preprint* arXiv:2502.15844.

[16] Liu, Tianyu, et al. (2021) A token-level reference-free hallucination detection benchmark for free-form text generation. *arXiv preprint* arXiv:2104.08704 .

[17] Shuster, Kurt, et al. (2021) Retrieval augmentation reduces hallucination in conversation. *arXiv preprint* arXiv:2104.07567.

[18] Yang, Chengrun, et al. (2023) Large language models as optimizers. *arXiv preprint* arXiv:2309.03409.

[19] Okafoeze, C. (2025). Analysing the potential solutions to LLM hallucinations in abstractive text summarisation.

[20] Chakraborty, N., Ornik, M., & Driggs-Campbell, K. (2025). Hallucination detection in foundation models for decision-making: A flexible definition and review of the state of the art. *ACM Computing Surveys.*

[21] Kulkarni, N., & Tupsakhare, P. (2024). Strategies for Avoiding GPT Hallucinations. *International.*

[22] Lin, Z., Guan, S., Zhang, W., Zhang, H., Li, Y., & Zhang, H. (2024). Towards trustworthy LLMs: a review on debiasing and dehallucinating in large language models. *Artificial Intelligence Review,* 57(9), 243.

[23] Talwar, S. (2025). Dynamic Just-In-Time App Servers with Automated Access Management on *AWS.*

[24] Li, J., Cheng, X., Zhao, W. X., Nie, J. Y., & Wen, J. R. (2023). Halueval: A large-scale hallucination evaluation benchmark for large language models. *arXiv preprint* arXiv:2305.11747.

[25] Lin, S., Hilton, J., & Evans, O. (2021). Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint* arXiv:2109.07958.

[26] He, P., Liu, X., Gao, J., & Chen, W. (2020). Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint* arXiv:2006.03654.

[27] Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002, July). Bleu: a method for automatic evaluation of machine translation. *In Proceedings of the 40th annual meeting of the Association for Computational Linguistics* (pp. 311-318).

[28] Lin, C. Y. (2004, July). Rouge: A package for automatic evaluation of summaries. *In Text summarization branches out* (pp. 74-81).

[29] Saha, B., Saha, U., & Malik, M. Z. (2024). Advancing Retrieval-Augmented Generation with Inverted Question Matching for Enhanced QA Performance. *IEEE Access.*