Detecting Hallucinations in Large Language Model Generation: A Token Probability Approach

Anonymous ACL submission

Abstract

With the rise of Large Language Models (LLMs) in recent times, concerns about their tendency to hallucinate and produce inaccurate outputs have also increased. Detecting such hallucinations is crucial for ensuring trustworthiness in applications relying on LLM-generated content. Current methods, often resource-intensive and reliant on extensive LLMs and intricate linguistic and semantic analyses, are not easily reproduced. This paper seeks to introduce a simpler method to detect hallucinations in LLM generations using purely numerical features. By evaluating token probabilities within the generated content and vocabulary, the method achieves promising results, surpassing state-of-the-art outcomes in Summarization and Question Answering on the Hallucination Evaluation for Large Language Models (HaluEval) benchmark. This method demonstrates effectiveness in pinpointing hallucinatory content, offering a more efficient pathway for real-time LLM output evaluation without the need for intricate linguistic analyses.

1 Introduction

Large Language Models (LLMs) have become the core of many state-of-the-art Natural Language Processing (NLP) algorithms and have revolutionized various domains in NLP and computer vision and even more specialized applications in healthcare, finance, and the creative arts. Because of their impressive Natural Language Generation (NLG) capabilities (Zhao et al., 2023; Kaddour et al., 2023), they have attracted great interest from the public with great modern tools like ChatGPT (Hosseini et al., 2023), Github-Copilot (Chen et al., 2021), Dalle (Zeqiang et al., 2023), and others (Zhao et al., 2023). These models, with millions to billions of parameters, are often praised for their impressive ability to generate human-like text and tackle intricate tasks with limited to no fine-tuning with techniques like In-Context-Learning (Lu et al., 2023).

Since many of the most popular applications and state-of-the-art algorithms in NLP rely on LLMs, any error they produce affects the results. Particularly in the cases of a Chatbot like ChatGPT, the generated responses are expected to maintain factual consistency with the source text (Lei et al., 2023). Currently, a pressing concern with LLMs is their propensity to "hallucinate," which intuitively means to produce outputs that, while seemingly coherent, might be misleading, fictitious, or not genuinely reflective of their training data or real-world facts (Ji et al., 2023). It has been widely observed that models can confidently generate fictitious information, and worryingly, there are few effective approaches to identify LLM hallucinations (Ji et al., 2023; Kaddour et al., 2023) suitably. And if we cannot identify them even less, we can fix them in real time on an application like ChatGPT.

Furthermore, the consequences of hallucinatory-generated text when used by the public are a significant ethical concern. This fictitious content can lead to misinformation and have severe implications in delicate medical, legal, educational, and financial fields. Besides the ethical consequences, these errors can lead to limitations in the use of the LLMs to automate programming tasks completely and tedious hand-work, limiting their contribution to NLP tasks (Ji et al., 2023; Kaddour et al., 2023).

While there have been efforts to detect and mitigate these hallucinations, many of the prevalent methods rely on leveraging other massive LLMs (Li et al., 2023; Zhang et al., 2023) or intricate linguistic and semantic analyses (Zhang et al., 2023; Manakul et al., 2023; Lei et al., 2023; Wang et al., 2022). The former approach escalates the computational costs, making it less accessible to researchers with limited resources. The latter, though effective to some extent, can be cumbersome and may need to be better for real-time applications like ChatGPT. Additionally, current research has shown that even state-of-the-art approaches (Ji et al., 2023;

Kaddour et al., 2023; Li et al., 2023; Lei et al., 2023) struggle to detect hallucinations. An example of that is the research done on the recently released Hallucination Evaluation for Large Language Models (HaluEval) benchmark dataset (Li et al., 2023) with four different tasks: Summarization, Question Answering, Dialogue, and General User Queries. In this benchmark, the initial and current approaches using the latest billion parameter models like GPT-3.5 (Ye et al., 2023), GPT-4 (Mao et al., 2023), LLama-2 (Touvron et al., 2023), Alpaca (Hu et al., 2023), struggle to get a good performance on this benchmark.

However, recent research has hinted at the potential of numerical features mathematically (Lee, 2023) and empirically (Manakul et al., 2023), such as the entropy of vocabulary and token probabilities, as indicators of hallucinations on LLM outputs. If effectively utilized, these features could provide a resource-efficient method to detect and mitigate hallucinations. In this paper, we investigate this particular approach further. Based on the premise that numerical trends can effectively differentiate authentic content from fabricated outputs, we conduct a detailed assessment using the HaluEval benchmark.

The results of our research not only highlight the effectiveness of this method in comparison with current approaches but also pave the way for potential uses that validate the credibility of LLM outputs, decreasing the demand for heavy computational power or complex linguistic analysis. The results obtained on tasks like Summarization and Question Answering surpass significantly the current state-of-the-art results. Our main contributions are (i) the performance evaluation of two simple classifiers (Logistic Regression and a Simple Neural Network) using numerical features based on generated token probabilities of a given LLM with great results in most tasks in the HaluEval benchmark. (ii) We provide the impact of using different LLMs with the same approach on the obtained results. (iii) Finally, we study the importance of each numerical feature we decided to use per task. We release all our code at [Removed for blind review].

This paper is structured as follows. First, we present the related works. Second, we describe our methodology. Next, we offer the experiments performed and the results obtained from them in a given dataset. After that, we present a discussion and future work section, followed by the conclusions. Finally, we conclude the paper with the limitations section.

2 Related Work

The occurrence of hallucinations in Large Language Models (LLMs) raises concerns, compromising performance in practical implementations like chatbots producing incorrect information. Various research directions have been explored to detect and mitigate hallucinations in different Natural Language Generation tasks (Ji et al., 2023). A verification system has been proposed for text summarization to detect and mitigate inaccuracies (Zhao et al., 2020; Huang et al., 2021; Ji et al., 2023). In dialogue generation, hallucinations have been studied with retrieval augmentation methods (Shuster et al., 2021; Ji et al., 2023). Also, researchers aim to understand why hallucinations occur in different tasks and how these reasons might be connected (Zheng et al., 2023; Das et al., 2023).

Recent approaches to detect and mitigate hallucinations include self-evaluation (Kadavath et al., 2022) and self-consistency decoding for intricate reasoning tasks (Wang et al., 2022). Structured data interfaces like knowledge graphs are proposed for gathering evidence (Jiang et al., 2023). Token probabilities as an indicator of model certainty have been used, addressing uncertainty in sequential generation tasks (Xiao and Wang, 2021; Malinin and Gales, 2020). Scores from conditional language models are used to assess text characteristics (Yuan et al., 2021; Fu et al., 2023). Recently the work SelfCheckGPT suggests that LLM's probabilities correlate with factuality (Manakul et al., 2023). Finally, a mathematical investigation by Lee et al. (Lee, 2023) suggests that token probabilities are crucial in generating hallucinations in GPT models under certain assumptions.

Despite research efforts, the full range of token probabilities influencing hallucinations is yet to be explored. This study investigates the influence of varying token probabilities generated by different LLMs, emphasizing the practicality of leveraging large LLMs with fewer parameters for real-time applications, given challenges associated with replicating techniques using larger models like GPT-3, Llama-2, and Alpaca.

3 Methodology

We implement two classifiers, a Logistic Regression (LR) and a Simple Neural Network (SNN) using four numerical features obtained from the

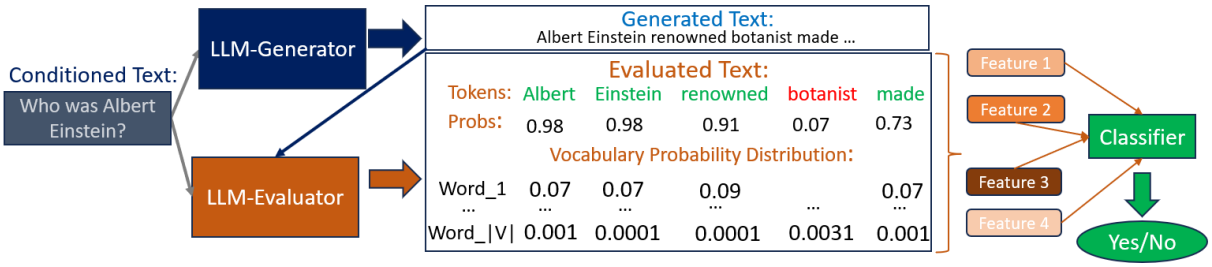

Figure 1: General Pipeline of the Proposed Methodology.

token probabilities and vocabulary entropy from a forward pass to an LLM with the conditional generation approach (Zhang et al., 2022). In this section, we described our entire methodology to detect hallucinations on a generated text by an LLM conditioned on a piece of text, which can be a document, query, instruction, or dialogue history.

3.1 Problem Statement

Given a pair of texts *(condition-text, generated-text)* that represent the text used to condition the LLM to its generation. We want to detect if a given *generated-text* is a hallucination.

3.2 General Pipeline

Given a set of pairs of texts of the type *(conditioned-text, generated-text)* from an LLM (we will call it the LLM-Generator (LLM_G)), we extract four numerical features based on the generated tokens and vocabulary tokens probabilities from another LLM that we call the LLM-Evaluator (LLM_E).[1] The four numerical features are the minimum token probability from the generated text; the average token probabilities; the maximum difference across all the tokens in the *generated-text* between the token with the highest probability according to LLM_E and the probability that LLM_E gives to the current token; and finally the minimum difference across all the tokens in the generated text between the token with the highest probability and the token with the lowest probability according to LLM_E.

Then, using these four features, we trained two different classifiers: a Logistic Regression (LR) and a Simple Neural Network (SNN). Finally, we evaluate these classifiers on a test set they did not see before. Figure 1 illustrates the process.

3.3 Features Description

We will delve into more detail in this section on each feature extracted. Every feature is computed using token probabilities and the vocabulary probability distribution corresponding to each token on the *generated-text*. However, let's give some definitions:

1. We will name the token at position j on the *conditioned-text* as c_j. The token at position i on the *generated-text* as t_i. The token at position k of the Vocabulary of the LLM_E as v_k.

2. Let m be the total tokens according to LLM_E's Tokenizer of the *conditioned-text* and n the total tokens of *generated-text*.

3. We will define the token probability of t_i given LLM_E as $P_{LLM_E}(t_i) = P(t_i|t_{i-1}, ..., t_1, c_m, ..., c_1, \theta)$. Where θ are the parameters of the LLM_E model.

4. We will define the token probability of each token of the Vocabulary of the LLM_E corresponding to t_i as $P_{LLM_{E_i}}(v_k) = (v_k|t_{i-1}, ..., t_1, c_m, ..., c_1; \theta)$ for every k.

5. We will define the token with the highest probability at position i in the *generated-text* according to LLM_E as the $v^* = \arg\max_k P_{LLM_{E_i}}(v_k)$

6. We will define the token with the lowest probability at position i according to LLM_E as the $v^- = \arg\min_k P_{LLM_{E_i}}(v_k)$

Now, we will provide a natural language description of the four features and, next, the mathematical definition.

Minimum Token Probability (mtp): Take the minimum of the probabilities that the LLM_E gives to the tokens on the *generated-text*.

Average Token Probability (avgtp): Take the average of the probabilities that the LLM_E gives to the tokens on the *generated-text*.

Max.-Diff. Vocab and Token Probs. (MDVTP): Take the maximum from all the differences

[1] Which could be the same as LLM_G.

between the token with the highest probability according to LLM_E at position i and the assigned probability from LLM_E to t_i.

Min. of the Max. Diff. Vocab. Probs (MMDVP): Take the maximum from all the differences between the token with the highest probability according to LLM_E at position i (v^*) and the token with the lowest probability according to LLM_E at position i (v^-).

The mathematical definition of each of these features would be:

$$mtp = \min_i P_{LLM_E}(t_i) \quad (1)$$

$$avgtp = \frac{\sum_{i=1}^{n} P_{LLM_E}(t_i)}{n} \quad (2)$$

$$MDVTP = \max_{1 \le i \le n} (P_{LLM_E}(v^*) - P_{LLM_E}(t_i))$$

$$MMDVP = \min_{1 \le i \le n} (P_{LLM_{E_i}}(v^*) - P_{LLM_{E_i}}(v^-))$$

These four numerical features are inspired by the mathematical investigation of the GPT model by Lee (2023), and recent results from Manakul et al. (2023) that suggest the correlation between the minimum token probability on the generation, the average of the token probabilities, the average entropy, and the maximum entropy. In the mathematical investigation by Lee (2023), two key assumptions are made (Assumption 6 and 7), which state:

Assumption 6: "When the input context does not provide sufficient information for a clear and optimal token choice, the estimated probabilities $p(x_{i+1})$ obtained are distributed such that the difference between the highest and subsequent probabilities is relatively small."

Assumption 7: "Hallucination takes place when the GPT model generates a low-probability token x_{i+1}, given the previous tokens $x_1, x_2, \ldots, x_i$, and subsequently employs this token as input for predicting the next token x_{i+2}."

The author proposes that a reliable indicator of hallucination during GPT model generation is the low probability of a token being generated. This is based on the assumption that forcing the model to generate such a low-probability token occurs when the difference between the token with the highest probability and all other tokens is less than a small constant δ. To avoid the computational cost of calculating differences across an extensive vocabulary and large generated text, the $MMDVP$ is utilized as an indicator.

Diverging from previous papers, the approach here differs in several aspects. Instead of using only the Language Model generating the text (LLM_G), the argument is made that depending on the task and model type, different Language Models (LLM_E) can provide consistent but quantitatively different results than using probabilities from LLM_G. The belief is that probabilities from a different model, varying in architecture, size, parameters, context length, and training data, can also serve as reliable indicators of hallucinations in the text generated by LLM_G. Moreover, since LLM_E and LLM_G are not the same in this approach, an additional numerical feature, $MDVTP$ (Maximum Difference in Vocabulary Token Probabilities), is introduced. This feature indicates a high difference between the maximum probability token in the vocabulary of LLM_E and the token generated by LLM_G, suggesting a disagreement between the two models on the token in that position.

3.4 Feature Extraction

In the previous section, we described the numerical features selected, but there is still the process of extracting these features. To extract the features, we used LLM_E models that can be used for the Conditional Generation Task, where the core idea is that they generate text based on a given condition or context. Particularly, in our case, is a force decoding since the tokens of *generated-text* were generated by a different LLM (LLM_G). Instead of letting the model generate the answer token-by-token from the *conditioned-text* alone, we provide it with the token predicted by LLM_G at each step. This way, LLM_E is forced to follow the path to generate the *generated-text* and, from there, extract the token probabilities from LLM_E if it would generate that sequence itself. Then, using these token probabilities, we compute the four numerical features previously described.

3.5 Models Specification

The classifiers used are a Logistic Regression (Wright, 1995) (LR) and a Simple Neural Network (SNN). Both classifiers for a data point of the type *(conditioned-text, generated-text)* only use the four numerical features extracted. We selected the Logistic Regression for its simplicity, fast training, and effectiveness in binary classification tasks. However, we implemented a basic neural network to explore potential and more complex non-linear relationships in the data and provide a more intri-

cate comparison to the logistic regression model. The specific architecture of this network is outlined below:

Input Layer: Consisting of 4 neurons, each representing one of the numerical features.

Hidden Layer 1: Consisting of 512 neurons using the ReLU (He et al., 2018) activation function.

Hidden Layer 2: Another layer consisting of 512 neurons using the ReLU activation function.

Output Layer: Consisting of a single neuron utilizing a sigmoid activation function for binary classification.

4 Experimental Setup and Results

In this section, we described the details of the experimental setup and our results obtained. All our current experiments have been done on the HaluEval benchmark dataset (Li et al., 2023).

4.1 Datasets

The current dataset used in our experiments is the Hallucination Evaluation for Large Language Models (HaluEval) benchmark an extensive collection of generated and human-annotated hallucinated samples for evaluating the performance of LLMs in recognizing hallucinations. HaluEval includes 5,000 general user queries with ChatGPT responses and 30,000 task-specific examples (10,000 per task) from three tasks: question answering, knowledge-grounded dialogue, and text summarization. Specifically, the authors consider three types of hallucination patterns for knowledge-grounded dialogue, i.e., extrinsic-soft, extrinsic-hard, and extrinsic-grouped; in question answering with four types, i.e., comprehension, factualness, specificity, and inference; and three types of hallucination patterns for text summarization, i.e., factual, non-factual, and intrinsic (Li et al., 2023).

We intend to experiment with more Hallucination datasets and benchmarks to come. Still, the results are already interesting enough to be shared with the research community.

4.2 LLM Evaluators used

The LLMs selected as evaluators to study the impact of factors such as the architecture, training method, size, and training data include GPT-2, specifically its large version (`gpt2-large`) (Radford et al., 2019); Bidirectional and Auto-Regressive Transformers (BART), particularly its CNN-Large version (`bart-large-cnn`) (Lewis et al., 2019); Longformer Encoder-Decoder (`LED`) (Beltagy et al., 2020), with a particular focus on the version fine-tuned on the arXiv dataset (`led-large-16384-arxiv`). We utilized the Hugging Face transformers library for evaluation.[2]

In the case of BART and LED, we used their `BartForConditionalGeneration` and `LEDForConditionalGeneration` setup, respectively. In the case of GPT-2 we used its `GPT2LMHeadModel` setup. Additionally, when we do forward to these models, with a pair of *(conditioned-text, generated-text)*, in the case of BART and GPT-2 there is a maximum length of 1024 since we need to get the token probabilities of the *generated-text* all the experiments truncate the *conditioned-text* to only the first 700 words and that 700 words are the only ones used as *conditioned-text* in the forward pass so we could have a span of at least 300 word for the *generated-text* that can be large in tasks like the summarization or general user queries. We did not test GPT-3 versions due to cost limitations and lack of access to the token probabilities.

4.3 Training Process of the Classifiers

We took the data points of every task to train both classifiers for each of the tasks in the HaluEval benchmark. We converted them to two data points: *(conditioned-text, right-answer)* and *(conditioned-text, hallucinated-answer)*. Therefore, for our approach, the datasets would be of 20,000 examples for each of the question answering, knowledge-grounded dialogue, and text summarization tasks where in each case half of the dataset is comprised of data points of the type *(conditioned-text, right-answer)* and the other half are of the type *(conditioned-text, hallucinated-answer)*. In the case of the general-user queries, the dataset is already on the format of having each data point classified as an hallucination or not, therefore the size of the dataset is the same which is 5,000.

Then, with this adaptation of the HaluEval benchmark dataset when we were approaching a given task, we will sample randomly 10% of the data points (half with the *right-answer* and the other half of the same *conditioned-text* but with its respective *hallucinated-answer*).[3] These 10%

[2] https://huggingface.co/

[3] Example: For the Summarization task, there are 20,000 data points, so we randomly sample 2,000 data points where 1,000 have the correct answer and the other 1,000 the halluci-

data points are used to train both classifiers (LR and SNN), and we test the model capabilities on the remaining 90% of the dataset for a given task.

To train the LR, we used the sklearn library[4] using the Limited-memory Broyden–Fletcher–Goldfarb–Shanno Algorithm (lbfgs) solver (Saputro and Widyaningsih, 2017) with the default parameters set by sklearn.

The SNN was trained during 10^4 epochs. We used the Adam (Kingma and Ba, 2014) optimizer with a learning rate of 10^{-3} . All the experiments were performed on Google Colab[5] with a the T4 GPU.

4.4 Results

We evaluate each classifier trained on the 10% data of the given task on the other 90%. We wanted to study the performance of each model in detail with both classes. The positive class which classifies a data point as a pair of *(conditioned-text, right-answer)* and the negative class which classifies a data point as a pair of *(conditioned-text, hallucinated-answer)*. Therefore, to ensure a comprehensive understanding of the capabilities of each model, we selected the following metrics: Accuracy, F_1, Precision Recall Area Under Curve (PR-AUC), and the negative class counterpart, the Negative F_1 Score computed from the Negative Predictive Value (NPV) and True Negative Rate (TNR).

For the sake of comparison, Table 1 shows the current state of the art published, particularly the best result obtained from all the methods explored in each paper. Next, Table 2 shows the accuracy results on the test set for each task using every LLM_E selected and the Logistic Regression as the classifier. As it can be appreciated, the Logistic Regression obtains great results compared to what previous approaches would have gotten on the 90% of the dataset.

Tables 3, 4, 5 show our average[6] results per model of our approach in each metric evaluated on the test set for each of the tasks of summarization, question-answering, knowledge-grounded dialogue, and general user queries respectively. It is true that the current methods tested are based on In-Context-Learning approaches and Zero-Shot fashion and evaluated in 100% of the dataset. While our approach employs supervised learning, we consider it a fair comparison as we utilize only 10% of the data for training, testing the models on the remaining 90%. We argue that current approaches won't yield significantly better results on the 90% of the dataset than what they achieve on the full 100%. However, for other approaches that have been tested on a subset of a dataset, we cannot be entirely certain if those methods would outperform ours. Let's now discuss our findings per task.

4.4.1 Summarization

Our results from Table 3 indicate that by using the `gpt-2-large` model as LLM_E and the LR classifier surpasses state-of-the-art accuracy. Employing the SNN also yields outstanding performance, indicating a high accuracy for predictions in both positive and negative classes. Notably, PR-AUC is 93%, with F_1 and Negative-F_1 scores of 94.60% and 94.94%, showcasing a balanced model performance in terms of precision, recall, false positives, and false negatives for both classes. These results outperform current state-of-the-art approaches, even with the latest GPT model versions.

Additionally, `bart-cnn-large` outperforms state-of-the-art using LR and obtains an impressive accuracy of 82.3% with the SNN classifier. This success is noteworthy given differences in architecture, parameters, and training data compared to `gpt-2-large`, even when `bart-cnn-large` maintains an advantage by incorporating the CNN-Daily-Mail dataset (Chen et al., 2016) in its training, a foundation in the HaluEval benchmark for summarization.

In contrast, without prior training on this data, the `LED` model excels by considering the entire context without truncation. `LED` achieves 78% accuracy using the SNN classifier, surpassing the state of the art. This shows the significance of numerical features on the HaluEval benchmark, irrespective of differences in hypotheses obtained from a distinct LLM_E.

4.4.2 Question Answering

This task is divided into two aspects: one involving only the question and answer, and the other incorporating knowledge with the correct answer as part of the *conditioned-text*. Our results from Table 4 show a surprising outcome in the Question Answering task. Contrary to expectations, the

nated answer.

[4]https://pypi.org/project/sklearn/

[5]https://colab.research.google.com/

[6]The average is because the training set and testing set are sampled randomly in each run we have been able to run three iterations in each case and average them.

Paper	Summ.	QA	KGD	GUQ
(Li et al., 2023) (The entire dataset)	0.61	0.77	0.74	0.87
(George and Stuhlmüller) (Data Points: 4000)	0.76	-	-	-
(Lei et al., 2023) (Data Points: 2130 in Summ. and 4170 in QA)	0.677	0.849	-	-

Table 1: Current State of the Art results on the HaluEval benchmark for each task measure in Accuracy. The text next to the cite of the paper is the number of data points per task used for evaluation.

Model	Summ.	QA	KGD	GUQ
gpt2-large	**0.93**	0.76	0.60	0.55
bart-large	0.68	**0.93**	0.68	0.54
LED	0.52	0.87	0.62	0.52

Table 2: Our results for each LLM_E and task using the LR classifier and measure in accuracy on the test set.

Model	Acc	F_1	PR-AUC	Neg F_1
gpt2-large	**0.94**	**0.94**	**0.97**	**0.94**
bart-large	0.82	0.83	0.78	0.81
LED	0.78	0.77	0.86	0.77

Table 3: Summarization Task test set results average using the SNN Classifier.

best performance comes not from gpt-2-large but from bart-cnn-large. It achieves an accuracy of 93.57% with the LR classifier and 94.64% with the SNN. These results are accompanied by a PR-AUC of 96%, F_1 score of 94.62%, and Negative-F_1 score of 94.65%, indicating a high rate of correct predictions across positive and negative classes. Furthermore, the LED model surpasses the state-of-the-art performance, with an accuracy of 87.48% (LR) and 88.08% (SNN). The F_1 score is 88.35%, and the Negative-F_1 score is 87.8%, complemented by a PR-AUC of 93%.

Also interesting is how the inclusion of the Knowledge on the *conditioned-text* did not improve the results, and in some instances like gpt-2-large the performance decreased.

4.4.3 Knowledge-Grounded Dialogue

The results of this task in Table 5 showed that all the LLM_E selected using both classifiers were not enough to surpass the state-of-the-art. However, the results are still competitive, obtaining the best results with bart-cnn-large. Additionally, integrating the Knowledge on the *conditioned-text* only decreased the results.

4.4.4 General User Queries

For this task, a table wasn't included due to result similarities. When employing various LLM_E models with the SNN classifier, the results indicated overfitting to the negative class, yielding an accuracy of 81%, F_1 of 1%, PR-AUC of 10%, and F_1-Negative of 90%. This overfitting is attributed to dataset imbalance, where out of 5,000 examples, only 977 are not hallucinations. An alternative attempt with a training set of 500 positive and 500 negative examples tested on the remaining 4,000 revealed limited success, with the best accuracy at 69% and F_1 at 0.23%.

Model	Acc	F_1	PR-AUC	Neg F_1
gpt2-large	0.78	0.78	0.86	0.77
bart-large	**0.94**	**0.94**	**0.96**	**0.94**
LED	0.88	0.88	0.93	0.87
+Knowledge				
gpt2-large	0.74	0.75	0.83	0.73
bart-large	**0.94**	**0.94**	**0.96**	**0.94**
LED	0.88	0.88	0.93	0.88

Table 4: Question Answering Task test set results average using the SNN Classifier. The +Knowledge rows highlight the results of the models by using the extra Knowledge.

Model	Acc	F_1	PR-AUC	Neg F_1
gpt2-large	0.64	0.63	0.68	0.65
bart-large	**0.69**	**0.64**	**0.78**	**0.72**
LED	0.58	0.58	0.68	0.60
+Knowledge				
gpt2-large	0.63	0.63	0.68	0.63
bart-large	0.67	0.65	0.77	0.70
LED	0.59	0.59	0.68	0.61

Table 5: Knowledge-Grounded Dialogue Task test set results average using the SNN Classifier.

5 Discussion

The results are based on a supervised learning approach, different from current methods that do not utilize any data for training. Notably, excellent performance was observed in Summarization and Question Answering tasks using gpt-2-large and bart-cnn-large as LLM_E. Also, competitiveness was noted in the Knowledge-Grounded Dialogue task, contrasting with lower performance in the General-User-Queries dataset compared to

state-of-the-art approaches.

This suggests a potential mismatch between the Conditional Generation Approach and tasks involving executing instructions, such as knowledge-grounded dialogue and general-user-queries. The complexity of nuanced dialogues and diverse user queries may require specialized models. Another clear possibility is that the numerical features are not enough in these tasks to detect the hallucinations of the HaluEval benchmark and might follow another type of pattern, like the contextual intricacies of a real-time dialogue. In any case, both are interesting research questions that can enlighten more on the path to understanding and mitigating the hallucinations in LLMs.

Another research question is the impact of manual annotations on the approach's efficacy, hinting at the potential advantages of using automatically generated benchmarks and datasets. Despite this, even with automatically generated data, current state-of-the-art LLMs, employing techniques like Chain-Of-Thought (Li et al., 2023), perform worse than the proposed approach. This prompts consideration for a hybrid approach in future work.

Feature importance analysis [7] highlights $avgtp$ as a crucial feature in most tasks, representing the confidence of LLM_E in generating a sequence. Low-confidence sequences may indicate hallucinations, while high-confidence sequences align more closely with training data. However, results reveal that for specific LLM_E and task pairings, the critical feature can vary, exemplified by `gpt-2-large` in the Question Answering task, where $MDVTP$ emerges as the most crucial feature.

6 Future Works

The first avenue is broadening the set of numerical features to capture more intricate patterns, potentially enhancing the model's performance, particularly in tasks like dialogue and general user queries. A second path would be to explore the impact of using different LLMs as LLM_E, including larger models such as GPT-3.5, GPT-4, LLama2, and Alpaca. Testing results with $LLM_E = LLM_G$ (token probabilities from ChatGPT) is suggested for those with access to it, providing insights into potential performance variations.

Additionally, the third idea involves fine-tuning or adapting the models for specific tasks, especially those with differential performance. Exploring alternative LLM_E models, not in Conditional Generation mode, tailored for tasks like dialogue generation or instruction execution, is also under consideration. In addition, investigating the impact of varying training data amounts and distributions, including supervised and transfer learning, is considered to understand classifier learning patterns and generalizability to other datasets.

Finally, the existing classifiers are notably simple, and there is room for improvement by modifying the architecture of the Simple Neural Network. Attempts to increase complexity yielded similar performance or led to overfitting on the small training data. While this may be effective in scenarios with extensive training data for comparison, implementing a model selection strategy with a validation set could lead to better results.

7 Conclusions

This paper introduces a novel approach to detecting hallucinations in LLMs generations to boost their trustworthiness and applicability in real-world scenarios. Using a method focused on four numerical features based on token probabilities. We exceeded existing standards in areas like Summarization and Question-Answering using the HaluEval benchmark as an experimental playground, highlighting the effectiveness of our technique and potential integration with other approaches.

The contributions of this work include the evaluation of two classifiers, Logistic Regression, and a Simple Neural Network, using numerical features derived from token probabilities. Our work also highlights the importance of each numerical feature in detecting hallucinations for different tasks. Additionally, the research explores the impact of different LLMs, such as GPT-2, BART-CNN, and LED, on the proposed method's performance.

The implications of this research extend to every domain relying on LLMs, including Information Retrieval, Natural Language Generation, and NLP in general. By enhancing the trustworthiness and reliability of LLM outputs, the proposed method contributes to the ethical and responsible use of these models in sensitive applications, such as medical, legal, educational, and financial domains. This work is a big step toward creating a reliable and flexible method to detect hallucinations in LLMs. This paper will help future research and contribute to the larger academic conversation about making trustworthy and capable LLMs.

[7] Showed in the Appendix

Limitations

The first limitation is the numerical features and models selected as LLM_E. While our current approach has demonstrated effectiveness in specific tasks, it may only capture the richness and complexity of some textual content types. The derived features need to be more sufficient for tasks like knowledge-grounded dialogue, which involve intricate context and real-time exchanges.

Our method outperformed state-of-the-art in tasks like summarization and question answering. However, in dialogue and general user queries, it achieved competitive but not leading results. This could hint at potential over-specialization or the need for task-specific feature engineering. Another reason could be the inherent limitations of the LLMs selected as LLM_E. Furthermore, we have yet to test as LLM_E the same model since we cannot access the probabilities from ChatGPT, which generated all the *generated-text* in the HaluEval benchmark. Additionally, because of the context length limitation of some of the LLMs, we needed to truncate the *conditioned-text* to 700 words, which might cause us to lose the necessary context to get the right token probabilities to classify correctly. More experiments can be done with different truncation lengths and also LLMs with higher context lengths.

One of the main limitations is that the results and the effectiveness of our approach may be tied to the characteristics of the dataset used. If the dataset has inherent biases or lacks diversity in certain aspects, the model's performance could be skewed. For instance, it might be in the specific patterns obtained on the HaluEval benchmark that these four numerical features are good indicators for detecting this type of hallucination. However, it doesn't change the fact that current complex state-of-the-art approaches have yet to show this level of performance under the same circumstances.

We need a study separation on the different types of hallucinations per task divided on the HaluEval benchmark, which we intend to do in the next weeks and add to this paper. This analysis will allow us to study which type of hallucinations designed on the HaluEval benchmark are easier or harder to detect with this approach.

Finally, this method is grounded in binary classification. In real-world scenarios, hallucination might be more nuanced, with varying degrees of severity, which our current approach might not account for. Furthermore, there needs to be more interpretability; even when we can get intuition from the numerical features, we cannot obtain the exact explanation of what specific wrong fact or fictitious information is being added. We intend to explore other ideas on datasets that make this separation to increase the interpretability.

References

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Danqi Chen, Jason Bolton, and Christopher D Manning. 2016. A thorough examination of the cnn/daily mail reading comprehension task. *arXiv preprint arXiv:1606.02858*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Souvik Das, Sougata Saha, and Rohini K Srihari. 2023. Diving deep into modes of fact hallucinations in dialogue systems. *arXiv preprint arXiv:2301.04449*.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. Gptscore: Evaluate as you desire. *arXiv preprint arXiv:2302.04166*.

Charlie George and Andreas Stuhlmüller. Factored verification: Detecting and reducing hallucination in summaries of academic papers.

Juncai He, Lin Li, Jinchao Xu, and Chunyue Zheng. 2018. Relu deep neural networks and linear finite elements. *arXiv preprint arXiv:1807.03973*.

Mohammad Hosseini, Catherine A Gao, David M Liebovitz, Alexandre M Carvalho, Faraz S Ahmad, Yuan Luo, Ngan MacDonald, Kristi L Holmes, and Abel Kho. 2023. An exploratory survey about using chatgpt in education, healthcare, and research. *medRxiv*, pages 2023–03.

Zhiqiang Hu, Yihuai Lan, Lei Wang, Wanyu Xu, Ee-Peng Lim, Roy Ka-Wei Lee, Lidong Bing, and Soujanya Poria. 2023. Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models. *arXiv preprint arXiv:2304.01933*.

Yichong Huang, Xiachong Feng, Xiaocheng Feng, and Bing Qin. 2021. The factual inconsistency problem in abstractive text summarization: A survey. *arXiv preprint arXiv:2104.14839*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Structgpt: A general framework for large language model to reason over structured data. *arXiv preprint arXiv:2305.09645*.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. 2023. Challenges and applications of large language models.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Minhyeok Lee. 2023. A mathematical investigation of hallucination and creativity in gpt models. *Mathematics*, 11(10):2320.

Deren Lei, Yaxi Li, Mingyu Wang, Vincent Yun, Emily Ching, Eslam Kamal, et al. 2023. Chain of natural language inference for reducing large language model ungrounded hallucinations. *arXiv preprint arXiv:2310.03951*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

Junyi Li, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. Halueval: A large-scale hallucination evaluation benchmark for large language models. *arXiv e-prints*, pages arXiv–2305.

Sheng Lu, Irina Bigoulaeva, Rachneet Sachdeva, Harish Tayyar Madabushi, and Iryna Gurevych. 2023. Are emergent abilities in large language models just in-context learning? *arXiv preprint arXiv:2309.01809*.

Andrey Malinin and Mark Gales. 2020. Uncertainty estimation in autoregressive structured prediction. *arXiv preprint arXiv:2002.07650*.

Potsawee Manakul, Adian Liusie, and Mark JF Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896*.

Rui Mao, Guanyi Chen, Xulang Zhang, Frank Guerin, and Erik Cambria. 2023. Gpteval: A survey on assessments of chatgpt and gpt-4. *arXiv preprint arXiv:2308.12488*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Dewi Retno Sari Saputro and Purnami Widyaningsih. 2017. Limited memory broyden-fletcher-goldfarb-shanno (l-bfgs) method for the parameter estimation on geographically weighted ordinal logistic regression model (gwolr). In *AIP conference proceedings*, volume 1868. AIP Publishing.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. *arXiv preprint arXiv:2104.07567*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Raymond E Wright. 1995. Logistic regression.

Yijun Xiao and William Yang Wang. 2021. On hallucination and predictive uncertainty in conditional language generation. *arXiv preprint arXiv:2103.15025*.

Junjie Ye, Xuanting Chen, Nuo Xu, Can Zu, Zekai Shao, Shichun Liu, Yuhan Cui, Zeyang Zhou, Chao Gong, Yang Shen, et al. 2023. A comprehensive capability analysis of gpt-3 and gpt-3.5 series models. *arXiv preprint arXiv:2303.10420*.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. *Advances in Neural Information Processing Systems*, 34:27263–27277.

Lai Zeqiang, Zhu Xizhou, Dai Jifeng, Qiao Yu, and Wang Wenhai. 2023. Mini-dalle3: Interactive text to image by prompting large language models. *arXiv preprint arXiv:2310.07653*.

Hanqing Zhang, Haolin Song, Shaoyu Li, Ming Zhou, and Dawei Song. 2022. A survey of controllable text generation using transformer-based pre-trained language models. *ACM Computing Surveys*.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. 2023. Siren's song in the ai ocean: A survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A

survey of large language models. *arXiv preprint arXiv:2303.18223*.

Zheng Zhao, Shay B Cohen, and Bonnie Webber. 2020. Reducing quantity hallucinations in abstractive summarization. *arXiv preprint arXiv:2009.13312*.

Shen Zheng, Jie Huang, and Kevin Chen-Chuan Chang. 2023. Why does chatgpt fall short in answering questions faithfully? *arXiv preprint arXiv:2304.10513*.

A Appendix

A.1 Feature Importance Analysis

We also performed experiments with different combinations of the four numerical features to determine which features were more important or not to get a particular result. Tables 6, 7, 8 showed for each task and model how the results in accuracy were affected by which features were used or not [8]. Once again, this is an average of the three iterations done to all experiments to avoid a lucky random.

Features				Results		
mtp	**avgtp**	**MDVTP**	**MMDVP**	**GPT-2**	**BART**	**LED**
x	**x**	**x**	**x**	**0.94**	0.82	0.78
x				0.5	0.82	0.72
	x			**0.94**	0.65	0.71
		x		0.51	**0.59**	0.53
			x	0.56	0.65	0.52

Table 6: Feature Importance in the Summarization Task using the Accuracy metric for all models.

Features				Results		
mtp	**avgtp**	**MDVTP**	**MMDVP**	**GPT-2**	**BART**	**LED**
x	**x**	**x**	**x**	0.78	**0.94**	0.88
x				0.66	0.57	0.5
	x			0.53	**0.94**	0.84
		x		0.74	0.63	0.67
			x	0.65	0.6	0.56

Table 7: Feature Importance in the Question Answering Task using the Accuracy metric for all models.

As can be observed, even when the combination of features like mtp, $MDVTP$, and $MMDVP$ achieve good results and sometimes even by themselves, it is clear that the main important feature in most cases is the $avgtp$ for most tasks. However, interesting enough, in the case of the Question Answering task, this changed for the `gpt-2-large` model, in which the main feature to obtain its results was $MDVTP$. This suggests

[8]We do not include the rest of the metrics in this table because it overloads it unnecessarily and since the accuracy is a good metric for comparison given that the dataset for these three tasks is balanced.

Features				Results		
mtp	**avgtp**	**MDVTP**	**MMDVP**	**GPT-2**	**BART**	**LED**
x	**x**	**x**	**x**	**0.64**	**0.69**	**0.58**
x				0.5	0.54	0.57
	x			0.54	0.69	0.53
		x		0.63	0.54	0.53
			x	0.54	0.53	0.51

Table 8: Feature Importance in the Knowledge-Grounded Dialogue Task using the Accuracy metric for all models.

that the importance of a given feature is also correlated to the LLM_E used, since for the case of `bart-cnn-large`, the essential feature is $avgtp$.