Reasoning Audits with Benchmarks: Identifying Hallucination Triggers in Large Language Models

Amine Turki, Raj Dhyaneshvar, Daniel Li, Mingxuan Tang
University of Tennessee, Knoxville

***Abstract*—Large Language Models (LLMs) frequently generate confident but ungrounded outputs, generally referred to as hallucination. This work investigates hallucination triggers through a multi-stage analysis combining feature extraction, chain-of-thought (CoT) auditing, and supervised learning. Using data generated via the HalluLens framework, TruthfulQA, and NovelHopQA, we construct labeled dataset with responses, CoT traces, and hallucination judgments from an LLM-based evaluator. In addition, a set of 63 textual features is derived from metrics based on linguistic, readability, sentiment, and embedding, and a CoT drift analysis that identify the earliest reasoning step leading to hallucination. Multiple classifiers including Logistic Regression, Random Forest, XGBoost, SVM, and MLP are trained to predict hallucination from these features. Results show that CoT-derived attributes, such as adjective density, numeric-token frequency, and semantic similarity scores, are consistently the strongest predictors. However, class imbalance limits model performance, particularly in identifying non-hallucinatory outputs. Overall, our findings highlight the importance of CoT structure in hallucination and the use of drift-aware reasoning audits to improve hallucination detection in future LLM systems.**

***Index Terms*—Hallucination, Chain of Thought feature, Machine Learning models, Embedding Comparison**

I. Introduction

Reliability in large language model (LLM) response is a major topic in the artificial intelligence (AI) research area. In addition to evaluating the factual correctness of LLM answers, inconsistent response from user input and training data, known as hallucination, are being studied to assert the safe reasoning of LLMs [1].

A. Existing Benchmark

There are various studies that have been done on Hallucinations in NLP. In FaithBench research, they focus on the test of model performances in judging how faithful a piece of summary stays to their respective context [2]. In HalluMix, they have similar focus but cover a broader field where they also include short question answering, multi-document questions and natural language inference [3]. Unlike the previous two, HALoGen focuses on the evaluation of a model's hallucination rate [4]. They do this by breaking down a model's output to atomic units that can be used to verify against a high quality knowledge source [3]. They also categorized hallucination into three classes, however, due to not having a unified concept that strictly defines all the types and causes of hallucination, this makes the field of hallucination research a bit unorderly [3].

To be the best of our knowledge, the most recent one, HalluLens research paper, the researchers try to shed light on the ambiguity regarding hallucinations so as to not mistaken it for factuality, while also setting up a comprehensive evaluation metric that can be used for model's hallucination performances [1]. The paper has stated that an oracle for factuality can be difficult to define as it somewhat overlaps with hallucinations, however factuality is not to be mistaken as a type of hallucination [1]. The paper referred to (Wang et al., 2024) and (Ji et al., 2023) where it states that factuality is the absolute correctness of the content generated with respect to established verification sources, while hallucinations is the consistency of the model output with respect to the knowledge that the model had access to, either in its training data or as input at inference time.

B. Research Plan & Methodologies

Using existing available datasets and HalluLens benchmark, we will highlight types of textual features that correlate with higher hallucination in LLMs. In addition, the ability of machine learning models to predict the likelihood of hallucination based on extracted features will be tested. Finally, detection of early signs of hallucinations before output will be studied by auditing the chain-of-thought (CoT).

II. Dataset generation

A. HalluLens

HalluLens GitHub [5] has been used to generate a high-quality dataset allowing feature-level analysis of hallucination triggers. HalluLens relies on WikiRank, GoodWiki, Wikipedia Dump, ITIS Taxonomy, and Medicine Dataset (Kaggle) [5]. GPT-4o-mini was the model used to generate question-answer pairs. gpt5 was the judge model in charge of evaluating whether or not GPT-4o-mini was hallucinating in its answers based on the benchmark. HalluLens allows to generate data with three different tasks:

- **PreciseWikiQA , task 1** – Metrics: False refusal rate, hallucination rate, correct answer rate. Rely on GoodWiki and WikiRank
- **LongWiki (long response), task 2** – Metrics: False refusal rate, precision, Recall@K, F1@K. Rely on GoodWiki and WikiRank
- **NonExistentRefusal (non-existent knowledge), task 3** – Fabricated entities; metric: false acceptance rate. ITIS Taxonomy and a Medicine Dataset (Kaggle)

1767 points have been generated with task 1 and 2312 points have been generated with task 3. The following features have been kept in the final dataset to train machine learning models:

- prompt : the initial prompt (questions)
- generation : the answers
- halu_test_res : hallucination score (0: No hallucination detected, 1: Hallucination detected)

B. NovelHopQA and TruthfulQA

NovelHopQA and TruthfulQA were used in this project to gather prompts that were fed to an llm, generating a response evaluated by a judge llm.

- **TruthfulQA** - TruthfulQA questions serve to ask for factual information. ex. What is the smallest country in the world that is at least one square mile in area?
- **NovelHopQA** - NovelHopQA questions serve to test analytical reasoning based on passages. The llm is given context in a passage and asked to answer a question based on that information in the prompt. The judge llm is given an optimal answer to use as reference for judging hallucinatory responses. ex. You are given the following text from a novel: Before Oliver had time to look round, Sikes had caught him... Question: What action did Sikes take immediately after catching Oliver under the arms?

C. Chain-of-Thought Extraction and Drift Analysis

In addition to the prompt, model response, and hallucination label, we extracted an estimated chain-of-thought (CoT) for each generated response. The CoT was produced by prompting an LLM to construct the reasoning steps as it generated the responses to the question. Each CoT consists of a small sequence of steps summarizing the logical progression of the model's output.

To detect early signs of hallucination, we applied a drift analysis procedure to every CoT. The objective of drift analysis is to identify the first step in which the model's reasoning becomes unsupported, incorrect, or inconsistent with the input prompt, leading it to hallucinate.

The drift analysis pipeline included the following components:

- **CoT Parsing:** For each sample, the CoT steps were normalized and segmented into individual reasoning units.
- **Error-Point Identification:** A secondary LLM(GPT-4.0) was prompted to identify the first CoT step that diverges from the prompt leading to hallucination.
- **Drift step index & Drift step text:** The index and the text of the earliest hallucinated step was recorded as the "drift_step_index," and "drift_step_text," providing a label and the text, where the reasoning begins to drift from the ground truth causing hallucination.
- **Drift step explanation:** The secondary LLM was prompted to reason out the choice and give a brief explanation of why it suspected the text as a drift.

This drift-aware annotation helped to interpret the hallucination detection by linking the hallucination outcome not only to the final answer but also to the internal reasoning path leading to it.

III. Machine learning model performance

The dataset generated using HalluLens benchmark [1] have been used to train the following models. Here are the variables of the dataset:

- 'prompt' : questions
- 'generation' : answers
- 'cot_steps' : the chain of the thought estimation
- 'halu_test_res' : hallucination score (0: No hallucination detected, 1: Hallucination detected)

ML Models	Type
Logistic Regression [6]	Statistical Model
Correlation Matrix [7]	Statistical Model
Mutual Information [8]	Information theory
Random Forest [9]	Tree-Based
XGBoost [10]	Gradient Boosting
SVM with RBF Kernel [11]	Kernel Methods
MLP Classifier [12]	Deep Learning

TABLE I
Machine Learning tested

A. Textual features computed

Type	Package	Nb features
Linguistic	spaCy	21 features
Readability	textstat	5 features
Sentiment	VADER	4 features
Embedding	BLEURT, Cross encoder, cosign similarity	3 features

TABLE II
Textual features

The following linguistic features have been computed using spaCy package [13]: basic structures, vocabulary richness, length, Part-of-speech (POS) ratios, stop-words, punctuation, named entities, and numbers.

The following readability features have been computed using textstat package [14]: Flesch Reading Ease score, Flesch-Kincaid Grade Level score, Gunning Fog index score, number of difficult words, and Dale-Chall score.

The following sentiment analysis features have been computed using VADER package [15]: negative, neutral, positive, and overall sentiments.

Embedding comparison between the prompt and the chain-of-thought have been computed using different scores: BLEURT from Google [16], roberta cross-encoder score [17], and cosine similarity [18].

Textual features of the chain-of-thought and the prompts have been computed. We end-up with 63 features. The target is the hallucination score. The details of the features can be found in our Github.

B. Model Validation

The following diagnostics and evaluation procedures were used to assert if the model is generalizing the data well, learning patterns correctly, not overfitting or underfitting:

- log-loss [19]
- cross-validation [20]
- learning curve [21] [22]
- ROC-AUX score [23]

Model	Log-loss	cross-validation	learning curve	ROC-AUX scores
Logistic Regression	✓	✓	✓	✓
Random Forest	Overfitting	Not generalizing well	Overfitting	Overfitting
Xgboost	✓	✓	Overfitting	✓
SVM	✓	✓	Overfitting	✓
MLP Classifier	Underfitting	Underfitting	Underfitting	Underfitting

TABLE III
VALIDATION MODELS

C. Machine Learning Classifier Performance

TABLE IV
COMPARISON OF MODEL PERFORMANCE

Model	**Precision (0/1)**	**Recall (0/1)**	**F1 (0/1)**	**Accuracy**
Logistic Regression	0.51/0.80	0.22/0.94	0.30/0.86	0.77
Random Forest	0.51/0.80	0.22/0.94	0.30/0.86	0.77
XGBoost	0.70/0.85	0.43/0.94	0.53/0.89	0.82
SVM (RBF)	0.86/0.83	0.35/0.98	0.50/0.90	0.84
MLP Classifier	0.44/0.79	0.20/0.92	0.28/0.85	0.76

A score of 1 corresponds to hallucination detection, and 0 corresponds to non-hallucination. The fact that all the models struggle to identify non-hallucination answers reflects the imbalance in the dataset: 3074 hallucination answers against 926 non-hallucination answers.

Hallucination rate is high using HalluLens benchmark due to how challenging the questions are. Task 1, in the HalluLens GitHub, evaluates the ability of the model to answer very specific questions from Wikipedia. Task 3 analysis LLM answers when it is being asked a question regarding a non-existent entity (fake medicine, fictional animals, species, bacteria, plants). Multiple researches regarding causes of hallucination show that LLM, under challenging questions, are trained to guess answers, like a student facing a hard quiz, rather than acknowledging their lack of knowledge [24]. This explains the high hallucination rate in task 3.

Some models during the validation steps, such as XGBoost and SVM, show that we need more data to perform better results. $10,872,912$ tokens were already used to generate the dataset for a total of $27.32. Generating more data using HalluLens benchmark would require more funding and push the scope of this project beyond just the context of class assignment.

Next steps would be to fine-tune these machine learning frameworks by doing a grid search using Optuna package [25]. An alternative is to use transformer-based classifier, given the huge number of features (63 features), even though using this kind of framework would require GPU resources to be effective.

D. Textual features importance

TABLE V
TOP-5 MOST IMPORTANT FEATURES PER MODEL

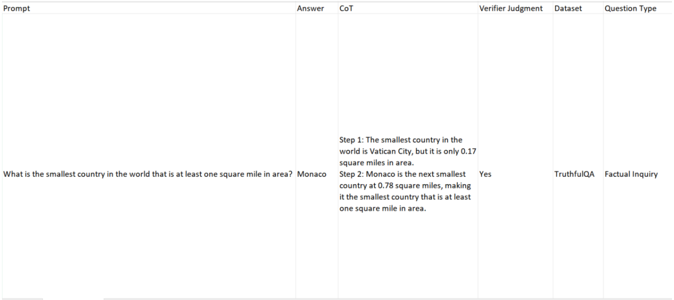
Model	**Top Features (Descriptions)**
Logistic Regression	1. BLEURT semantic similarity score 2. Overall sentiment of CoT (compound polarity) 3. Dale–Chall readability difficulty score (CoT) 4. Overall sentiment of prompt (compound polarity) 5. U.S. grade-level readability (CoT)
Correlation Matrix	1. Adjective density (CoT) 2. Noun density (CoT) 3. Count of rare/difficult words (CoT) 4. Auxiliary verb density (CoT) 5. Flesch Reading Ease (CoT)
Mutual Information	1. Adjective density (CoT) 2. Numeric token density (CoT) 3. Noun density (CoT) 4. Flesch Reading Ease score (CoT) 5. Stop-word density (CoT)
Random Forest	1. Numeric token density (CoT) 2. Adjective density (CoT) 3. Noun density (CoT) 4. Number of numeric tokens (CoT) 5. Named-entity density (CoT)
XGBoost	1. Adjective density (CoT) 2. Numeric token density (CoT) 3. Named-entity density (CoT) 4. Auxiliary verb density (CoT) 5. Stop-word density (CoT)
SVM	1. Stop-word density (CoT) 2. Adjective density (CoT) 3. Noun density (CoT) 4. Punctuation density (CoT) 5. BLEURT semantic similarity score (prompt vs CoT)
MLP	1. Flesch Reading Ease (CoT) 2. Flesch Reading Ease (prompt) 3. Gunning Fog Index (prompt complexity) 4. Number of sentences (CoT) 5. Total number of tokens (CoT)

Feature importance was computed using built-in functions and SHapley Additive exPlanations (SHAP) [26]. Almost all the features involve the chain-of-thought (CoT), which means that textual features in the CoT are more impactful than textual features in the prompt. Given the fact that some models did not have good validation metrics (section 3.2) and that more data is needed to improve performance, we cannot claim that specific features cause hallucination. The Next step would be to increases our dataset, dig deeper into the CoT features linked to hallucination, and connect these features to the initial user prompt.

IV. USING LLM AND JUDGE LLM TO EXTRACT FEATURE IMPORTANCE

A. Process

In this experiment, an LLM (gpt-3.5-turbo) receives prompts from the extracted datasets and returns responses that are classified as hallucinations or not hallucinations by an LLM (gpt-4o) serving as a judge. The question and the associated chain-of-thought (CoT) and answer are stored in a .csv file with a judge verdict of yes or no for hallucinatory status, sorted by question type and dataset origin.



Prompt	Answer	CoT	Verifier Judgment	Dataset	Question Type
What is the smallest country in the world that is at least one square mile in area?	Monaco	Step 1: The smallest country in the world is Vatican City, but it is only 0.17 square miles in area. Step 2: Monaco is the next smallest country at 0.78 square miles, making it the smallest country that is at least one square mile in area.	Yes	TruthfulQA	Factual Inquiry

Fig. 1. Example prompt-answer+CoT entry

- The csv file is read.
- The CoT and judge LLM's judgement are paired and extracted.
- The data is split into two groups: hallucinatory and non-hallucinatory.
- CoT features such as length, number of entities, word choice, etc. are identified.
- A random forest classifier is trained on CoT's to predict hallucination.

B. Results

With 5000 prompts and a train-test split of 5:1, these results were gathered using RandomForest classifier.

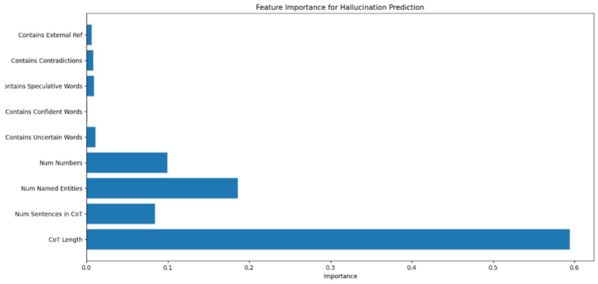


Fig. 2. CoT Feature Importance

	Precision	Recall	F1-score
False	0.83	0.92	0.87
True	0.24	0.11	0.15
Accuracy	–	–	0.78
Macro Avg	0.53	0.52	0.51
Weighted Avg	0.73	0.78	0.75

TABLE VI
RANDOMFOREST CLASSIFIER RESULTS

The precision, recall, and f1-scores fare well for non-hallucinatory (False) CoT's, but poorly for hallucinatory (True) CoT's. The non-hallucinatory entries can be removed, leaving only potentially hallucinatory entries. However, flagging hallucinatory responses directly will yield poor results. This is likely due to class imbalance, favoring predicting the more frequent class.

C. Analysis

Based on the feature importance for the extracted CoT features, word choices commonly seen as ”suspicious”, like ”definitely”, are not as condemning regarding hallucination. Similarly, references to outside sources do not necessarily indicate increased reliability (or untrustworthiness) in an LLM's response. CoT length, number of named entities/numbers, and number of sentences had high correlation with hallucination. This could indicate that large numbers of entities to keep track of when trying to interpret a prompt or a large amount of steps/thoughts needed to arrive at an answer can indicate to hallucination, perhaps due to increased difficulty in answering appropriately and correctly based on complex context.

V. Embedding Comparison Results

A. Method Process

The following is a basic process of how the embedding similarity was calculated Three variants of the Bert model embedding were tested and Bert-base model was selected and used for embedding similarity tests.

- Spacy's ”en_core_web_sm” model to handle POS tagging on the dataset texts
- The ”bert-base-uncased” model generates contextual embedding for each input
- Average the last four embedding layers of the ”bert-base-uncased” by default
- Average subword embeddings to get whole word embeddings
- Match each POS tagged word to its embedding
- Calculate the embedding similarity using Cosine similarity between two target text

B. Experiment results

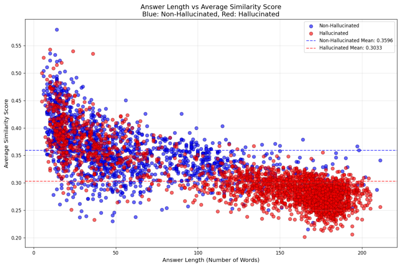


Fig. 3. Embedding similarity results on full_dataset.csv

In Figure 4, it is comparing the embedding similarity between the prompt and the generated response. All the samples are from the generated question/response stored in full_dataset.csv with 2561 hallucinated and 1525 non-hallucinated samples in total. The y-axis measures the embedding similarity rate, and the x-axis is the length of the generated response. The two lines through the middle shows the average embedding similarity for the two types of samples. There is a slight trend showing higher embedding similarity rate for non-hallucinated samples, however there might be bias considering the generated question styles in the HalluLens framework.

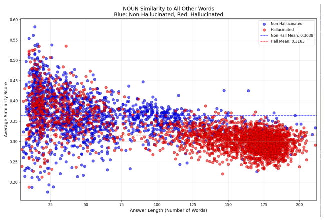


Fig. 4. Embedding similarity targeting noun

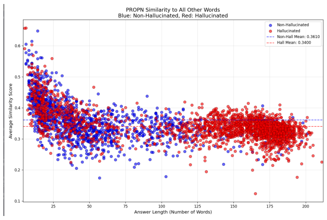


Fig. 5. Embedding similarity targeting proper noun

The two figures above target specific POS types where the embedding similarity is compared between one POS word type in the prompt to all the words in the generated response. Figure 5 is the embedding similarity of only nouns to the response, while Figure 6 compares proper nouns. A similar trend is shown for POS specific graphs, displaying the slight correlation between embedding similarity and response length in hallucination and non-hallucination cases.

VI. Conclusion

More data is needed to reliably claim that specific features trigger hallucination. The current results indicate that features in the chain of thought are highly correlated to hallucination. As a result, digging deeper in the chain of thought textual features linked to hallucination and associating these features with user prompts is a next step. Machine learning frameworks struggle to detect hallucination. Consequently, transformer-based classifier implementation in our pipeline is a next step that will be focused on.

References

[1] Y. Bang et al., "Hallulens: Llm hallucination benchmark," 2025. arXiv: 2504.17550 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2504.17550.

[2] F. S. Bao et al., *Faithbench: A diverse hallucination benchmark for summarization by modern llms*, 2024. arXiv: 2410.13210 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2410.13210.

[3] D. Emery, M. Goitia, F. Vargus, and I. Neagu, *Hallumix: A task-agnostic, multi-domain benchmark for real-world hallucination detection*, 2025. arXiv: 2505.00506 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2505.00506.

[4] A. Ravichander, S. Ghela, D. Wadden, and Y. Choi, *Halogen: Fantastic llm hallucinations and where to find them*, 2025. arXiv: 2501.08292 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2501.08292.

[5] Y. Bang et al., *Hallulens: Llm hallucination benchmark*, 2025. arXiv: 2504.17550 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2504.17550.

[6] D. R. Cox, "The regression analysis of binary sequences," *Journal of the Royal Statistical Society: Series B*, 1958.

[7] K. Pearson, "Note on regression and inheritance in the case of two parents," *Proceedings of the Royal Society of London*, 1895.

[8] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, 1948.

[9] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[10] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[11] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.

[12] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986.

[13] M. Honnibal, I. Montani, S. V. Landeghem, and A. Boyd, *Spacy: Industrial-strength natural language processing in python*, Version 3.x, 2020. [Online]. Available: https://spacy.io.

[14] S. Bansal, *Textstat: Text statistics and readability metrics*, 2018. [Online]. Available: https://github.com/shivam5992/textstat.

[15] C. Hutto and E. Gilbert, "VADER: A parsimonious rule-based model for sentiment analysis of social media text," in *Proceedings of the 8th International Conference on Weblogs and Social Media*, 2014. [Online]. Available: https://ojs.aaai.org/index.php/ICWSM/article/view/14550.

[16] T. Sellam, D. Das, and A. Parikh, "Bleurt: Learning robust metrics for text generation," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020. [Online]. Available: https://arxiv.org/abs/2004.04696.

[17] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019. [Online]. Available: https://arxiv.org/abs/1908.10084.

[18] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008. [Online]. Available: https://nlp.stanford.edu/IR-book/.

[19] S. Kullback and R. A. Leibler, "On information and sufficiency," *Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[20] M. Stone, "Cross-validatory choice and assessment of statistical predictions," *Journal of the Royal Statistical Society: Series B*, vol. 36, no. 2, pp. 111–147, 1974.

[21] S.-i. Amari and N. Murata, "A statistical learning theory of learning curves," *Neural Networks*, vol. 5, no. 2, pp. 283–297, 1992.

[22] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012, Learning curves discussed in Chapter 5.

[23] T. Fawcett, "An introduction to roc analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.

[24] A. T. Kalai, O. Nachum, S. S. Vempala, and E. Zhang, *Why language models hallucinate*, 2025. arXiv: 2509.04664 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2509.04664.

[25] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, *Optuna: A next-generation hyperparameter optimization framework*, 2019. arXiv: 1907.10902 [cs.LG]. [Online]. Available: https://arxiv.org/abs/1907.10902.

[26] S. Lundberg and S.-I. Lee, *A unified approach to interpreting model predictions*, 2017. arXiv: 1705.07874 [cs.AI]. [Online]. Available: https://arxiv.org/abs/1705.07874.