InterrogateLLM: Zero-Resource Hallucination Detection in LLM-Generated Answers

Yakir Yehuda[*1,2] **Itzik Malkiel**[*1] **Oren Barkan**[3]
Jonathan Weill[4] **Roy Ronen**[1] **Noam Koenigstein**[4]

[1]Microsoft [2]Technion [3] The Open University [4]Tel-Aviv University

Abstract

Despite the many advances of Large Language Models (LLMs) and their unprecedented rapid evolution, their impact and integration into every facet of our daily lives is limited due to various reasons. One critical factor hindering their widespread adoption is the occurrence of hallucinations, where LLMs invent answers that sound realistic, yet drift away from factual truth. In this paper, we present a novel method for detecting hallucinations in large language models, which tackles a critical issue in the adoption of these models in various real-world scenarios. Through extensive evaluations across multiple datasets and LLMs, including Llama-2, we study the hallucination levels of various recent LLMs and demonstrate the effectiveness of our method to automatically detect them. Notably, we observe up to 87% hallucinations for Llama-2 in a specific experiment, where our method achieves a Balanced Accuracy of 81%, all without relying on external knowledge [1].

1 Introduction

Human studies have shown that people tend to be inconsistent when they are not telling the truth (Brewer et al., 1999). As such, a common interrogation technique consists of repeated interviews that attempt to challenge the interviewer's consistency in order to assess its credibility (Granhag and Strömwall, 2001). Truth tellers' answers are well-grounded in their memory, hence, inconsistencies in the respondent's answers are a strong indication of her not telling the truth (Brewer et al., 1999; Dianiska and Meissner, 2023). Inspired by these studies, we present a novel method for hallucination detection in LLMs. Our approach, which we call InterrogateLLM, employs a systematic evaluation of model-generated responses for potential hallucinations by repeating the process of reconstructing a query from its generated answer.

Repeated interviews are a very common and effective verification technique for human interrogations, however, it is not foolproof. In some cases, respondents manage to provide repeated false states that are consistent, while in other cases, truth-tellers may provide inconsistent responses due to memory errors (Bartlett, 1995). In a similar fashion, our method is not flawless; it represents an additional step towards addressing the yet unsolved problem of hallucination detection. Nevertheless, similar to the use of consistency tests in humans, commonly employed for their effectiveness, our method also demonstrates high efficacy.

In recent years, the emergence of LLMs such as GPT-3 (Brown et al., 2020), PaLM (Chowdhery et al., 2022), and LLama (Touvron et al., 2023a,b) has revolutionized natural language processing. These models enable machines to understand and generate human-like text with unprecedented fluency and coherence. Trained on vast amounts of text data, they have demonstrated remarkable capabilities in various applications, from automated content generation to virtual assistants, and beyond. However, their remarkable performance comes with a set of challenges and concerns that need to be addressed for their responsible and effective use. A major concern is the phenomenon of hallucination, whereby these language models generate misleading, potentially harmful, or erroneous text. Hallucination can be characterized by the presence of false information in the output generated by the language model that lacks a factual basis. There are significant challenges associated with the deployment of large language models in real-world applications, especially in those involving critical information or decision-making processes.

Detecting and minimizing hallucinations in LLMs is crucial for ensuring their trustworthiness and reliability, especially in contexts where these models play a pivotal role in communication and

[*]Denotes equal contribution.
[1]Our code, datasets, and task prompts can be found here.

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 9333–9347
August 11-16, 2024 ©2024 Association for Computational Linguistics

decision-making. Existing methods for evaluating model-generated text often rely on surface-level metrics such as fluency and coherence, which may not effectively capture the underlying issue of hallucinations. Therefore, there is a pressing need for a systematic and effective method to detect and mitigate hallucinations in the outputs of these models. Despite its significance, addressing this challenge remains an open issue (Ji et al., 2023).

Our method, InterrogateLLM, operates on the premise that language models exhibiting hallucinations produce inconsistent and incorrect responses to subsequent queries based on the hallucinated information. To identify hallucination in a generated answer, our approach involves prompting the model multiple times to reconstruct the input query using the generated answer. Subsequently, InterrogateLLM quantifies the inconsistency level between the original query and the reconstructed queries. By leveraging the observed inconsistencies, our approach effectively identifies potential instances of hallucination. When a large-language model generates a hallucination, it struggles to consistently reconstruct the original query, leading to variations in responses. This interrogation strategy serves as the cornerstone of our approach for detecting hallucinations in generated answers.

The contributions of our paper are outlined as follows: (1) introduction of the InterrogateLLM method designed for detecting hallucinations in textual answers generated by LLMs. (2) we propose an innovative evaluation approach specifically tailored to the task of hallucination detection, leveraging three datasets associated with our proposed text generation tasks. (3) we investigate the hallucination levels exhibited by recent LLMs, including Llama2, shedding light on their fidelity levels. (4) we present comprehensive performance reports on InterrogateLLM and its variants, conducting a thorough comparison with alternative methods through extensive evaluations.

2 Related Work

Hallucinations have been explored in various natural language generation tasks, including translation, summarization (Kryscinski et al., 2020; Maynez et al., 2020), dialogue generation (Shuster et al., 2021), and question-answering (Lin et al., 2022). This is well-documented in a recent comprehensive survey conducted by (Ji et al., 2023), which provides an insightful overview of hallucinations in diverse natural language generation contexts.

In (Liu et al., 2022), the authors presented a token-level reference-free hallucination detection task along with an additional dataset designed for hallucination detection in free-form text. This dataset consists of textual passages with perturbations, and the objective is to determine whether the entire passage exhibits hallucinations. It is crucial to emphasize that our task differs from their setup, as we specifically address hallucination detection within few-shot prompts involving query-answer sequences.

To address inconsistencies in generated text, SelfCheckGPT, introduced by Manakul et al. (2023b), leverages multiple stochastic samples generated by LLMs using the same query. SelfCheckGPT evaluates the coherence between the response and the stochastic samples by querying the same LLM multiple times. Specifically, it incorporates an additional prompt that includes a stochastic sample and a sentence from the generated text and predicts whether the sentence is supported by the stochastic sample. The approach validates each sentence by conditioning the LLM on each stochastic sample. The methodology of SelfCheckGPT encompasses various approaches, including one based on BERTScore (Fu et al., 2023), and another employing a multiple-choice question answering and generation approach (MQAG) (Manakul et al., 2023a), as well as n-gram and LLM-Prompting. Our method is benchmarked against this baseline, using the last approach in our study.

In recent research, Azaria and Mitchell (2023) proposed a method employing a multilayer perceptron classifier that uses hidden representations from language models to predict sentence truthfulness. However, this approach necessitates labeled data for supervised training and access to the internal states of the language model, which may not always be readily available. In (Kadavath et al., 2022), the authors present a self-evaluation technique where models are trained to predict their knowledge of the answer to any given free-form question. This approach entails prompting the language model to internally assess the accuracy of its previous predictions, including estimating the likelihood that its generated response or answer is correct. It is worth noting that this method requires labeled data for model training, making it a supervised task, which differs from our settings.

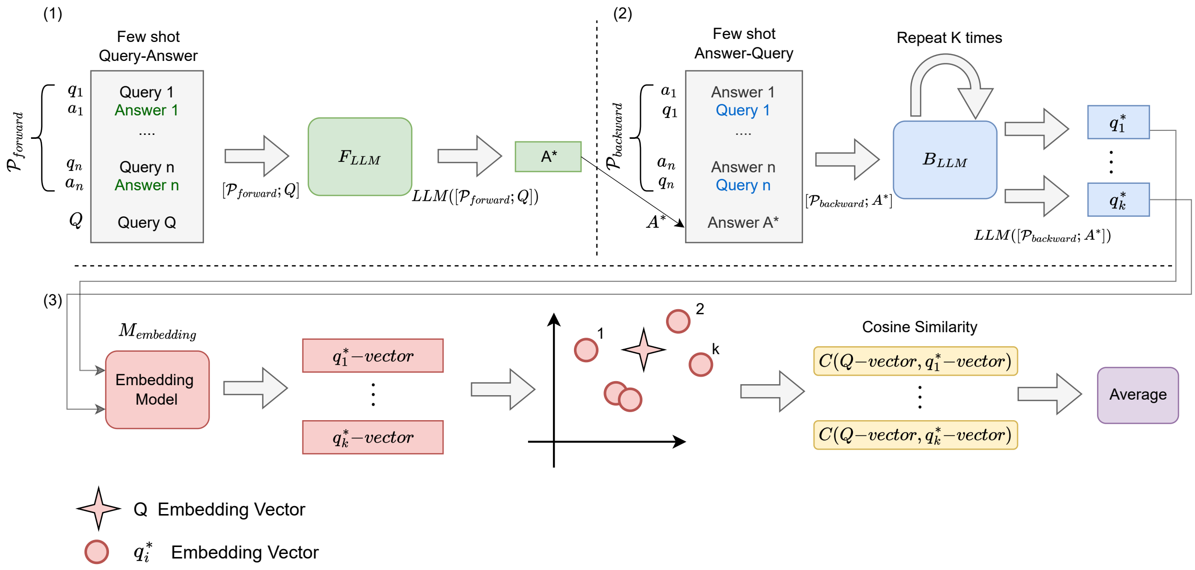

Figure 1: An illustration of the InterrogateLLM method. (1) A few-shot prompt and a query are fed into F_{LLM}, which generates an answer. (2) The shots in the prompt are then reversed, forming a sequence of answer-question pairs, with the generated answer constructed on top. The B_{LLM} is then used to generate K queries that correspond to the generated answer. Ideally, the generated queries should recover the original query from the forward phase. (3) The set of recovered questions is then embedded by a language model and compared with the original question, producing a final score that determines whether the generated answer suffers from hallucination.

3 Problem setup

We assume a source domain of textual queries and a target domain of textual answers. A few-shot prompt[2] (Brown et al., 2020), a corresponding query Q and a LLM denoted by F_{LLM}, are provided. The query is constructed on top of the prompt and fed into the LLM to generate an answer to the query. Our task is to detect whether the generated answer suffers from hallucinations.

The few-shot prompt is constructed as a sequence of query-answer pairs. The pairs are denoted by $\{(q_i, a_i)\}_{i=1}^n$, where q_i represents a query and a_i its corresponding answer. The prompt can be expressed as follows:

$$\mathcal{P}_{forward} = q_1, a_1, \dots, q_n, a_n \tag{1}$$

The F_{LLM} is queried with the concatenation of the query Q on top of the prompt $\mathcal{P}_{forward}$, which retrieves a generated answer denoted by A^*, signifying the response to the query Q. In other words, the prompt $\mathcal{P}_{forward}$ and the query Q are fed into the LLM as follows:

$$A^* = F_{LLM}([\mathcal{P}_{forward}; Q]) \tag{2}$$

[2]While our approach assumes a provided few-shot prompt, it stays adaptable to many zero-shot tasks where the creation of few-shot prompts is feasible.

Our task is to determine whether the generated answer A^* exhibits hallucinatory content.

4 The InterrogateLLM method

In our approach, we introduce a backward process for reconstructing the original query Q from the generated answer A^*. We create a new prompt by reversing the given prompt $\mathcal{P}_{forward}$. The reversed prompt rearranges the order of the query-answer pairs to pairs of answer-query. The reversed prompt, denoted as $\mathcal{P}_{backward}$, can be expressed as follows:

$$\mathcal{P}_{backward} = a_1, q_1, \dots, a_n, q_n \tag{3}$$

The generated answer A^* is then concatenated to the end of the reversed prompt $\mathcal{P}_{backward}$, and the entire sequence is passed either by the same LLM defined above, by a different LLM, or by an ensemble of LLMs. For ease of reference and clarity, we collectively refer to the LLMs involved in this step as B_{LLM}. In other words, in this process, we map the generated answer to the source domain, by querying one or more LLMs, each trying to reconstruct the original query Q. By denoting the set of reconstructed queries as Q^*, this *"backward"* process can be expressed as:

$$Q^* = B_{LLM}([\mathcal{P}_{backward}; A^*]) \tag{4}$$

Algorithm 1: Hallucination Detection

Input: Q, F_{LLM}, B_{LLM}, $M_{embedding}$, $\left\{\left(q^{(i)}, a^{(i)}\right)\right\}_{i=1}^{n}$, τ

Output: True if the generated answer is an hallucination, False otherwise

Step 1: Query Forward Pass
$\mathcal{P}_{forward} = q_1, a_1, \ldots, q_n, a_n$
$A^* = F_{LLM}([\mathcal{P}_{forward}; Q])$

Step 2: Query Reconstruction
$\mathcal{P}_{backward} = a_1, q_1, \ldots, a_n, q_n$
$Q^* = \{\}$
for $i = 1$ **to** K **do**
    Substep 1: Reverse Pass
    $Q^* = Q^* \cup B_{LLM}([\mathcal{P}_{backward}; A^*])$

Step 3: Text Embedding $Q_{vec} = M_{embedding}(Q)$
$Q^*_{vec} = \{M_{embedding}(q^*) : \forall q^* \in Q^*\}$

Step 4: Verification
$\mathrm{sim}(Q, Q^*) = \underset{\forall q^*_{\text{vec}} \in Q^*_{\text{vec}}}{\text{AVG}} ([C(Q_{\text{vec}}, q^*_{\text{vec}})])$
if $sim(Q, Q^*) \geq \tau$ **then**
    return *False*
else
    return *True*

Note that the size of Q^* depends on the number of LLMs used in B_{LLM}.

The motivation for employing a backward process is to reconstruct the original query Q based on the generated answer A^*. If the initial LLM suffers from hallucinations during the generation of A^*, then A^* may drift from the correct answer to Q. Consequently, a backward process operating on A^* is prone to deviating from Q on the way back. In other words, in the case of hallucination in A^*, the set of reconstructed queries Q^* is likely to diverge from the original query Q.

In InterrogateLLM, this backward process is repeated multiple times (K times for each model in B_{LLM}, see Sec. 5.2 for more details), with variable temperature values, as explained below. Therefore,

$$|Q^*| = K * |B_{LLM}|$$

To determine if A^* suffers from hallucination, a language embedding model is utilized to assess the similarity between the set of reconstructed queries Q^* and the original query Q. Both the generated queries and the original query are transformed into vectors within the same embedding space. For a given embedding model $M_{embedding} : text \rightarrow \mathbb{R}^D$, which generates D-dimensional vectors from the input text, the embedding vector for the original query Q is denoted as $Q_{vec} = M_{embedding}(Q)$. Similarly, the embedding vectors for the generated queries are denoted by:

$$Q^*_{vec} = \{M_{embedding}(q^*) : \forall q^* \in Q^*\}$$

Subsequently, the cosine similarity between the embedding vectors of the predicted queries Q^*_{vec} and the original query Q_{vec} is calculated as follows:

$$\mathrm{sim}(Q, Q^*) = \underset{\forall q^*_{\text{vec}} \in Q^*_{\text{vec}}}{\text{AVG}} ([C(Q_{\text{vec}}, q^*_{\text{vec}})]) \quad (5)$$

Here, C represents the cosine similarity function:

$$C(u, v) = \left(\frac{u \cdot v}{\|u\| \|v\|}\right) \quad (6)$$

for $u, v \in \mathbb{R}^D$, where D is the dimension of the vectors. In other words, the cosine similarity is calculated for each q^*_{vec} in the set Q^*_{vec}, and the results are then averaged to obtain the final similarity score.

Finally, InterrogateLLM predicts hallucinations if the similarity score exceeds a predetermined threshold τ. In essence, when the reconstructed queries exhibit a significant divergence from the original query, InterrogateLLM signifies that there is a potential hallucination in A^*. More details about the selection of τ can be found in Sec. 5.2. The InterrogateLLM method is illustrated in Fig. 1, and outlined in Alg. 1.

4.1 Variable temperatures

We introduce an exploratory extension into InterrogateLLM, exploring the impact of diverse temperature values on the accuracy of the detections. In standard LLMs, the temperature parameter influences the likelihood of selecting the next token during the answer generation process. A higher temperature (e.g., 1.0) makes the output more *creative* and *diverse*, while a lower temperature (e.g., 0.2) makes the output more focused and deterministic. Specifically, the temperature is applied through a softmax function that transforms a vector into a probability distribution. In text generation, the softmax function is applied to the model's logit vector, which corresponds to the supported tokens in the vocabulary. The softmax operation can be written as follows:

$$P_i = \frac{e^{z_i/T}}{\sum_{j=1}^{N} e^{z_j/T}} \quad (7)$$

Where P_i is the probability of selecting the i-th token in the vocabulary, z is the logit vector, T is the temperature parameter and N is the number of tokens in the vocabulary. When T is high (low), the exponential function $e^{z_i/T}$ is less (more) sensitive to small differences in the logit values, making the probabilities more diverse (focused).

As complementary experimental explorations, we examine the influence of temperature values on InterrogateLLM during the backward process, which is iterated K times. By introducing dynamic temperature adjustments, our goal is to study the method's accuracy when employed with a range of backward processes exhibiting diverse creativity levels. To this end, we set the temperature for each backward process as follows:

$$T_i = T_0 + \frac{1.0 - T_0}{K} \cdot i \tag{8}$$

where T_i represents the temperature for the i-th backward pass ($0 \leq i < K$), and T_0 is the model default temperature (see Sec. 5.2 for more details).

This temperature scheduling allows for facilitating a controlled ascent in temperatures across the multiple backward processes, promoting enhanced exploration in the space of reconstructed queries. The details and results of this additional study are reported in the experiments, Sec. 5.6.

	Hallucination Rate		
F_{LLM}	Movies	Books	GCI
GPT3	37%	38%	0%
Llama-2 (7B)	87%	66%	25%
Llama-2 (13B)	72%	58%	60%

Table 1: Hallucination rates for each dataset and F_{LLM}.

5 Experiments

To assess the efficacy of InterrogateLLM in detecting hallucinations, and due to the absence of prior datasets for hallucination detection in few-shot prompt settings, we adapted three public datasets. For each dataset, we designed a text generation task along with a verification process to ensure the accuracy of the generated answers. The verification is implemented by employing simple heuristic functions that exploit additional information that is present in the datasets. During the evaluation of hallucination detection methods, the detection predictions are compared against the verification results. Importantly, the InterrogateLLM method operates independently of any external knowledge, making it versatile and applicable to a broad spectrum of tasks.

5.1 Datasets and Tasks

A comprehensive experimental evaluation was conducted using three different datasets to thoroughly evaluate our hallucination detection method across various domains. All three datasets provide a multifaceted evaluation of our technique, revealing its versatility across various types of information and content and allowing us to test the robustness of our hallucination detection method across a wide range of datasets and domains.

5.1.1 The Movies Dataset

The Movies Dataset[3] is a collection of movie-related data that is publicly available for analysis and research. The dataset contains a variety of details about movies that were released before July 2017. The dataset includes 26 million ratings and 750,000 tag applications for all 45,000 movies provided by 270,000 users.

A subset of 3000 samples with movie titles and release years associated with the movie cast was sampled from the Movies dataset. The task is to predict the cast of a movie based on the movie's name and release year. The few-shot prompt contains a few examples mapping a movie's name and release year to its cast. The prompt is in the following format: "Query: What actors played in the x movie y?" where x is the release year and y is the movie name. Cast members' full names are expected in answers, and ground truth labels use Intersection Over Union (IOU) scores, considering any IOU score below 80% as a hallucination.

5.1.2 Books Dataset

The second dataset ("books dataset")[4] is derived from Amazon and includes over 200,000 literary books. This public dataset provides an overview of diverse literary books available on the Amazon platform. Each record includes details like book title, authors, publishers, and publication year.

We sampled a subset of 3,000 samples, including titles, dates, authors, and publishers. The task is to predict the author and publication year based on the book title. The prompts are structured as "Who is the author of the book x, what year was it published?", where x is the book title. The ground truth is established by checking for a match between the elements (author name, release year) in the answer.

[3]The Movies Dataset
[4]Books Dataset

			Movies		Books		GCI	
F_{LLM}	Method		**AUC**	**B-ACC**	**AUC**	**B-ACC**	**AUC**	**B-ACC**
GPT3	$InterrogateLLM$	B_{LLM}: GPT3	0.817	**0.739**	0.709	**0.673**	-	**0.994**
		B_{LLM}: Llama-2 (7B)	0.751	0.639	0.646	0.616	-	0.983
		B_{LLM}: Llama-2 (13B)	0.789	0.695	0.684	0.640	-	0.983
		Ensemble	**0.818**	0.699	**0.710**	0.656	-	0.983
	SBERT-cosine		0.616	0.500	0.534	0.500	-	0.550
	ADA-cosine		0.709	0.500	0.530	0.500	-	0.591
	SelfCheckGPT		0.782	0.684	0.685	0.629	-	0.977
Llama-2 (7B)	$InterrogateLLM$	B_{LLM}: GPT3	0.824	0.786	**0.828**	**0.787**	0.965	0.952
		B_{LLM}: Llama-2 (7B)	0.823	0.750	0.761	0.707	0.959	0.958
		B_{LLM}: Llama-2 (13B)	0.828	0.775	0.795	0.734	**0.969**	**0.960**
		Ensemble	**0.874**	**0.813**	0.822	0.761	0.951	0.948
	SBERT-cosine		0.586	0.516	0.552	0.486	0.957	0.548
	ADA-cosine		0.770	0.501	0.641	0.499	0.950	0.820
	SelfCheckGPT		0.820	0.634	0.784	0.710	0.963	0.927
Llama-2 (13B)	$InterrogateLLM$	B_{LLM}: GPT3	0.806	0.753	0.804	**0.754**	0.989	0.982
		B_{LLM}: Llama-2 (7B)	0.788	0.706	0.742	0.697	**1.000**	**1.000**
		B_{LLM}: Llama-2 (13B)	0.801	0.746	0.771	0.709	0.995	0.991
		Ensemble	**0.842**	**0.773**	**0.807**	0.733	0.992	0.964
	SBERT-cosine		0.539	0.505	0.573	0.497	0.955	0.546
	ADA-cosine		0.728	0.500	0.600	0.500	0.966	0.852
	SelfCheckGPT		0.794	0.689	0.751	0.693	0.934	0.891

Table 2: Hallucination detection results for all models and datasets. InterrogateLLM is reported with $K = 5$ and variable temperature values. For each dataset and F_{LLM}, we compare InterrogateLLM and its variants to all other baselines. As GPT3 does not suffer from hallucinations on the GCI dataset, only the ACC metric is reported (in the B-ACC column).

5.1.3 Global Country Information (GCI)

The "Global Country Information"[5] (GCI) is a public dataset containing information on 181 countries. Detailed information about each country is provided, including its name, land area, capital or major city, GDP, and more. This dataset offers a comprehensive representation of global country information. In the GCI dataset, we concentrate on country and capital pairs. The task involves determining a country's capital by asking, "What is the capital of x?"

Samples from the above three datasets can be found in the supplementary Sec. B. The prompts used in each dataset and the reversed prompts created by InterrogateLLM, can be found in the code[6].

5.2 Implementation details

We set $K = 5$ and $\tau = 0.91$ across all experiments. Maintaining a relatively small value for K facilitates rapid benchmarking of various models on datasets in our evaluations, that encompass tens of thousands of generated answers. The hyperparameter τ was determined through an analysis of ada002 embeddings on a third-party dataset. This involved embedding both similar and dissimilar sentence pairs within the QQP dataset (Chen et al., 2018) and selecting the optimal threshold that effectively distinguished between the two distributions. The initial temperature T_0 was set to the default temperature of each of the evaluated LLMs, specifically 0.6 for GPT3 and both Llama-2 models. The embedding model used in InterrogateLLM leverages the latest OpenAI's model, ada002[7].

In our experiments, we used one A100 GPU. A single application of InterrogateLLM with the full method for $k = 1$, using an ensemble of three models, takes up to 2 seconds. Consequently, benchmarking InterrogateLLM across the three datasets takes up to $\sim$ 3.44 hours. Further insights into the hyperparameters and experimental environment will be detailed in the following subsections.

5.3 Baselines

We compare our method with the following baselines, evaluated on all datasets and F_{LLM} models: **SBERT-cosine:**in this baseline, we employ a pre-trained SBERT model (Reimers and Gurevych,

[5]GCI Dataset
[6]GitHub project
[7]https://platform.openai.com/docs/guides/embeddings/use-cases

2019) to embed both the query and the generated answer. We then calculate the cosine similarity between them and predict "hallucination" if the similarity falls below a threshold $SBERT_\tau$. The threshold was determined by using the same process described in Sec.5.2, this time with SBERT embeddings.

ADA-cosine: similar to SBERT-cosine but employs the recent openAI model ada002. The value of τ used here is consistent with the one in Sec.5.2.

SelfCheckGPT with Prompt: utilizes the same F_{LLM} in each task, SelfCheckGPT generates additional N stochastic LLM response samples, denoted as $S_1, S_2, ..., S_n$, using the same query. Then, it scores the consistency between the generated response and the stochastic samples, by querying an LLM to determine whether the i-th sentence in A^* is supported by the corresponding sample S_i. The final inconsistency score is computed by averaging the sentence scores. In the experiments, this scoring step is evaluated using GPT-3 for all tasks.

5.4 The hallucination rates

We evaluate InterrogateLLM on answers generated by three recent LLMs for each of the datasets and tasks described above. The LLMs we evaluate are: GPT-3 (Brown et al., 2020) and Llama-2 (Touvron et al., 2023b) (7b and 13b models). Interestingly, in Tab. 1 we report the hallucination rates in the generated answers of the three models across the different datasets. Notably, GPT-3 exhibits a lower hallucination rate across all datasets and tasks, compared to the Llama-2 models.

5.5 Hallucination detection results

Binary predictions (hallucinations or not) are compared to the ground truth test labels of each dataset.

For each dataset and task, we employ InterrogateLLM with four different LLM choices for the backward step: GPT-3, Llama-2 (7B), and Llama-2 (13B), either individually or as ensembles of all three models. In Tab. 2, we report the area under the curve (AUC) of the receiver operating characteristic and balanced accuracy (B-ACC) metrics.

As can be seen in the table, the different variants of our method improve upon all the baselines by a sizeable margin. Importantly, we note sizeable improvements also in comparison to SelfCheckGPT. This advantage attributed to InterrogateLLM stems from predicting the query back using the few-shot samples provided in the prompt, a factor entirely overlooked by SelfCheckGPT. Additionally, we observed that in many instances of hallucinations, the stochastic samples generated by SelfCheckGPT also exhibited the same mistake. Therefore, the SelfCheckGPT algorithm erroneously predicted the hallucinated A^* as factual truth. This emphasizes the importance of our unique backward validation strategy, which differs from the query that initially caused the hallucination. Within the variants of the InterrogateLLM method, we observe that the use of an ensemble in the backward process exhibits sizeable strides across the board, suggesting that model diversity can compensate for individual model weaknesses (see also Sec.7).

5.6 Ablation and hyper-parameter analysis

We conduct an ablation study to examine the impact of the multiple K backward processes performed in InterrogateLLM (Alg.1 line 4), the effectiveness of the variable temperature values (Eq.(8)), and the importance of the average function in Eq.5.

Variable K values The performance of InterrogateLLM with various values of K is evaluated on the Movies, Books, and GCI datasets, and the results are reported in Tab. 3, and Tab. 9, 10 from the supplementary, respectively. Specifically, in this study, we evaluate InterrogateLLM with K taking values in the range $[1, ..., 5]$ (higher K values can be considered at the expense of more compute power). The tables reveal that utilizing $K > 1$ in the backward step is crucial in all three experiments. Notably, the best results are consistently obtained with higher K value, where $K = 5$ takes the lead in the majority of cases. Therefore, we hypothesize that increasing the value of K could potentially enhance the results, albeit at the expense of additional computational resources. In addition, we observe that the ensemble of all three models (GPT-3, Llama-2 (7B), and Llama-2 (13B)) yielded the highest performance across all K values. This suggests once again that combining recovery scores from multiple models enhances hallucination detection.

Fig. 2 depicts the enhancements arising from the different values of K, shown for each dataset separately, and reported with both AUC and B-ACC metrics. Each data point represents the average result across all three forward LLMs along with all their corresponding backward LLMs (i.e. the average of each column in tables 3,9 and 10). As can be seen, the data reveals a consistent trend wherein the

F_{LLM}	B_{LLM}	k=1 AUC	k=1 B-ACC	k=2 AUC	k=2 B-ACC	k=3 AUC	k=3 B-ACC	k=4 AUC	k=4 B-ACC	k=5 AUC	k=5 B-ACC
GPT3	GPT3	0.755	0.710	0.773	**0.722**	0.782	0.719	0.786	0.720	**0.790**	0.721
	Llama-2 (7B)	0.701	0.633	0.721	**0.641**	0.727	0.635	0.732	0.638	**0.734**	0.631
	Llama-2 (13B)	0.756	0.688	0.772	0.696	0.779	**0.698**	0.783	0.696	**0.787**	0.697
	Ensemble	0.796	0.690	0.803	0.694	0.811	0.694	0.814	0.695	**0.815**	**0.695**
Llama-2 (7B)	GPT3	0.775	0.774	0.786	0.778	0.788	0.776	0.794	**0.782**	**0.798**	0.780
	Llama-2 (7B)	0.798	0.754	0.815	0.766	0.825	0.757	0.831	0.760	**0.830**	**0.766**
	Llama-2 (13B)	0.810	0.782	0.824	0.778	0.828	0.780	0.836	0.781	**0.838**	**0.783**
	Ensemble	0.840	0.786	0.850	0.787	0.852	0.790	**0.853**	0.792	**0.853**	**0.795**
Llama-2 (13B)	GPT3	0.775	0.752	0.799	0.754	0.808	**0.762**	0.815	0.761	**0.819**	0.760
	Llama-2 (7B)	0.757	0.704	0.763	**0.710**	0.764	0.701	0.767	0.702	**0.769**	0.699
	Llama-2 (13B)	0.770	0.729	0.779	0.731	0.786	0.732	0.789	**0.736**	**0.790**	0.734
	Ensemble	0.819	0.754	0.821	0.758	0.823	0.758	0.823	**0.759**	**0.824**	0.755

Table 3: Results for the Movies dataset. Results are reported for different k values, ranging from 1 to 5, average score. The highest AUC and B-ACC values for each row are presented in bold.

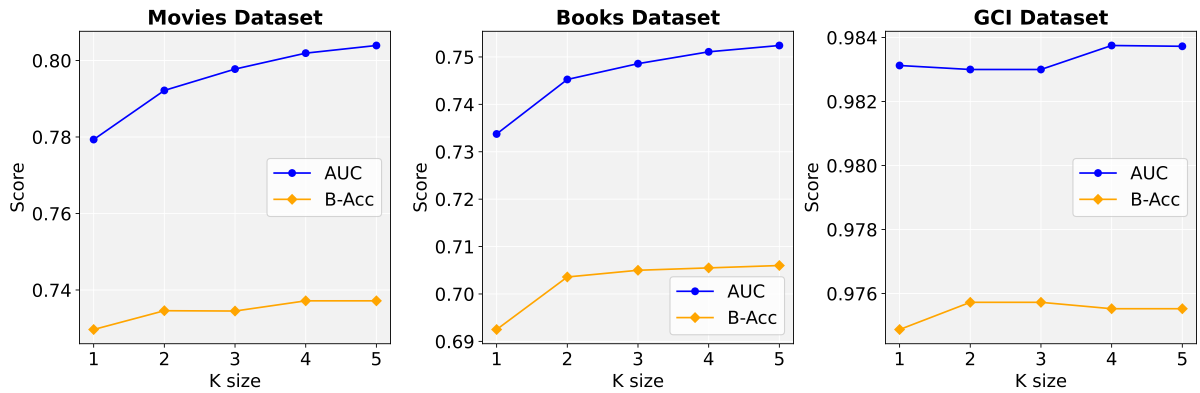

Figure 2: The average AUC and B-Acc scores across Movies, Books, and GCI datasets, per different K values (1-5).

F_{LLM}	B_{LLM}	Same temp AUC	Same temp B-ACC	Variable temp AUC	Variable temp B-ACC
GPT3	GPT3	0.790	0.721	**0.817**	**0.739**
	Llama-2 (7B)	0.734	0.631	**0.751**	**0.639**
	Llama-2 (13B)	0.787	**0.697**	**0.789**	0.695
	Ensemble	0.815	0.695	**0.818**	**0.699**
Llama-2 (7B)	GPT3	0.798	0.780	**0.824**	**0.786**
	Llama-2 (7B)	**0.830**	**0.766**	0.823	0.750
	Llama-2 (13B)	**0.838**	**0.783**	0.828	0.775
	Ensemble	0.853	0.795	**0.874**	**0.813**
Llama-2 (13B)	GPT3	**0.819**	**0.760**	0.806	0.753
	Llama-2 (7B)	0.769	0.699	**0.788**	**0.706**
	Llama-2 (13B)	0.790	0.734	**0.801**	**0.746**
	Ensemble	0.824	0.755	**0.842**	**0.773**
Average		0.803	0.734	**0.813**	**0.739**

Table 4: Results for Movies dataset with same temperature and variable temperature.

cumulative improvements exhibit a proportional relationship with the size of K.

Variable temperatures We extend our investigation to varying temperatures for the backward process. For each index $i \in \text{range}(K)$, the InterrogateLLM method utilizes a variable temperature T_i as defined in Eq.(8). This temperature adjustment aimed to augment the creativity and stochastic aspects of the backward model B_{LLM} throughout the query reconstruction process, fostering the generation of a more diverse set of reconstructed queries. Tab. 4, and Tab. 6, 7 from the supplementary Sec.A.3, present the results of InterrogateLLM with $K = 5$, when using the same temperature through all the backward processes versus using variable temperatures, as proposed in InterrogateLLM. As can be seen, the variable temperature improves the results across most experiments in the Movies datasets, while yielding on-par performance in the Books and GCI datasets (see Tab.6, 7). We hypothesize that the introduction of variable temperature, generating reconstructions with diverse levels of creativity, can be particularly helpful in mitigating instances of mode "collapse", in which certain backward models consistently generate identical reconstructions. In such cases, the incorporation of variable temperature becomes more important. The proposed method utilizes a diverse range of reconstructions. When the vast majority of these diverse reconstructions closely align with the original query, it signifies robust backward processes, better reflecting a non-hallucinated answer and consequently leading to improved accuracy scores. More ablative experiments related to the choices made in Eq. 5 can be found in the sup. Sec.A.1.

6 Conclusion

In this paper, we investigate the pressing issue of hallucinations in large language models. We introduced InterrogateLLM, a novel method designed for detecting hallucinations in few-shot settings. Our work contributes to the ongoing dialogue on the responsible use of AI-powered language models, offering a method that contributes to the reliability of LLM in diverse real-world applications.

As a future work, we would like to extend the method to Retrieval Augmented Generation settings, where a query is provided with a retrieved context, and the task is to generate an answer based on the provided information in the context.

7 Limitations

Throughout our study, we encountered several noteworthy limitations: (1) Source and Target Domain with Many-to-One Mapping: Generated answers associated with multiple different queries pose challenges in verification with InterrogateLLM. The backward process can reconstruct a diverse set of query candidates, deviating from the original query. (2) Hallucinating Back and Forth: Instances were observed where a single backward process by the same LLM model, which hallucinated an answer, could reconstruct the same query. This severe hallucination indicates a symmetric mapping between a query and a hallucinated answer, implying hallucinations in both directions. We observed a mitigation of this issue when employing an ensemble of models. (3) Detecting Hallucinations in Semi-Truth Answers: Identifying hallucinations in semi-truth answers proves more challenging. In some cases, the model only hallucinated a small portion of the answer (e.g., generating an entire cast of a movie with one additional actor not part of the movie). InterrogateLLM was able to recover the original movie, failing to detect the low-severity hallucination within the answer.

References

Amos Azaria and Tom Mitchell. 2023. The internal state of an llm knows when its lying.

Frederic Charles Bartlett. 1995. *Remembering: A study in experimental and social psychology*. Cambridge university press.

Neil Brewer, Rob Potter, Ronald P Fisher, Nigel Bond, and Mary A Luszcz. 1999. Beliefs and data on the relationship between consistency and accuracy of eyewitness testimony. *Applied Cognitive Psychology: The Official Journal of the Society for Applied Research in Memory and Cognition*, 13(4):297–313.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Zihang Chen, Hongbo Zhang, Xiaoji Zhang, and Leqi Zhao. 2018. Quora question pairs.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways.

Rachel E Dianiska and Christian A Meissner. 2023. The effect of credibility assessment techniques on consistency and subsequent memory for the truth. *Frontiers in Psychology*, 14.

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

Pär Anders Granhag and Leif A Strömwall. 2001. Deception detection based on repeated interrogations. *Legal and Criminological Psychology*, 6(1):85–101.

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 Comput. Surv.*, 55(12).

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. 2022. Language models (mostly) know what they know.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Tianyu Liu, Yizhe Zhang, Chris Brockett, Yi Mao, Zhifang Sui, Weizhu Chen, and Bill Dolan. 2022. A token-level reference-free hallucination detection benchmark for free-form text generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6723–6737, Dublin, Ireland. Association for Computational Linguistics.

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. 2023a. Mqag: Multiple-choice question answering and generation for assessing information consistency in summarization.

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. 2023b. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Kurt Shuster, Jack Urbanek, Emily Dinan, Arthur Szlam, and Jason Weston. 2021. Dialogue in the wild: Learning from a deployed role-playing game with humans and bots. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 611–624, Online. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models.

Supplementary Appendices

A More results

A.1 The Average-Max Analysis

For each query, we obtain a list of size K containing cosine similarity scores, representing the similarity between the generated query and the original query. To ascertain the closeness of the generated query to the original, we employ two strategies: the maximum (Max) and the average (Average) of the K cosine similarity scores. Notably, in the majority of cases, the average outperformed the maximum, indicating a more robust measure for query similarity. We attribute this observation to the sensitivity of the maximum strategy to noisy cosine similarity scores, which tend to be prevalent in certain instances. Our analysis suggests that the average is a more reliable indicator of query similarity, providing a stable and effective metric for the evaluation of our reconstruction approach. Full results with avg/max are presented in tables 5, 6, 7, 8, 9 and 10.

A.2 More results for variable K values

Tab. 8, 9 and 10. depicts the performance of InterrogateLLM on the books and GCI datasets, with variable K values. As can be seen, higher values of K result in enhanced accuracy.

A.3 More variable temperature results

Tab. 5, 6 and 7 depicts the performance of the InterrogateLLM method with and without variable temperature values.

B More information about the datasets

B.1 Movies Dataset Query-Answer Examples

We provide query-answer examples from the Movies dataset, where each query seeks information about the cast of a specific movie:

- Query-1: "What actors played in the 1995 movie Jumanji?"
- Answer-1: "The main cast included Robin Williams, Jonathan Hyde, Kirsten Dunst, Bradley Pierce, Bonnie Hunt, Bebe Neuwirth, David Alan Grier, Patricia Clarkson, Adam Hann-Byrd, Laura Bell Bundy, James Handy, Gillian Barber, Brandon Obray, Cyrus Thiedeke, Gary Joseph Thorup, Leonard Zola, Lloyd Berry, Malcolm Stewart, Annabel Kershaw, Darryl Henriques, Robyn Driscoll, Peter Bryant, Sarah Gilson, Florica Vlad, June Lion, Brenda Lockmuller."
- Query-2: "What actors played in the 2011 movie Kingdom Come?"
- Answer-2: "Selma Blair, Edward Burns, Bruce Campbell, Lizzy Caplan, Seymour Cassel, Don Cheadle, Joanne Cook, Rachael Leigh Cook, Tom Cook, Alan Cumming, Tom DiCillo, Drake Doremus."
- Query-3: "What actors played in the 2009 movie Inglourious Basterds?"
- Answer-3: "The actors are Brad Pitt, Diane Kruger, Eli Roth, Mélanie Laurent, Christoph Waltz, Michael Fassbender, Daniel Brühl, Til Schweiger, Gedeon Burkhard, Jacky Ido, B.J. Novak, Omar Doom."

B.2 Books Dataset Query-Answer Examples

Query-Answer examples from the Books dataset, where queries inquire about the author and publication year of specific books:

- Query-1: "Who is the author of the book Classical Mythology, and what year was it published?"
- Answer-1: "The author is Mark P. O. Morford, and it was published in 2002."
- Query-2: "Who is the author of the book Decision in Normandy, and what year was it published?"
- Answer-2: "The author is Carlo D'Este, and it was published in 1991."
- Query-3: "Who is the author of the book Clara Callan, what year was it published?"
- Answer-3: "The author is Richard Bruce Wright, and it was published in 2001 by HarperFlamingo Canada."

B.3 GCI Dataset Query-Answer Examples

Query-Answer examples from the GCI dataset, where each query asks about the capital of a specific country:

- Query-1: "What is the capital of France?"

- Answer-1: "The capital is Paris."
- Query-2: "What is the capital of Japan?"
- Answer-2: "The capital is Tokyo."
- Query-3: "What is the capital of Australia?"
- Answer-3: "The capital is Canberra."

		Same temp		Variable temp	
F_{LLM}	B_{LLM}	**AUC**	**B-ACC**	**AUC**	**B-ACC**
GPT3	GPT3 (avg)	0.790	0.721	**0.817**	**0.739**
	GPT3 (max)	0.768	0.730	**0.787**	**0.752**
	Llama-2 (7B) (avg)	0.734	0.631	**0.751**	**0.639**
	Llama-2 (7B) (max)	0.721	0.669	**0.726**	**0.690**
	Llama-2 (13B) (avg)	0.787	**0.697**	**0.789**	0.695
	Llama-2 (13B) (max)	0.766	0.725	**0.772**	**0.732**
	Ensemble (avg)	0.815	0.695	**0.818**	**0.699**
	Ensemble (max)	0.786	0.741	**0.798**	**0.756**
Llama-2 (7B)	GPT3 (avg)	0.798	0.780	**0.824**	**0.786**
	GPT3 (max)	0.758	0.765	**0.776**	**0.768**
	Llama-2 (7B) (avg)	**0.830**	**0.766**	0.823	0.750
	Llama-2 (7B) (max)	**0.814**	**0.781**	0.808	0.773
	Llama-2 (13B) (avg)	**0.838**	**0.783**	0.828	0.775
	Llama-2 (13B) (max)	**0.821**	**0.791**	0.802	0.780
	Ensemble (avg)	0.853	0.795	**0.874**	**0.813**
	Ensemble (max)	0.802	0.765	**0.810**	**0.772**
Llama-2 (13B)	GPT3 (avg)	**0.819**	**0.760**	0.806	0.753
	GPT3 (max)	**0.777**	**0.755**	0.769	0.748
	Llama-2 (7B) (avg)	0.769	0.699	**0.788**	**0.706**
	Llama-2 (7B) (max)	0.757	0.728	**0.772**	**0.738**
	Llama-2 (13B) (avg)	0.790	0.734	**0.801**	**0.746**
	Llama-2 (13B) (max)	0.770	0.739	**0.777**	**0.748**
	Ensemble (avg)	0.824	0.755	**0.842**	**0.773**
	Ensemble (max)	0.763	0.733	**0.792**	**0.756**
Average (avg)		0.803	0.734	**0.813**	**0.739**
Average (max)		0.775	0.743	**0.782**	**0.751**

Table 5: Results for Movies dataset, presenting results for constant and variable temperature, with both average and maximum scores.

		Same temp		Variable temp	
F_{LLM}	B_{LLM}	**AUC**	**B-ACC**	**AUC**	**B-ACC**
GPT3	GPT3 (avg)	0.698	**0.675**	**0.709**	0.673
	GPT3 (max)	0.685	**0.670**	**0.694**	0.667
	Llama-2 (7B) (avg)	0.640	0.616	**0.646**	0.616
	Llama-2 (7B) (max)	0.615	0.619	**0.632**	**0.625**
	Llama-2 (13B) (avg)	0.675	**0.642**	**0.684**	0.640
	Llama-2 (13B) (max)	0.656	0.643	**0.669**	**0.648**
	Ensemble (avg)	0.707	0.656	**0.710**	0.656
	Ensemble (max)	0.707	0.669	**0.719**	**0.681**
Llama-2 (7B)	GPT3 (avg)	0.821	0.777	**0.828**	**0.787**
	GPT3 (max)	0.811	0.780	**0.815**	**0.784**
	Llama-2 (7B) (avg)	0.761	0.707	0.761	0.707
	Llama-2 (7B) (max)	0.744	0.718	**0.752**	**0.725**
	Llama-2 (13B) (avg)	0.794	0.730	**0.795**	**0.734**
	Llama-2 (13B) (max)	0.783	0.745	**0.785**	**0.752**
	Ensemble (avg)	**0.824**	**0.769**	0.822	0.761
	Ensemble (max)	**0.831**	**0.793**	0.827	0.783
Llama-2 (13B)	GPT3 (avg)	0.799	**0.757**	**0.804**	0.754
	GPT3 (max)	0.792	0.758	**0.797**	**0.763**
	Llama-2 (7B) (avg)	**0.743**	**0.686**	0.742	0.679
	Llama-2 (7B) (max)	0.722	0.696	**0.731**	**0.707**
	Llama-2 (13B) (avg)	0.771	0.707	0.771	**0.709**
	Llama-2 (13B) (max)	0.754	0.714	**0.759**	**0.724**
	Ensemble (avg)	0.802	**0.739**	**0.807**	0.733
	Ensemble (max)	0.808	0.765	**0.817**	**0.774**
Average (avg)		0.752	**0.705**	**0.756**	0.704
Average (max)		0.742	0.714	**0.747**	**0.719**

Table 6: Results for Books dataset, presenting results for constant and variable temperature, with both average and maximum scores.

		Same temp		Variable temp	
F_{LLM}	B_{LLM}	**AUC**	**B-ACC**	**AUC**	**B-ACC**
GPT3	GPT3 (avg)	-	0.994	-	0.994
	GPT3 (max)	-	0.994	-	0.994
	Llama-2 (7B) (avg)	-	0.983	-	0.983
	Llama-2 (7B) (max)	-	0.983	-	0.983
	Llama-2 (13B) (avg)	-	0.983	-	0.983
	Llama-2 (13B) (max)	-	0.983	-	0.983
	Ensemble (avg)	-	0.983	-	0.983
	Ensemble (max)	-	0.983	-	0.983
Llama-2 (7B)	GPT3 (avg)	**0.969**	**0.972**	0.965	0.952
	GPT3 (max)	**0.968**	**0.972**	0.964	0.952
	Llama-2 (7B) (avg)	**0.974**	0.957	0.959	**0.958**
	Llama-2 (7B) (max)	**0.976**	0.961	0.960	**0.962**
	Llama-2 (13B) (avg)	**0.977**	0.959	0.969	**0.960**
	Llama-2 (13B) (max)	**0.977**	0.959	0.971	**0.967**
	Ensemble (avg)	**0.963**	**0.951**	0.951	0.948
	Ensemble (max)	0.944	**0.944**	**0.949**	0.941
Llama-2 (13B)	GPT3 (avg)	0.986	0.982	**0.989**	0.982
	GPT3 (max)	0.971	**0.978**	**0.983**	0.977
	Llama-2 (7B) (avg)	1.000	1.000	1.000	1.000
	Llama-2 (7B) (max)	1.000	1.000	1.000	1.000
	Llama-2 (13B) (avg)	**1.000**	0.991	0.995	0.991
	Llama-2 (13B) (max)	**0.998**	0.978	0.983	**0.986**
	Ensemble (avg)	**1.000**	**0.991**	0.992	0.964
	Ensemble (max)	**0.987**	**0.986**	0.983	0.967
Average (avg)		**0.983**	0.974	0.977	0.974
Average (max)		**0.977**	**0.976**	0.974	0.974

Table 7: Results for GCI dataset, presenting results for constant and variable temperature, with both average and maximum scores.

F_{LLM}	B_{LLM}	k=1		k=2		k=3		k=4		k=5	
		AUC	**B-ACC**	**AUC**	**B-ACC**	**AUC**	**B-ACC**	**AUC**	**B-ACC**	**AUC**	**B-ACC**
GPT3	GPT3 (max)	0.755	0.710	0.765	0.724	0.768	0.730	0.767	0.729	0.768	0.730
	GPT3 (avg)	0.755	0.710	0.773	0.722	0.782	0.719	0.786	0.720	0.790	0.721
	Llama-2 (7B) (max)	0.701	0.633	0.714	0.650	0.714	0.659	0.718	0.664	0.721	0.669
	Llama-2 (7B) (avg)	0.701	0.633	0.721	0.641	0.727	0.635	0.732	0.638	0.734	0.631
	Llama-2 (13B) (max)	0.756	0.688	0.761	0.707	0.764	0.715	0.765	0.721	0.766	0.725
	Llama-2 (13B) (avg)	0.756	0.688	0.772	0.696	0.779	0.698	0.783	0.696	0.787	0.697
	Ensemble (max)	0.782	0.736	0.778	0.742	0.785	0.744	0.787	0.745	0.786	0.741
	Ensemble (avg)	0.796	0.690	0.803	0.694	0.811	0.694	0.814	0.695	0.815	0.695
Llama-2 (7B)	GPT3 (max)	0.775	0.774	0.767	0.775	0.761	0.769	0.758	0.766	0.758	0.765
	GPT3 (avg)	0.775	0.774	0.786	0.778	0.788	0.776	0.794	0.782	0.798	0.780
	Llama-2 (7B) (max)	0.798	0.754	0.808	0.775	0.812	0.778	0.818	0.782	0.814	0.781
	Llama-2 (7B) (avg)	0.798	0.754	0.815	0.766	0.825	0.757	0.831	0.760	0.830	0.766
	Llama-2 (13B) (max)	0.810	0.782	0.812	0.781	0.814	0.785	0.822	0.791	0.821	0.791
	Llama-2 (13B) (avg)	0.810	0.782	0.824	0.778	0.828	0.780	0.836	0.781	0.838	0.783
	Ensemble (max)	0.810	0.782	0.810	0.776	0.810	0.773	0.809	0.770	0.802	0.765
	Ensemble (avg)	0.840	0.786	0.850	0.787	0.852	0.790	0.853	0.792	0.853	0.795
Llama-2 (13B)	GPT3 (max)	0.775	0.752	0.779	0.755	0.775	0.753	0.777	0.752	0.777	0.755
	GPT3 (avg)	0.775	0.752	0.799	0.754	0.808	0.762	0.815	0.716	0.819	0.760
	Llama-2 (7B) (max)	0.757	0.704	0.758	0.717	0.754	0.721	0.753	0.727	0.757	0.728
	Llama-2 (7B) (avg)	0.757	0.704	0.763	0.710	0.764	0.701	0.767	0.702	0.769	0.699
	Llama-2 (13B) (max)	0.770	0.729	0.770	0.732	0.770	0.735	0.770	0.736	0.770	0.739
	Llama-2 (13B) (avg)	0.770	0.729	0.779	0.731	0.786	0.732	0.789	0.736	0.790	0.734
	Ensemble (max)	0.793	0.765	0.777	0.751	0.774	0.743	0.766	0.739	0.763	0.733
	Ensemble (avg)	0.819	0.754	0.821	0.758	0.823	0.758	0.823	0.759	0.824	0.755

Table 8: Evaluation results for the Movies dataset across different k values (1 to 5), with average and maximum scores presented.

F_{LLM}	B_{LLM}	k=1		k=2		k=3		k=4		k=5	
		AUC	**B-ACC**	**AUC**	**B-ACC**	**AUC**	**B-ACC**	**AUC**	**B-ACC**	**AUC**	**B-ACC**
GPT3	GPT3 (max)	0.680	0.657	0.683	0.671	0.681	0.669	0.682	0.669	0.685	0.670
	GPT3 (avg)	0.680	0.657	0.691	0.673	0.692	0.676	0.695	0.681	0.698	0.675
	Llama-2 (7B) (max)	0.626	0.606	0.621	0.615	0.613	0.614	0.610	0.616	0.609	0.617
	Llama-2 (7B) (avg)	0.626	0.606	0.635	0.615	0.633	0.614	0.634	0.614	0.634	0.615
	Llama-2 (13B) (max)	0.654	0.623	0.654	0.634	0.659	0.640	0.660	0.643	0.656	0.643
	Llama-2 (13B) (avg)	0.654	0.623	0.665	0.633	0.670	0.637	0.673	0.638	0.675	0.642
	Ensemble (max)	0.693	0.668	0.696	0.670	0.698	0.668	0.703	0.668	0.707	0.669
	Ensemble (avg)	0.696	0.658	0.703	0.663	0.704	0.657	0.706	0.655	0.707	0.656
Llama-2 (7B)	GPT3 (max)	0.795	0.757	0.804	0.771	0.804	0.773	0.809	0.777	0.811	0.780
	GPT3 (avg)	0.795	0.757	0.811	0.772	0.815	0.773	0.820	0.774	0.821	0.777
	Llama-2 (7B) (max)	0.737	0.686	0.744	0.704	0.746	0.712	0.743	0.714	0.744	0.718
	Llama-2 (7B) (avg)	0.737	0.686	0.754	0.703	0.760	0.708	0.760	0.709	0.761	0.707
	Llama-2 (13B) (max)	0.773	0.720	0.778	0.734	0.779	0.738	0.781	0.741	0.783	0.745
	Llama-2 (13B) (avg)	0.773	0.720	0.785	0.729	0.791	0.732	0.793	0.731	0.794	0.730
	Ensemble (max)	0.806	0.777	0.818	0.787	0.822	0.789	0.827	0.793	0.831	0.793
	Ensemble (avg)	0.811	0.766	0.817	0.768	0.819	0.764	0.822	0.764	0.824	0.769
Llama-2 (13B)	GPT3 (max)	0.776	0.733	0.782	0.745	0.783	0.748	0.788	0.755	0.792	0.758
	GPT3 (avg)	0.776	0.733	0.789	0.750	0.794	0.755	0.797	0.754	0.799	0.757
	Llama-2 (7B) (max)	0.716	0.674	0.721	0.688	0.724	0.695	0.722	0.695	0.722	0.696
	Llama-2 (7B) (avg)	0.716	0.674	0.732	0.689	0.740	0.690	0.743	0.690	0.743	0.686
	Llama-2 (13B) (max)	0.748	0.694	0.749	0.706	0.749	0.709	0.751	0.714	0.754	0.714
	Llama-2 (13B) (avg)	0.748	0.694	0.761	0.707	0.764	0.703	0.769	0.707	0.771	0.707
	Ensemble (max)	0.788	0.752	0.800	0.765	0.803	0.765	0.807	0.766	0.808	0.765
	Ensemble (avg)	0.793	0.736	0.800	0.741	0.801	0.739	0.801	0.737	0.802	0.739

Table 9: Evaluation results for the Books dataset across different k values (1 to 5), with average and maximum scores presented.

F_{LLM}	B_{LLM}	k=1 AUC	k=1 B-ACC	k=2 AUC	k=2 B-ACC	k=3 AUC	k=3 B-ACC	k=4 AUC	k=4 B-ACC	k=5 AUC	k=5 B-ACC
GPT3	GPT3 (max)	-	0.994	-	0.994	-	0.994	-	0.994	-	0.994
	GPT3 (avg)	-	0.994	-	0.994	-	0.994	-	0.994	-	0.994
	Llama-2 (7B) (max)	-	0.983	-	0.983	-	0.983	-	0.983	-	0.983
	Llama-2 (7B) (avg)	-	0.983	-	0.983	-	0.983	-	0.983	-	0.983
	Llama-2 (13B) (max)	-	0.983	-	0.983	-	0.983	-	0.983	-	0.983
	Llama-2 (13B) (avg)	-	0.983	-	0.983	-	0.983	-	0.983	-	0.983
	Ensemble (max)	-	0.983	-	0.983	-	0.983	-	0.983	-	0.983
	Ensemble (avg)	-	0.983	-	0.983	-	0.983	-	0.983	-	0.983
Llama-2 (7B)	GPT3 (max)	0.969	0.968	0.969	0.968	0.969	0.972	0.968	0.972	0.968	0.972
	GPT3 (avg)	0.969	0.968	0.969	0.968	0.969	0.972	0.970	0.972	0.969	0.972
	Llama-2 (7B) (max)	0.975	0.957	0.976	0.961	0.976	0.961	0.976	0.961	0.976	0.961
	Llama-2 (7B) (avg)	0.975	0.957	0.975	0.961	0.974	0.961	0.974	0.957	0.974	0.957
	Llama-2 (13B) (max)	0.977	0.959	0.977	0.959	0.977	0.959	0.977	0.959	0.977	0.959
	Llama-2 (13B) (avg)	0.977	0.959	0.977	0.959	0.977	0.959	0.977	0.959	0.977	0.959
	Ensemble (max)	0.945	0.944	0.944	0.944	0.944	0.944	0.944	0.944	0.944	0.944
	Ensemble (avg)	0.964	0.951	0.963	0.951	0.964	0.951	0.963	0.951	0.963	0.951
Llama-2 (13B)	GPT3 (max)	0.980	0.982	0.980	0.982	0.980	0.982	0.980	0.982	0.971	0.978
	GPT3 (avg)	0.980	0.982	0.980	0.982	0.980	0.982	0.986	0.982	0.986	0.982
	Llama-2 (7B) (max)	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	Llama-2 (7B) (avg)	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	Llama-2 (13B) (max)	1.000	0.991	1.000	0.991	0.998	0.982	0.998	0.982	0.998	0.978
	Llama-2 (13B) (avg)	1.000	0.991	1.000	0.991	1.000	0.991	1.000	0.991	1.000	0.991
	Ensemble (max)	0.987	0.991	0.987	0.991	0.987	0.991	0.987	0.991	0.987	0.986
	Ensemble (avg)	1.000	0.991	1.000	0.995	1.000	0.991	1.000	0.991	1.000	0.991

Table 10: Evaluation results for the GCI dataset across different k values (1 to 5), with average and maximum scores presented.