The HalluRAG Dataset: Detecting Closed-Domain Hallucinations in RAG Applications Using an LLM's Internal States

Fabian Ridder, Malte Schilling

Computer Science Department, University of Münster
Einsteinstraße 62, 48149 Münster, Germany
{fridder, malte.schilling}@uni-muenster.de

Abstract

Detecting hallucinations in large language models (LLMs) is critical for enhancing their reliability and trustworthiness. Most research focuses on hallucinations as deviations from information seen during training. However, the opaque nature of an LLM's parametric knowledge complicates the understanding of why generated texts appear ungrounded: The LLM might not have picked up the necessary knowledge from large and often inaccessible datasets, or the information might have been changed or contradicted during further training. Our focus is on hallucinations involving information not used in training, which we determine by using recency to ensure the information emerged after a cut-off date. This study investigates these hallucinations by detecting them at sentence level using different internal states of various LLMs. We present HalluRAG, a dataset designed to train classifiers on these hallucinations. Depending on the model and quantization, MLPs trained on HalluRAG detect hallucinations with test accuracies ranging up to 75%, with Mistral-7B-Instruct-v0.1 achieving the highest test accuracies. Our results show that IAVs detect hallucinations as effectively as CEVs and reveal that answerable and unanswerable prompts are encoded differently as separate classifiers for these categories improved accuracy. However, HalluRAG showed some limited generalizability, advocating for more diversity in datasets on hallucinations.

Code — https://github.com/F4biian/HalluRAG
Dataset — https://doi.org/10.17879/84958668505
arXiv — https://arxiv.org/abs/2412.17056

Introduction

Large language models (LLMs) are generative machine learning models that generate sequences of tokens one by one. As these are trained on massive datasets, LLMs have become excellent at producing coherent text. When sufficiently trained, these models can be prompted on novel tasks (Radford et al. 2019), meaning that one can apply an LLM on a different task—e.g., asking questions—to which the model generates a corresponding output and answer. One impressive feature of LLMs is that the generated answers are not only coherent as a possible text but also appear reasonable. While this allows LLMs as generative models to create novel outputs that were not part of the training data, there is a downside: The produced answers can be incorrect or not factual while being plausible-sounding answers. Another limitation of directly questioning LLMs is that they are restricted to knowledge learned from training data.

Retrieval-augmented generation (RAG) (Lewis et al. 2021) is a commonly used technique aimed at overcoming these challenges (Shuster et al. 2021). In RAG, background knowledge indexed beforehand is retrieved from a database as additional context and input to the model. For example, a user's question is enriched with particular snippets from external documents (e.g., PDF, PowerPoint, or text files) that serve as additional input to the LLM. RAG exploits the capability to use large contexts as inputs, integrating knowledge from the context into the answer. Typical examples of RAG systems are chatbots that provide customer support based on a database of past transactions (Xu et al. 2024) or tutoring systems in which context, as well as answers, are enhanced by infusing background knowledge from trusted sources (Levonian et al. 2023; Kahl et al. 2024). Although LLM applications incorporating RAG are aimed at hallucinating less, hallucinations still occur. Generating different forms of non-factual answers—or those contradicting some of the available knowledge—remains a significant challenge for incorporating LLMs into workflows and services.

One approach to address hallucinations is to detect them, which allows the system to intervene. The system could refuse to return the hallucinated response and regenerate another response. In this work, we analyze an approach that leverages the LLM's internal states to determine if it currently fabricates hallucinations while answering a user's question. The analysis is based on a novel dataset for detecting closed-domain hallucinations—i.e., a dataset of queries that cover knowledge not included in training datasets but for which additional context can be provided as input to the LLM. Furthermore, we start by introducing a distinction between different types of hallucinations.

Hallucination Types and Characteristics

A hallucination is a statement that lacks grounding in the LLM's knowledge. Specifically, a statement is considered ungrounded if it does not exist in any form within the model's knowledge—neither as part of the training data nor as additional context (such as provided in RAG application).

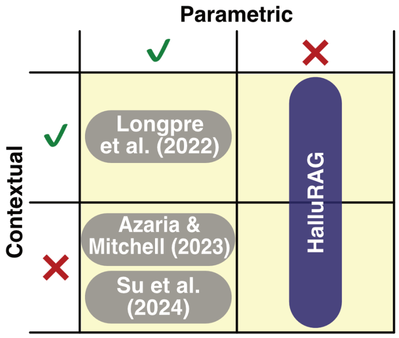

Figure 1: Differentiation of approaches (Azaria and Mitchell 2023; Su et al. 2024; Longpre et al. 2022) based on the type of queried knowledge. Current methods in the literature focus on knowledge that is assumed as entrained into the LLM's parameters (parametric). This is often difficult to assess as, first, the training data is not always accessible. Second, it is not clear if and how this information was accurately learned by the model during training. Or how further training on other data might have influenced the information, for example, when contradicting pieces of information were present in the training data. In contrast, our focus (in the HalluRAG dataset) is on knowledge the LLM could not have seen during training, avoiding speculative assumptions. This gives us full control over offering this knowledge as context to the model or dealing with questions the model can not answer in any case. The second dimension distinguishes if relevant information for answering the question was provided as part of the context.

This makes an ungrounded statement "fabricated" (Agrawal et al. 2024). Notably, the determination of whether a statement constitutes a hallucination does not depend on its factual accuracy or alignment with current scientific consensus as such but rather on how well the response aligns with the knowledge sources that constitute the model's foundation.

In an LLM, there are two forms of knowledge and memories: First, during training, knowledge is integrated into the model's parameters from the training data. This is known as **parametric** knowledge. Second, contextual information can be provided to the model as part of the input or prompt, referred to as **contextual** knowledge (Longpre et al. 2022). When considering hallucinations, we should distinguish between these two kind of knowledge sources: parametric and contextual. On the one hand, absolute or **open-domain hallucinations** are generated statements with no grounding in an LLM's parametric knowledge or training data. This makes open-domain hallucinations difficult to detect due to the vast and often opaque nature of training datasets, which may be unpublished or lack detailed references. On the other hand, **closed-domain hallucinations** involve statements ungrounded with respect to the provided context and the knowledge given as input in a prompt (Agrawal et al. 2024; Friel and Sanyal 2023). Additional distinctions for hallucinations have been introduced (Zhang et al. 2023) that address special cases (Longpre et al. 2022) and lead to further subdivisions (Maynez et al. 2020). From our point of view, it is important to explicitly distinguish how a hallucination fabricates new knowledge: Does it contradict or deviate from knowledge the model has seen during training and encoded in its parameters? This is the most widely assessed type of hallucination, where statements are actually false and contradict—or are assumed as contradicting—the training data (Fig. 1). Further research (see below on Related Work) has investigated hallucinations where LLMs fabricate statements that are false with respect to knowledge provided as additional input. Our dataset, HalluRAG, is designed to specifically address knowledge that the model has not seen during training, allowing systematic control over whether information is provided as context.

Related Work

There are two main characteristics or dimensions of **hallucination detection methods**: first, the level of access to the LLM, and second, the use of references for comparison during the detection process.

Detection methods require different levels of access to an LLM. At one end of such a spectrum, **black-box** methods utilize only the LLM's output token, focusing exclusively on the generated language. On the other end, **white-box** methods tap into the model's internal states for detection, requiring full access to the LLM, which is possible when the model is run locally. **Gray-box** detection methods are an intermediate case, employing the output probability distribution for tokens—as provided as well by many online LLM services—but without requiring full access to internal states (Manakul, Liusie, and Gales 2023).

The second characteristic differentiates whether references are available for comparison purposes. **Reference-based** (supervised) methods use a reference text to validate output accuracy, whereas **reference-free** (unsupervised or zero-reference) methods assess output without the need for any reference for comparison. This is particularly important, as in many cases, no reliable references are available (Fang et al. 2024). White-box and gray-box approaches are typically applied without references, as hallucination detection should rely on the internal states or the output distribution alone, without considering the generated text itself.

Hallucination detection is a rapidly growing field of research, with an increasing number of proposed methods in all mentioned areas. Reference-based black-box methods were proposed early on. For example, RefChecker (Hu et al. 2024) evaluates 'knowledge triplets' by comparing them to real references under various context conditions. Such methods have been further improved, e.g., Agrawal et al. used Bing to verify if suggested references by an LLM really exist. Reference-free black-box methods focus on the parametric knowledge of LLMs and prompt the LLM to either explicitly or implicitly evaluate generated answers by themselves: Kadavath et al. proposed such self-evaluation to measure confidence in the models' answers. In contrast, SelfCheckGPT (Manakul, Liusie, and Gales 2023) checks for self-consistency across multiple generated outputs for the same prompt. Manakul, Liusie, and Gales also suggested a reference-free gray-box method using likelihood and en-

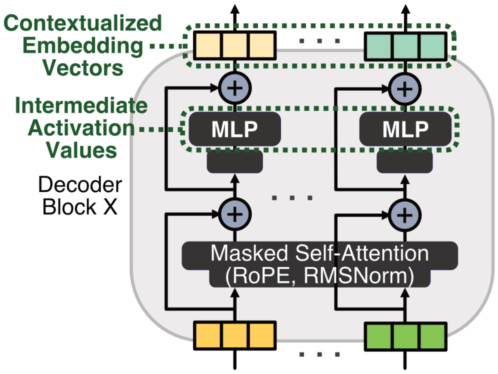
Contextualized Embedding Vectors
Intermediate Activation Values
Decoder Block X
MLP
MLP
Masked Self-Attention (RoPE, RMSNorm)

Figure 2: Locations of intermediate activation values and contextualized embedding vectors in the simplified architecture of LLaMA 2 7B including RMSNorm (Zhang and Sennrich 2019) and Rotary Position Embeddings (Su et al. 2023). While Azaria and Mitchell and Su et al. used contextualized embedding vectors as input to a binary classifier, we extend this approach by also considering intermediate activation values as classifier inputs.

tropy metrics that allow to flag hallucinations based on the measured response randomness.

Lastly, reference-free white-box methods involve analyzing the internal components of an LLM, such as the INSIDE EigenScore (Chen et al. 2024) for assessing self-consistency through covariance analysis, the MIND classifier (Su et al. 2024), and SAPLMA's template-based classifier (Azaria and Mitchell 2023). Closest to our approach is the work of Su et al. and Azaria and Mitchell, who follow a white-box, reference-free detection method. Both evaluate generated sentences solely by examining the model's internal states without requiring an external reference during production, but with full access to the LLM's internal structure. This is achieved by training a neural network that uses the internal states as an input and predicts the likelihood of a hallucination in the concurrently generated output sequences. While our approach uses a similar setup, it specifically focuses on distinguishing closed-domain hallucinations and, as such, focuses only on knowledge queries that are outside of the training data (Fig. 1). In contrast, Su et al. and Azaria and Mitchell assume that Wikipedia content is reliably encoded in an LLM's parametric knowledge, which seems reasonable for basic and foundational information, such as elements on the periodic table. However, this is not necessarily true for other types of knowledge, which might change over time or for which contradictory information has been encountered during training. Moreover, it is difficult to determine whether such knowledge—even if seen during training—has been accurately encoded in the LLM's parameters. Additionally, our proposed method takes into account intermediate activation values from the decoder blocks' multilayer perceptrons (MLPs), an internal state type that previous research has mostly neglected, as it primarily focused on contextualized embedding vectors (also referred to as activation values, see Fig. 2).

Method

We aim to train an MLP to identify sentence-level hallucinations by analyzing specific internal states in RAG applications. In particular, we focus on closed-domain hallucinations, which involve querying the model for answers that it cannot infer from its training. To train the classifier, we require a suitable supervised dataset that links the internal state—corresponding to a given input that lead to a generated sentence—to a hallucination label.

While we aim for reference-free hallucination detection in RAG applications, **question-answering datasets** are suitable for training these methods. There are several fitting datasets, e.g., the NoMIRACL dataset (Thakur et al. 2024) and the RAGTruth dataset (Wu et al. 2023). The latter offers $18,000$ word-level human-annotated hallucinations from LLM outputs across tasks such as summarization, question answering, and data-to-text, focusing on Mistral (Jiang et al. 2023) and LLaMA 2 models (Touvron et al. 2023). However, both datasets have some common drawbacks. First, the prompt formats are not diverse. Second, it is unclear whether information on questions and answers has been explicitly or potentially used during training of the LLM, as there are no criteria regarding recency. Recency—considered as novel information that has emerged during recent times and was not available before a given cut-off date—is a crucial factor that is not guaranteed by either RAGTruth or NoMIRACL. Therefore, we created a novel dataset called HalluRAG, as we consider recency a promising means of determining what information an LLM may have been trained on. This enables us to focus exclusively on closed-domain hallucinations, explicitly controlling the answerability of prompts (Fig. 1). Since Su et al. found that the SAPLMA classifier by Azaria and Mitchell might have been overfitted, possibly due to its simplistic template-based approach, we aim for HalluRAG to encompass a broader variety of formats and topics. Our objective is to create a heterogeneous dataset that allows training MLP-based classifiers, which can be broadly applied to any RAG application. As Su et al. found a distinction between a forced hallucination and one that emerged naturally, hallucinations in HalluRAG will not be enforced but should only emerge, leading to **natural** hallucinations.

In the following, we will, first, present the HalluRAG dataset, which includes RAG prompts, generated responses, and internal states, fulfilling our requirements. Second, we will explain the training of an MLP-based classifier that uses the internal states of an LLM and estimates whether a generated sentence is hallucinated.

Creating the HalluRAG Dataset

We selected Wikipedia as a semi-structured source of information because of its wide variety of topics and the large number of references within its articles. Furthermore, Wikipedia provides detailed timestamps indicating when information was created or updated. This is particularly important, as we use recency to ensure that information could not have been used for training. We used Wikipedia timestamps to verify that information is recent and was not available—even in an older form or relying on similar older content—during training of an LLM based on the LLM's cutoff date.

The process of generating HalluRAG is briefly illustrated in Figure 3 and explained in detail below.

1. Extracting recent sentences from Wikipedia: For a given cut-off date—in our case February 22, 2024—all newer articles from the English Wikipedia were considered. From all potential articles, sentences with at least one reference, that consisted of more than 50 characters, and were not linked to other Wikipedia articles were collected. Furthermore, we checked each date, access date, and archive date in the references to ensure they were not older than the cut-off date. If all of these conditions were met, a sentence was considered a dataset candidate.
2. Question generation: Given a dataset candidate and its surrounding text, we prompted GPT-4o (gpt-4o-2024-05-13) (OpenAI et al. 2024) to generate a question that could have been raised in a RAG application. GPT-4o was further instructed to extract the corresponding answer from the dataset candidate's sentence, which was necessary for our hallucination annotation in step 5. Additionally, we use the extracted answer as an effective verification step, ensuring it was a substring of the original sentence to remove the chance of hallucinations at this stage.
3. Creating pairs of questions and answers: For each question, we generated two RAG prompts—one **answerable** and one **unanswerable**—reflecting the real-world variability in retrieval systems, where the correct chunk may not always be provided or for cases when the question lacks a definitive answer. The first prompt included the relevant passage from Wikipedia where the answer was present, while the second contained an unrelated chunk from a different Wikipedia article. To add diversity, we randomly varied one of three prompt templates, a chunk size (350, 550, or 750 characters), and the number of chunks per prompt (1, 3, or 5).
4. Observing internal states: We passed these RAG prompts to an LLM to generate answers for each question, using a temperature of 0.0 and a token limit of 500. For HalluRAG, we used LLaMA-2-7B-Chat-HF (Touvron et al. 2023) and Mistral-7B-Instruct-v0.1 (Jiang et al. 2023).[1] During each generation, specific internal states (see next section) were extracted and stored for each sentence of the generated response: the contextualized embedding vectors from the last token's middle and last decoder blocks (noted as 'cev (middle)' and 'cev (last)') and the intermediate activation values from the same blocks (denoted 'iav (middle)' and 'iav (last)').[2] Previous work has suggested that the final token embeddings best capture the essence of the preceding text, effectively compressing its content into a single vector (Azaria and Mitchell 2023; Su et al. 2024).
5. Labeling of responses as hallucinations: To label each sentence as either hallucinated or not, we used GPT-4o (gpt-4o-2024-05-13) to compare each generated sentence against the entire response, the Wikipedia passage, and the quoted answer from step 2. GPT-4o was prompted with a detailed Chain-of-Thought (CoT) prompt (Wei et al. 2023) to set four booleans: `conflicting`, `grounded`, `has_factual_information`, and `no_clear_answer`. Each combination of these booleans was mapped to either 1 ("hallucinated"), 0 ("non-hallucinated"), or 'None' ("invalid") based on the prompt's answerability. Sentences labeled as 'None' were withheld from training, testing, and validation. This four-boolean framework showed distinct advantages, as it made GPT-4o's decision-making more transparent and forced it to break the task down. This helps mitigate GPT-4o-induced hallucinations, as the model justifies each boolean with verifiable substring checks. As a benchmark, we compared these with 274 human-labeled sentences. GPT-4o achieved an F1-score of 96.05% and an accuracy of 97.81% (the six found 'misclassifications' appeared as debatable cases to us). Overall, GPT-4o appears as a reliable evaluator for this task.

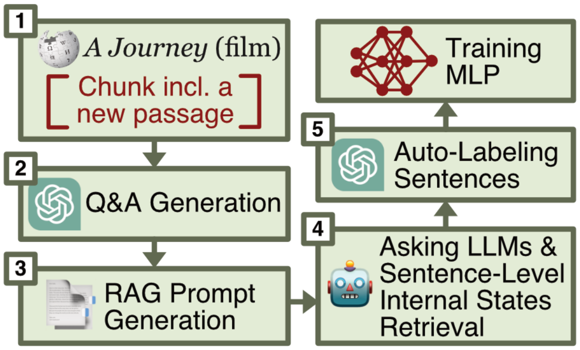

Figure 3: The whole process of a valid passage on Wikipedia turned into a RAG prompt and internal states for the HalluRAG dataset to eventually train a multilayer perceptron.

In total, HalluRAG comprises $19,731$ validly annotated sentences generated by LLaMA-2-7B, LLaMA-2-13B, or Mistral-7B, across different quantizations (float8, int8, or int4). As mentioned, we are only interested in natural hallucinations, not forced ones. Therefore, as a first observation, we report the hallucination rates for the different models: The LLaMA-2-7B configurations exhibit a stable hallucination rate of approximately 21% on HalluRAG, whereas Mistral-7B demonstrates a significantly lower hallucination rate of about 10%. When grouping by prompt template, we observed that the hallucination rate varied considerably. For instance, LLaMA-2-7B showed hallucination rates between 32% and 40% with a template from the Langchain hub, compared to around 16% with template 1 (designed by us). This indicates that prompt engineering is already a straightforward yet effective method for reducing hallucinations.

Training a Classifier for Hallucination Detection

We trained a neural network as a classifier on the HalluRAG dataset. The input to the network consists of different internal states from an LLM, obtained during the generation

[1] LLaMA-2-13B-Chat-HF (Touvron et al. 2023) was also observed, but showed some unexpected behavior.

[2] We thank Su et al. for providing their code on GitHub and for their assistance.

Internal State	**All (%)**	**None (%)**	**float8 (%)**	**int8 (%)**	**int4 (%)**
LLaMA 2 7B Chat HF					
cev (middle)	65.41±0.87	62.82±0.55	63.57±1.40	62.79±1.82	71.01±0.59
cev (last)	60.40±1.79	60.93±1.05	57.61±2.03	59.68±4.08	71.24±2.29
iav (middle)	65.98±1.66	64.62±1.43	61.40±3.59	62.40±2.03	71.46±0.67
iav (last)	64.93±2.34	62.60±1.67	58.57±2.52	62.55±1.38	71.07±0.94
Mistral 7B Instruct					
cev (middle)	67.47±1.16	*66.98±4.29*	*54.29±7.85*	*65.26±0.62*	*68.59±5.26*
cev (last)	73.28±3.51	*67.27±4.83*	*67.28±9.13*	*68.02±7.70*	*75.16±1.84*
iav (middle)	69.95±2.38	*56.33±1.66*	*61.35±2.75*	*67.31±0.36*	*71.94±5.97*
iav (last)	74.91±0.92	*65.84±3.62*	*70.58±5.03*	*71.43±3.00*	*78.94±1.88*

Table 1: Average test accuracies and standard deviations of training ten MLPs on the internal states of the responses taken from HalluRAG. The column names indicate the quantization, where 'None' means the use of no quantization and 'All' considers all quantizations together (None, float8, int8, and int4). All internal states have been extracted from the last token's middle and last decoder block or 'layer.' We abbreviate contextualized embedding vectors as 'CEV' and intermediate activation values as 'IAV'. The values based on a single quantized Mistral configuration are italic, since their test sets appear insufficiently small for being analyzed in more detail.

of our dataset. The target for the classifier is the assigned label, indicating whether the answer is hallucinated or non-hallucinated. The MLP structure consists of four linear layers (`input_size`—256—128—64—1) with ReLU activation and a final sigmoid function, following the structure used in MIND (Su et al. 2024) and SAPLMA (Azaria and Mitchell 2023). The learning rate was set to $2.5\mathrm{e}{-6}$, with a weight decay of $1\mathrm{e}{-5}$ and an initial dropout rate of 15% (similar to MIND's 20%). We employed early-stopping: training stopped after a maximum of 800 epochs or when the validation loss did not improve for 30 epochs. We proceeded with the checkpoint that had the lowest validation loss for testing. Ten MLPs were trained per LLM configuration to account for statistical variability. We analyzed the mean and standard deviation of test accuracies across all these models.

However, the number of generated sentences by the LLMs varied across our different configurations (chunk size, chunk count, template, and answerability), inferring an imbalance in the data. To address this, we oversampled based on the configuration parameters and the label (hallucinated vs. non-hallucinated). In this way, we ensured a balance of one-to-one with respect to answerability and hallucination across training, validation, and test sets. For chunk sizes, numbers of chunks per prompt, and prompt templates, we ensured a ratio of one-third in each case. This approach enables interpretable and comparable test accuracies.

Results

Classifier Results on HalluRAG

For each configuration (different LLMs, quantizations), ten classifiers were trained independently to determine whether a given input—provided as the internal states of the LLM—should be deemed as a hallucination. The neural networks were trained on balanced datasets, using oversampling as described above. An overview of the results is provided in table 1. For the LLaMA-2-7B-Chat-HF model, test accuracies are around or above 60%, with int4 quantizations achieving a consistently higher accuracy of 71%. Notably, some larger-than-expected standard deviations make comparing different layers and quantizations challenging, but emphasize the importance of multiple training runs.

When comparing different layers, the middle layers of both, contextualized embedding vectors (CEVs) and intermediate activation values (IAVs), generally show slightly higher average accuracies than the last layers.

In contrast, the Mistral-7B-Instruct-v0.1 model showed a different pattern compared to LLaMA-2-7B: the last layer's CEVs and IAVs lead to accuracies above 70%, while middle layers remained below 70%. Importantly, in nearly all experiments, MLPs trained on sentences generated by Mistral-7B outperformed those trained on LLaMA-2-7B data.[3]

In particular, regardless of the specific model, all int4 quantizations consistently demonstrate the highest test accuracies in Table 1, as well as in other results. Furthermore, in combining internal states by concatenating their vectors does not seem to boost test accuracies considerably.

We also tested the impact of withholding specific variations of the training data (e.g., a chunk size of 350 or prompt template 2) by excluding them from training and validation, but still evaluating on the full test set, including the previously withheld parameters. The results show no significant impact on test accuracy, except when training and validating solely on answerable questions, which on the test set lead to accuracies no better than random guessing.

Generalization in Hallucination Classifiers

We further cross-tested the MLPs trained on HalluRAG against RAGTruth (Wu et al. 2023), as a similar dataset that however lacks a clear answerability distinction, and vice versa. This generally showed a poor performance for the LLaMA-2-7B model, which performed close to chance level (Table 2). Mistral-7B again showed much better performance when compared to the LLaMA-2-7B model, but still performed worse than when trained on its specific dataset.

[3] Test accuracies for LLaMA-2-13B-Chat-HF yielded results at chance level of around 50%.

Internal State	H-H-R (%)	R-R-H (%)
LLaMA-2-7B-Chat-HF		
cev (middle)	51.04±0.30	52.84±1.34
cev (last)	49.58±0.51	50.24±0.59
iav (middle)	52.66±0.42	54.60±0.93
iav (last)	49.70±0.14	51.16±1.18
Mistral-7B-Instruct-v0.1		
cev (middle)	56.60±0.79	62.27±0.97
cev (last)	57.11±1.56	58.46±0.33
iav (middle)	55.08±1.70	64.01±0.23
iav (last)	55.88±0.25	58.65±0.02

Table 2: Testing for generalization on a different dataset: Average test accuracies and standard deviations of testing ten MLPs, which on the one side have been trained and validated on HalluRAG but tested on RAGTruth (H-H-R), and on the other side trained and validated on RAGTruth but tested on HalluRAG (R-R-H). All quantizations together were used. All internal states have been extracted from the last token's middle and last decoder block or 'layer.' We abbreviate contextualized embedding vectors as 'CEV' and intermediate activation values as 'IAV'.

This might indicate a general problem of overfitting in both HalluRAG- and RAGTruth-trained MLPs, with Mistral-7B demonstrating at least better-than-random test accuracies.

Training Separate Classifiers for Answerable and Unanswerable Questions

The HalluRAG dataset distinguishes answerable and unanswerable questions. We trained, validated and tested classifiers separately on both partitions of the balanced dataset. The results (Table 3) demonstrate a significant boost in test accuracies when using separate classifiers. For answerable questions, accuracies improved notably compared to the earlier results (Table 1). For example, LLaMA-2-7B now achieves test accuracies above 75%. However, as an exemption, for Mistral-7B, the accuracy of the last layer drops from around 74% to 70%. For unanswerable questions, the improvement is even more pronounced, with LLaMA-2-7B reaching over 80% accuracy. While training MLPs separately might simplify the problem, the classification remains non-trivial, particularly for answerable questions. For unanswerable questions, however, MLPs primarily need to determine from the internal states whether the LLM has responded with an "I don't know" (non-hallucinated) or something else (hallucinated).

For instance, the sentence *'I apologize, but I cannot provide an answer to your question as it is not based on any factual information provided in the context.'* by LLaMA-2-7B as a response to an unanswerable question is considered non-hallucinated. Same applies to its grounded response *'The statue of Queen Elizabeth II in Oakham was funded through public subscription by Rutland people.'* to an answerable question. Notably, Mistral-7B MLPs achieved near-perfect classification, with accuracies close to 100%. While this shifts the problem to detecting prompt answerability, it still offers valuable insights into LLM behavior.

Internal State	Answ. (%)	Unansw. (%)
LLaMA-2-7B-Chat-HF		
cev (middle)	75.21±0.45	82.89±0.43
cev (last)	77.64±0.63	87.13±0.40
iav (middle)	75.52±0.59	81.98±0.83
iav (last)	78.22±0.54	86.26±0.14
Mistral-7B-Instruct-v0.1		
cev (middle)	69.56±0.16	99.55±0.24
cev (last)	70.66±0.53	100.00±0.00
iav (middle)	68.55±2.08	99.87±0.04
iav (last)	70.38±0.33	100.00±0.00

Table 3: Average test accuracies and standard deviations of training ten MLPs on internal states of the responses based on either answerable or unanswerable questions of all LLM configurations (None, float8, int8, and int4). All internal states have been extracted from the last token's middle and last decoder block or 'layer.' We abbreviate contextualized embedding vectors as 'CEV' and intermediate activation values as 'IAV'.

Discussion and Conclusion

In this work, we introduced HalluRAG, a dataset designed to detect closed-domain hallucinations in RAG systems by employing recency. We are focussing on novel information that were added after a specified cut-off data, and could not have been used for training. HalluRAG leverages an auto-labeling process using GPT-4o to annotate hallucinated and non-hallucinated sentences, achieving an F1-score of 96.05%. Even though HalluRAG aimed for diverse topics and formats, our evaluation demonstrates that the size of HalluRAG and the usage of $1,080$ prompts per LLM configuration appears still too small and the missing diversity still limit the MLPs' generalizability. Furthermore, GPT-4o-generated questions, based on Wikipedia passages, closely mirror the wording of the answers in these passages, which differs from real-world scenarios.

Training a classifier for hallucinations on HalluRAG demonstrated moderate success, with classification accuracies significantly above chance level, ranging from 60% to 75%. Even though these results appear not sufficient on their own as a safeguard for hallucination detection, they clearly demonstrate that LLM layers contain information about the likelihood of hallucinations. While this behavior had been established for CEVs, it is a new finding for IAVs, indicating that this information is broadly distributed inside of an LLM. Notably, adding more internal states to the MLP input did not substantially enhance performance, suggesting that a single internal state captures as much relevant information as multiple internal states in most cases. Further research could investigate the performance across additional layers.

When testing for generalization capabilities of the trained classifiers on the RAGTruth dataset, these performed poorly. Similarly, MLPs trained on the RAGTruth dataset did not generalize well when tested on HalluRAG. This suggests that there might be overfitting for both MLP types (Table 2). From our point of view, this further emphasizes the need to create a wide variety of training data for hallucinations, using diverse prompt templates with substantial differences in format and length, rather than minor format changes. It would also be interesting to investigate the results of training and testing on a merged dataset combining RAGTruth and HalluRAG.

A surprising finding is that hallucinations in answerable and unanswerable prompts appear to be encoded differently, as test accuracies considerably increase for both cases when trained and tested separately (Table 3). We attribute the significantly higher test accuracies for unanswerable prompts to the fact that, in these cases, the internal states primarily encode an 'I don't know' signal or a similar response. Consequently, based on this encoding, an MLP only needs to determine whether the state reflects genuine cluelessness (non-hallucinated) or not (hallucinated). In contrast, for answerable prompts, the internal state encoding is inherently more complex, as it involves more than just distinguishing between cluelessness and its absence, which explains the comparatively lower test accuracy in these cases. Training separately appears to enable the classifier to focus specifically on one type of hallucination, thereby boosting accuracy. This distinction should be further investigated to gain a better understanding of how these types differ.

Importantly, test accuracies for Mistral-7B-Instruct-v0.1 are consistently higher than those for the LLaMA-2 series, indicating that the trained MLPs detect hallucinations more easily from Mistral-7B's internal states. This suggests that the likelihood of a hallucination is more clearly encoded in the internal states of Mistral-7B. We hypothesize that the training approach of Mistral-7B contributed to this in two ways. First, the distinct training approach prioritizes the efficient and thorough use of its 7 billion parameters rather than simply increasing the parameter count (Jiang et al. 2023). Second, a dataset of higher quality for improved responses was used. This might have resulted in a clearer and more effective representation of language within Mistral-7B, making it easier for an MLP to distinguish between a hallucination and a non-hallucination.

Ethical Statement

We constructed the HalluRAG dataset using publicly available Wikipedia articles, which, while open-source and potentially prone to inaccuracies or biases, ensure no reliance on personal data. Auto-labeling for hallucinated and non-hallucinated sentences was performed using GPT-4o, whose inherent biases may affect the dataset's fairness. To promote transparency, all data and code are publicly available for review and responsible use.

References

Agrawal, A.; Suzgun, M.; Mackey, L.; and Kalai, A. T. 2024. Do Language Models Know When They're Hallucinating References? arXiv:2305.18248.

Azaria, A.; and Mitchell, T. 2023. The Internal State of an LLM Knows When It's Lying. arXiv:2304.13734.

Chen, C.; Liu, K.; Chen, Z.; Gu, Y.; Wu, Y.; Tao, M.; Fu, Z.; and Ye, J. 2024. INSIDE: LLMs' Internal States Retain the Power of Hallucination Detection. arXiv:2402.03744.

Fang, X.; Huang, Z.; Tian, Z.; Fang, M.; Pan, Z.; Fang, Q.; Wen, Z.; Pan, H.; and Li, D. 2024. Zero-resource Hallucination Detection for Text Generation via Graph-based Contextual Knowledge Triples Modeling. arXiv:2409.11283.

Friel, R.; and Sanyal, A. 2023. Chainpoll: A high efficacy method for LLM hallucination detection. arXiv:2310.18344.

Hu, X.; Ru, D.; Qiu, L.; Guo, Q.; Zhang, T.; Xu, Y.; Luo, Y.; Liu, P.; Zhang, Y.; and Zhang, Z. 2024. RefChecker: Reference-based Fine-grained Hallucination Checker and Benchmark for Large Language Models. *arXiv preprint arXiv:2405.14486*.

Jiang, A. Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D. S.; de las Casas, D.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; Lavaud, L. R.; Lachaux, M.-A.; Stock, P.; Scao, T. L.; Lavril, T.; Wang, T.; Lacroix, T.; and Sayed, W. E. 2023. Mistral 7B. arXiv:2310.06825.

Kadavath, S.; Conerly, T.; Askell, A.; Henighan, T.; Drain, D.; Perez, E.; Schiefer, N.; Hatfield-Dodds, Z.; DasSarma, N.; Tran-Johnson, E.; Johnston, S.; El-Showk, S.; Jones, A.; Elhage, N.; Hume, T.; Chen, A.; Bai, Y.; Bowman, S.; Fort, S.; Ganguli, D.; Hernandez, D.; Jacobson, J.; Kernion, J.; Kravec, S.; Lovitt, L.; Ndousse, K.; Olsson, C.; Ringer, S.; Amodei, D.; Brown, T.; Clark, J.; Joseph, N.; Mann, B.; McCandlish, S.; Olah, C.; and Kaplan, J. 2022. Language Models (Mostly) Know What They Know. arXiv:2207.05221.

Kahl, S.; Löffler, F.; Maciol, M.; Ridder, F.; Schmitz, M.; Spanagel, J.; Wienkamp, J.; Burgahn, C.; and Schilling, M. 2024. Evaluating the impact of advanced llm techniques on ai-lecture tutors for a robotics course. *arXiv preprint arXiv:2408.04645*.

Levonian, Z.; Li, C.; Zhu, W.; Gade, A.; Henkel, O.; Postle, M.-E.; and Xing, W. 2023. Retrieval-augmented generation to improve math question-answering: Trade-offs between groundedness and human preference. *arXiv preprint arXiv:2310.03184*.

Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; tau Yih, W.; Rocktäschel, T.; Riedel, S.; and Kiela, D. 2021. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. arXiv:2005.11401.

Longpre, S.; Perisetla, K.; Chen, A.; Ramesh, N.; DuBois, C.; and Singh, S. 2022. Entity-Based Knowledge Conflicts in Question Answering. arXiv:2109.05052.

Manakul, P.; Liusie, A.; and Gales, M. J. F. 2023. SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models. arXiv:2303.08896.

Maynez, J.; Narayan, S.; Bohnet, B.; and McDonald, R. 2020. On Faithfulness and Factuality in Abstractive Summarization. arXiv:2005.00661.

OpenAI; Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; Avila, R.; Babuschkin, I.; Balaji, S.; Balcom, V.; Baltescu, P.; Bao, H.; Bavarian, M.; Belgum, J.; Bello, I.; Berdine, J.; Bernadett-Shapiro, G.; Berner, C.; Bogdonoff, L.; Boiko, O.; Boyd, M.; Brakman, A.-L.; Brockman, G.; Brooks, T.; Brundage, M.; Button, K.;

Cai, T.; Campbell, R.; Cann, A.; Carey, B.; Carlson, C.; Carmichael, R.; Chan, B.; Chang, C.; Chantzis, F.; Chen, D.; Chen, S.; Chen, R.; Chen, J.; Chen, M.; Chess, B.; Cho, C.; Chu, C.; Chung, H. W.; Cummings, D.; Currier, J.; Dai, Y.; Decareaux, C.; Degry, T.; Deutsch, N.; Deville, D.; Dhar, A.; Dohan, D.; Dowling, S.; Dunning, S.; Ecoffet, A.; Eleti, A.; Eloundou, T.; Farhi, D.; Fedus, L.; Felix, N.; Fishman, S. P.; Forte, J.; Fulford, I.; Gao, L.; Georges, E.; Gibson, C.; Goel, V.; Gogineni, T.; Goh, G.; Gontijo-Lopes, R.; Gordon, J.; Grafstein, M.; Gray, S.; Greene, R.; Gross, J.; Gu, S. S.; Guo, Y.; Hallacy, C.; Han, J.; Harris, J.; He, Y.; Heaton, M.; Heidecke, J.; Hesse, C.; Hickey, A.; Hickey, W.; Hoeschele, P.; Houghton, B.; Hsu, K.; Hu, S.; Hu, X.; Huizinga, J.; Jain, S.; Jain, S.; Jang, J.; Jiang, A.; Jiang, R.; Jin, H.; Jin, D.; Jomoto, S.; Jonn, B.; Jun, H.; Kaftan, T.; Kaiser, Ł.; Kamali, A.; Kanitscheider, I.; Keskar, N. S.; Khan, T.; Kilpatrick, L.; Kim, J. W.; Kim, C.; Kim, Y.; Kirchner, J. H.; Kiros, J.; Knight, M.; Kokotajlo, D.; Kondraciuk, Ł.; Kondrich, A.; Konstantinidis, A.; Kosic, K.; Krueger, G.; Kuo, V.; Lampe, M.; Lan, I.; Lee, T.; Leike, J.; Leung, J.; Levy, D.; Li, C. M.; Lim, R.; Lin, M.; Lin, S.; Litwin, M.; Lopez, T.; Lowe, R.; Lue, P.; Makanju, A.; Malfacini, K.; Manning, S.; Markov, T.; Markovski, Y.; Martin, B.; Mayer, K.; Mayne, A.; McGrew, B.; McKinney, S. M.; McLeavey, C.; McMillan, P.; McNeil, J.; Medina, D.; Mehta, A.; Menick, J.; Metz, L.; Mishchenko, A.; Mishkin, P.; Monaco, V.; Morikawa, E.; Mossing, D.; Mu, T.; Murati, M.; Murk, O.; Mély, D.; Nair, A.; Nakano, R.; Nayak, R.; Neelakantan, A.; Ngo, R.; Noh, H.; Ouyang, L.; O'Keefe, C.; Pachocki, J.; Paino, A.; Palermo, J.; Pantuliano, A.; Parascandolo, G.; Parish, J.; Parparita, E.; Passos, A.; Pavlov, M.; Peng, A.; Perelman, A.; de Avila Belbute Peres, F.; Petrov, M.; de Oliveira Pinto, H. P.; Michael; Pokorny; Pokrass, M.; Pong, V. H.; Powell, T.; Power, A.; Power, B.; Proehl, E.; Puri, R.; Radford, A.; Rae, J.; Ramesh, A.; Raymond, C.; Real, F.; Rimbach, K.; Ross, C.; Rotsted, B.; Roussez, H.; Ryder, N.; Saltarelli, M.; Sanders, T.; Santurkar, S.; Sastry, G.; Schmidt, H.; Schnurr, D.; Schulman, J.; Selsam, D.; Sheppard, K.; Sherbakov, T.; Shieh, J.; Shoker, S.; Shyam, P.; Sidor, S.; Sigler, E.; Simens, M.; Sitkin, J.; Slama, K.; Sohl, I.; Sokolowsky, B.; Song, Y.; Staudacher, N.; Such, F. P.; Summers, N.; Sutskever, I.; Tang, J.; Tezak, N.; Thompson, M. B.; Tillet, P.; Tootoonchian, A.; Tseng, E.; Tuggle, P.; Turley, N.; Tworek, J.; Uribe, J. F. C.; Vallone, A.; Vijayvergiya, A.; Voss, C.; Wainwright, C.; Wang, J. J.; Wang, A.; Wang, B.; Ward, J.; Wei, J.; Weinmann, C.; Welihinda, A.; Welinder, P.; Weng, J.; Weng, L.; Wiethoff, M.; Willner, D.; Winter, C.; Wolrich, S.; Wong, H.; Workman, L.; Wu, S.; Wu, J.; Wu, M.; Xiao, K.; Xu, T.; Yoo, S.; Yu, K.; Yuan, Q.; Zaremba, W.; Zellers, R.; Zhang, C.; Zhang, M.; Zhao, S.; Zheng, T.; Zhuang, J.; Zhuk, W.; and Zoph, B. 2024. GPT-4 Technical Report. arXiv:2303.08774.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

Shuster, K.; Poff, S.; Chen, M.; Kiela, D.; and Weston, J. 2021. Retrieval Augmentation Reduces Hallucination in Conversation. In Moens, M.-F.; Huang, X.; Specia, L.; and Yih, S. W.-t., eds., *Findings of the Association for Computational Linguistics: EMNLP 2021*, 3784–3803. Punta Cana, Dominican Republic: Association for Computational Linguistics.

Su, J.; Lu, Y.; Pan, S.; Murtadha, A.; Wen, B.; and Liu, Y. 2023. RoFormer: Enhanced Transformer with Rotary Position Embedding. arXiv:2104.09864.

Su, W.; Wang, C.; Ai, Q.; HU, Y.; Wu, Z.; Zhou, Y.; and Liu, Y. 2024. Unsupervised Real-Time Hallucination Detection based on the Internal States of Large Language Models. arXiv:2403.06448.

Thakur, N.; Bonifacio, L.; Zhang, X.; Ogundepo, O.; Kamalloo, E.; Alfonso-Hermelo, D.; Li, X.; Liu, Q.; Chen, B.; Rezagholizadeh, M.; and Lin, J. 2024. NoMIRACL: Knowing When You Don't Know for Robust Multilingual Retrieval-Augmented Generation. arXiv:2312.11361.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; Bikel, D.; Blecher, L.; Ferrer, C. C.; Chen, M.; Cucurull, G.; Esiobu, D.; Fernandes, J.; Fu, J.; Fu, W.; Fuller, B.; Gao, C.; Goswami, V.; Goyal, N.; Hartshorn, A.; Hosseini, S.; Hou, R.; Inan, H.; Kardas, M.; Kerkez, V.; Khabsa, M.; Kloumann, I.; Korenev, A.; Koura, P. S.; Lachaux, M.-A.; Lavril, T.; Lee, J.; Liskovich, D.; Lu, Y.; Mao, Y.; Martinet, X.; Mihaylov, T.; Mishra, P.; Molybog, I.; Nie, Y.; Poulton, A.; Reizenstein, J.; Rungta, R.; Saladi, K.; Schelten, A.; Silva, R.; Smith, E. M.; Subramanian, R.; Tan, X. E.; Tang, B.; Taylor, R.; Williams, A.; Kuan, J. X.; Xu, P.; Yan, Z.; Zarov, I.; Zhang, Y.; Fan, A.; Kambadur, M.; Narang, S.; Rodriguez, A.; Stojnic, R.; Edunov, S.; and Scialom, T. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv:2307.09288.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E.; Le, Q.; and Zhou, D. 2023. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. arXiv:2201.11903.

Wu, Y.; Zhu, J.; Xu, S.; Shum, K.; Niu, C.; Zhong, R.; Song, J.; and Zhang, T. 2023. RAGTruth: A Hallucination Corpus for Developing Trustworthy Retrieval-Augmented Language Models. arXiv:2401.00396.

Xu, Z.; Cruz, M. J.; Guevara, M.; Wang, T.; Deshpande, M.; Wang, X.; and Li, Z. 2024. Retrieval-Augmented Generation with Knowledge Graphs for Customer Service Question Answering. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, 2905–2909. New York, NY, USA: Association for Computing Machinery. ISBN 9798400704314.

Zhang, B.; and Sennrich, R. 2019. Root Mean Square Layer Normalization. arXiv:1910.07467.

Zhang, Y.; Li, Y.; Cui, L.; Cai, D.; Liu, L.; Fu, T.; Huang, X.; Zhao, E.; Zhang, Y.; Chen, Y.; Wang, L.; Luu, A. T.; Bi, W.; Shi, F.; and Shi, S. 2023. Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models. arXiv:2309.01219.