The Hallucination Detection Challenge: How Large Language Models Fail to Recognise Their Own Fabrications

Anonymous ACL submission

Abstract

Large Language Models (LLMs) which are capable of generating human-like, fluent text, are increasingly involved in the text editing process by humans, leading to a growing number of texts co-authored by LLMs and humans. This raises the question of whether LLMs can assess the factuality of texts co-authored by LLMs and humans. In this paper, we have created a binary dataset composed of texts co-authored by LLMs and humans. The dataset utilizes an automated generation approach, allowing for the easy expansion of the dataset's size, and it features a high degree of similarity between positive and negative examples, which increases the difficulty of model inference on this dataset. After observing that the performance of LLMs on this dataset did not meet expectations, we introduced a confidence score for the output results of LLMs based on their output consistency, thereby significantly enhancing the precision of the model's predictive results.

1 Introduction

The rapid development of LLMs has significantly enhanced their capability to generate coherent and contextually relevant text, establishing them as extremely useful tools for many Natural Language Processing (NLP) tasks. However, LLMs remain prone to generating factual inaccuracies or hallucinated content - textual outputs that deviate from established facts or introduce unverified claims(Friel and Sanyal, 2023). This limitation raises substantial challenges for real-world applications of LLMs in domains requiring stringent factual accuracy, such as news generation, educational tools, and decision support systems.

There is already a large body of work (Dhuliawala et al., 2023) on the detection and mitigation of hallucination phenomena. The ultimate aim is to eliminate hallucinations in LLM-generated responses. To advance this aim, much current work is focused on detecting hallucinations made by LLMs. The detection techniques themselves vary in the extent to which they rely on LLMs. For example, there are now a number of variations based on self-consistency (Wang et al., 2022), where the final answer is chosen as the most frequently occurring response across various samples produced by an LLM. Alternatively, (Dhuliawala et al., 2023) havine the LLM generate the initial question and also prompt the LLM to produce a series of related validation questions. Subsequently, the LLM is tasked with independently answering these validation questions. Based on the outcomes of these answers and the initial response, the LLM then regenerates the final answer.

Employing LLMs to detect hallucinations is of course like getting a student to check their own work. One may expect this not to work well since the LLM carrying out the hallucination detection relies on the same statistical patterns in training data which causes the hallucinations in the first place. Therefore, perhaps surprisingly, Manakul et al. (2023) have shown that this is a promising direction.

However, current hallucination detection datasets predominantly adopt the following approach to dataset generation: 1) collecting factual statements from web sources, 2) preprocessing them into question-answer pairs, and 3) feeding questions to LLMs for alternative answer generation. Consequently, most studies rely on datasets that inadequately capture the subtle distinctions between factual and non-factual statements in real-world scenarios. For example, Lee et al. (2024) introduces the NEC dataset, which requires a model to distinguish the correct responses to questions, misaligned responses generated under misleading conditions, and fabricated responses based on non-existent concepts. In these cases the non-factual content is often very different semantically from factual content, making it fairly straightforward to detect.

This study introduces the NEC dataset, which comprises three distinct categories of LLM-generated responses correct responses to questions, misaligned responses generated under misleading conditions, and fabricated responses based on non-existent fictional concepts.

Everyday experience tells us that hallucinations made by LLMs are often very close to the truth, which is what makes them plausible to the reader. In generation, the substitution of a semantically similar word or insertion of a modifier can be very plausible but lead to the generated sentence having different truth conditions. For example, substituting the word *feline* for *canine* in the sentence "Beethoven is the canine hero from the film series Beethoven, who is pet to the Newton family." would lead to a very plausible but untrue statement.

Therefore, there is a pressing need for more challenging benchmark datasets that better reflect the complexity of real-world hallucination detection tasks, particularly in cases where factual and non-factual statements exhibit high semantic similarity. In response to this research gap, we have designed a method to semi-automatically generate a binary dataset for hallucination detection. We use a large-scale QA dataset as our seed dataset, concatenating the questions and answers to create positive samples with correct facts. We then employ various automatic generation methods to produce incorrect answers, and by concatenating the questions with these automatically generated incorrect answers, we create negative samples i.e., erroneous facts. The dataset generated using this method has a high degree of textual similarity between negative and positive samples because the negative samples are entirely derived from the positive ones, which makes our dataset more challenging.

Subsequently, we focus on the ability of LLMs in detecting hallucinations in text, postulating that a LLM which is better able to detect certain hallucinations is less likely to make similar hallucinations. Further, if a LLM can detect hallucinations in LLM generated text, then a simple method to reduce hallucinations is to set up a pipeline where one LLM generates text and it is approved by another LLM as hallucination-free before returning to the user.

We recognize that the lengthy text in our dataset might contain only a small portion of erroneous factual information, while the majority of the context remains accurate. The LLM might struggle to precisely identify the incorrect information embedded within the correct content, potentially leading to suboptimal performance. Therefore, we investigate the extent to which the LLM performance can be improved by shortening the text length.

We also investigate the extent to which providing contextual evidence helps the LLM in detecting the hallucinated statement. It may be that the statement alone seems plausible enough, but if it is given the correct information as context in the prompt, will it be able to distinguish fact from hallucinated near-fact in this case?

Finally, we adopt a method inspired by self-consistency which uses sampling and aggregation to define the confidence of LLMs in their response. However, rather than requiring a binary decision as to the factuality of the input statement, our method allows the LLM to conclude that it "unknown" if its confidence is low. Thus we investigate the extent to which precision of the LLM's hallucination detection abilities can be increased. If we eliminate cases where there is disagreement in the responses, do we eliminate hallucinations? Or are there cases when the LLM is completely convinced that a non-factual statement is true?

In summary, this paper addresses a number of research gaps by introducing a novel dataset and conducting a comprehensive analysis of the limitations of LLMs in handling such complex hallucination detection challenges. Our contributions are four-fold. First, we provide a binary classification fact-checking dataset co-authored by humans and LLMs. This dataset presents a higher level of difficulty due to the high semantic similarity between positive and negative examples. Second, we demonstrate that the suboptimal performance of the LLM on our dataset is not related to the length of the text data. Even when the LLM processes shorter and more concise texts, its ability to detect hallucinations remains poor. Third, we show that, even when the LLMs are provided with the necessary evidence for reasoning, their performance on our dataset remains suboptimal, highlighting the dataset's inherent challenges. Fourth, we demonstrate that current LLMs perform poorly on our dataset, even when employing techniques like self-consistency, indicating limited effectiveness in handling the dataset's complexity.

2 Related Work

In this section, we survey existing approaches to hallucination detection as the foundational task.

Next, we investigate three potential factors that may influence the LLMs' performance in hallucination detection which will be further analyzed in experiments.

Hallucination Detection Hallucination in NLP was first introduced by Wiseman et al. (2017), referring to phenomena where models generate text containing logical errors or factual errors. Then, many researchers have proposed hallucination detection methods to mitigate the risks.

For example, Chen et al. (2023) trained a small binary classification model using LLM-generated content and human annotations to evaluate the factual accuracy of generated text. Other work requires LLMs to verify their own claims in some way. While, Friel and Sanyal (2023) uses sampling & aggregation. Here, the LLM repeatedly performs binary judgments on whether its own output contains hallucinations and provides reasoning. The percentage of "yes" responses is counted to calculate a hallucination probability score. An alternative approach Dhuliawala et al. (2023) is to generate multiple verification questions about the generated text. The LLM independently answers these questions to check for errors in the original output. Another alternative, InterrogateLLM (Yehuda et al., 2024), is to reconstruct the original query from the generated answer and measure the inconsistency between the reconstructed query and the original query to detect hallucinations. More recently, Zhang et al. (2024) proposed a Self-Alignment-based fact-checking method. This approach breaks down the original LLM output into multiple atomic claims. Then, the model is directed to score the factual accuracy of each claim using its internal knowledge, in order to determine the overall factual correctness of the generated content.

Input Length To our knowledge, the effect of input length on LLM performance at fact-checking and hallucination detection has not been looked at before. However, Levy et al. (2024) evaluated 5 long-context LLMs and found that all models showed clear performance drops in reasoning tasks as input length increased. This happened even when the input was much shorter than their input limitation. In addition, when key paragraphs were placed at the end of the input, the models usually achieved the highest accuracy. This suggests a recency bias in their processing. In Li et al. (2024)'s study, researchers observed that LLMs became significantly worse at understanding task definitions as context length grew.

Retrieval-augmented Generation In order to supplement the limited and potentially out-dated knowledge in LLM training data, retrieval-augmented generation (RAG) have been been proposed which combine the generative power of LLMs with external knowledge bases which provide access newer, broader or more-focussed information. For example, Peng et al. (2023) developed LLM-AUGMENTER, which uses external knowledge and automated feedback to greatly reduce hallucinations in ChatGPT while keeping generated responses fluent and informative. In another study by Quelle and Bovet (2024), providing LLMs with context retrieved from external sources was found to significantly improve their performance in fact-checking tasks. The models also became better at explaining their reasoning. Furthermore, Wang et al. (2024b) showed that retrieval augmentation allows LLMs to use up-to-date information not included in their pre-trained knowledge, making them more practical for real-world applications.

Based on this evidence, one would expect an LLM prompted with the correct information to be better at detecting hallucinations and factual inaccuracies in subsequent information. We test this hypothesis empirically in our work.

Self-consistency It has been seen that the performance of LLMs can often be enhanced by employing a "sample and select" strategy that involves generating multiple samples and evaluating their responses before making a final choice. One widely recognized and effective method within this general approach is Self-Consistency(Wang et al., 2022), which determines the final answer based on the most frequently occurring response across various samples produced by LLMs. In subsequent research, several variations have emerged from the original SC method.

Wang et al. (2024a) introduced Soft Self-Consistency (SOFT-SC), which replaces SC's binary scoring with a continuous score derived from model likelihoods. This modification allows for selection even when the actions are sparsely distributed. SOFT-SC has demonstrated better performance than SC when LLMs generate fewer samples. Chen et al. (2024) proposed Universal Self-Consistency (USC), extending the concept of SC to tasks that involve free-form answers, such as code

generation and summarization. USC leverages the LLMs themselves to identify the most consistent answer. Huang et al. (2023) proposed the Multi-Perspective Self-Consistency (MPSC) framework, which incorporates both inter and intra-consistency across outputs from multiple perspectives. MPSC enhances the coding performance of LLMs by integrating consistency across solutions, specifications, and test cases. It selects the most reliable code by analyzing a tripartite graph that represents these perspectives.

3 Dataset

Since we focus on the factual errors in hallucination, we start by creating a dataset of facts and near-facts which can be considered as truth and hallucination, derived from a QA dataset. Here we use the WikiQA dataset(Yang et al., 2015), but in principle the method could be applied to other QA datasets. WikiQA is a dataset for open-domain question answering. The question is from Bing query logs and the answers are selected sentences from a Wikipedia page. Since the main information in these QAs is general knowledge(Yang et al., 2015), rather than opinions, the answers can be considered as true facts. There are three parts to each item in the WikiQA dataset: Questions, Long Answers and Short Answers. As illustrated in the example in Table 1, both the Short Answer ("Pom Klementieff") and the Long Answer ("Pom Klementieff (born 3 May 1986) is a French actress....") can be considered to answer the Question ("Who played mantis guardians of the galaxy 2"). A Short Answer is the key information inside of the Long Answer, and the remaining part of the Long Answer can be considered as the context of the Short Answer. Actually, the Long Answers are also single sentences extracted from Documents corresponding to the Questions. But we only utilized the Documents in Section 4.2 as evidence.

We build up our own binary classification fact checking dataset based on this WikiQA Dataset. Since the Long Answers can correctly answer the Questions, the combination of Long Answer and Question can be identified as true fact, i.e., a positive sample. For the negative samples, we replace the Short Answers in the Long Answers with Synthetic Short Answers. The synthetic Short Answers differ from the Original Short Answers in semantics, and therefore are unlikely to answer the question correctly. Thus, the combination of Synthetic Long Answer and Question can be identified as a potential False fact, i.e., a candidate negative sample.

Since we want our negative samples to be "near-facts" and appear plausible to an LLM, we generate the Synthetic Short Answers by replacing the Short Answers in Long Answers.

Automatic Generation of the Dataset To generate diverse Synthetic Long Answers, we masked the Short Answer segments in Original Long Answers with <MASK> tokens and implemented multiple automated generation strategies. These included using a masked language model (MLM)(Devlin et al., 2019) to predict tokens for the masked positions, prompting(Radford and Narasimhan, 2018) `GPT-3.5-turbo` to generate contextually appropriate text replacements, and employing in-context learning(Brown et al., 2020) with `GPT-3.5-turbo` by providing example pairs of masked Long Answers and their corresponding synthetic versions. Through multiple strategies, we were able to produce 7 distinct Synthetic Short Answers for each instance.

3.1 Filter dataset

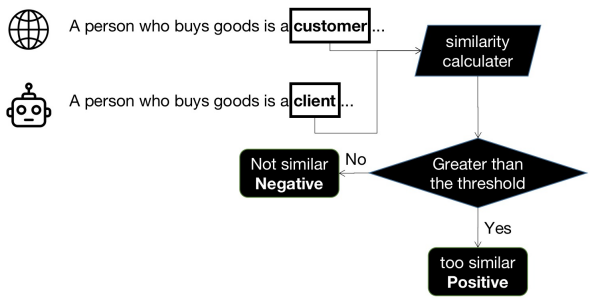

Figure 1: Illustration of negative sample filtering process

One drawback of our automatic generation method is that the generated content is uncontrollable. Thus there is a strong possibility that at least some of the generated Synthetic Short Answers still answer the Questions correctly. As illustrated in Figure 1, we employ a binary classification method that classifies Synthetic Short Answers as positive samples if they can correctly answer the Question, and as negative samples otherwise.

We assume that only the Synthetic Short Answers that have the same semantic meaning as the Original Short Answers can answer the Questions correctly, and hence need to be re-labelled as positive samples. We acknowledge that it is possible

Question	Who played Mantis in Guardians of the Galaxy 2?
Long Answer	**Pom Klementieff** (born 3 May 1986) is a French actress. She was trained at the Cours Florent drama school and has appeared in several films including "Oldboy" and "Guardians of the Galaxy Vol. 2".
Short Answer	**Pom Klementieff**

Table 1: An Example Entry from the WikiQA Dataset

for some questions to have multiple different answers (with different meanings) but these are very rare in practice. Even in contrived examples, a fully correct answer would normally list all of the possible answers. In our initial evaluation of 300 examples, we did not find any where an answer with sufficiently different meaning to the Original Short Answer was also correct.

Consequently, here, we output the representations by sentence transformers (`paraphrase-MiniLM-L6-v2`)(Reimers and Gurevych, 2019) of Synthetic Short Answers and Original Short Answers and calculate the cosine similarity of the two representations. Synthetic Short Answers with cosine similarity above a threshold will be eliminated.

A rational method is required to ascertain the threshold to filter the Synthetic Short Answers. This process can be defined as a binary classification task, where we set the threshold by maximising the F1 score on a sample. We manually annotated 300 data instances, which consist of Synthetic Short Answers and their corresponding Original Short Answers to determine if they share the same semantic meaning. Instances with identical meanings are categorized as positive samples, while those with different meanings are considered negative samples.

To set the threshold we use the Elbow method: Given that the distribution of similarity ranges from 0 to 1, we sequentially set the threshold from 0 to 1, with increments of 0.01. For each threshold, we calculate the corresponding F1 score. Ultimately, the threshold that yields the highest F1 score is selected as the final threshold.

4 Experimental Methodology

The overall aim is to examine the extent to which LLMs can detect their own fabrications, as provided in our dataset. As described in more detail in Sections 4.1 - 4.3, we present each sample independently to the LLM as a binary decision task where it must decide whether the statement is true or false. Furthermore, in Section 4.4, we introduce a method based on sampling and aggregation to define the confidence of LLMs in their output results.

4.1 Performance of LLMs on Hallucination Detection Task

Here, we investigate the ability of currently mainstream LLMs, `GPT-3.5` and `Gemini-2.0`, to distinguish positive and negative samples in our dataset. However, both Long Answers and Short Answers are designed to be plausible answers to the Questions. Hence, our initial experiments concentrated on which of these answers which was provided to the LLM in the prompt in addition to the Question, resulting in two settings: *long* answer and *short* answer.

We note that in the *short* answer setting, the length of the text is shorter and it contains only the key information. This contrasts to the *long* answer setting, where the Short Answer, containing the key information, is surrounded by context. Since the key information in the *short* setting is more obvious, we hypothesised that performance of the LLMs would be higher in this setting.

4.2 Analyzing the Role of Evidence Length

When LLMs fail to determine whether a instance contains hallucinations, one potential reason is their parametric inner knowledge lacks sufficient information for making determination. This raises a critical research question: If this missing information is provided as external knowledge in the input, do LLMs demonstrate sufficient reasoning capabilities to evaluate the instance?

Therefore, we tested the impact of evidence of different lengths on the final results. We used the Original Long Answer from the dataset as 1-sentence evidence. We employed a sentence-transformer to embed every sentence in the document and calculated the semantic similarity between the 1-sentence evidence and all other sentences using cosine-similarity. We selected the sentences with the highest similarity and, accord-

ing to their order of appearance in the document, reassembled them to form 5-sentence.

Since the subject of our statement is the Long Answer or Short Answer, which is almost identical to the 1-sentence evidence, theoretically, the difficulty of 1-sentence evidence is the lowest, and the difficulty increases progressively with the length of the evidence.

4.3 Impact of Generated Positive Samples

In our dataset, the positive samples consist of two categories: original positive samples and generated positive samples. In these experiments, we evaluate LLM performance on both types of positive samples to demonstrate that generated positive samples can serve as an effective augmented data.

4.4 Consistency augmentation

Since large language models (LLMs) underperformed on our dataset, we employed a sampling and aggregation approach to enhance their performance. We conducted two sets of experiments with different sampling strategies. First, we employed Multi-Prompt Sampling, where, for each input instance, we generated 9 responses using 8 semantically similar prompts alongside the original prompt. These variant prompts were produced by a generative model and verified by human reviewers to ensure semantic consistency. Second, we employed Same-Prompt Sampling, where we ran the same prompt 9 times for each input. Due to the randomness in the LLM's generation process, the model produced different outputs each time despite receiving identical inputs(Wiher et al., 2022).

After completing the sampling process, we obtained 9 outputs for each data point in both experimental groups. We first performed aggregation using a voting-based approach. Due to the inherent randomness in the generative model's decoding process, the voting method integrates all 9 outputs, providing a more robust result compared to using a single output. This helps mitigate occasional errors that may occur in individual LLM outputs. We used the voting results as the baseline for this set of experiments.

While the voting method aggregates all 9 outputs, it fails to capture the model's uncertainty when the results are closely split (e.g., 5 True and 4 False). To address this, we introduced a consistency-based threshold to refine the voting outcome.

Specifically, we added an "unknown" label to the voting scheme. For a label to be accepted, it must win the majority among the 9 outputs, and meet or exceed a predefined threshold. For instance, if the outputs contain 5 True and 4 False with a threshold of 6, the final label becomes "unknown", even though True outnumbers False, it lacks sufficient consistency. This threshold quantifies the degree of agreement required for the LLM's output to be deemed confident.

5 Experimental Results

5.1 Performance of LLMs on Hallucination Detection Task

As shown in Table 2, all four experimental groups exhibit high recall but low precision on positive samples, indicating that the LLMs consistently classify many of the data as non-hallucinated (positive) samples. There are much more negative samples than positive samples in our dataset, leading to many negative samples being misclassified as positive. This suggests a systematic bias in LLMs toward predicting samples as positive in our task.

For negative samples, all groups show high precision but low recall, implying that while LLMs are highly conservative in labeling hallucinated content, their predictions for hallucinations are highly reliable when made.

The overall accuracy and F1-scores remain low across all experiments. Even the best-performing setting (GPT+short) achieves only 0.608 accuracy. LLMs struggle with our dataset demonstrating the difficulty of our proposed benchmark.

Compared to GPT, Gemini demonstrates more balanced predictions on the *long* setting dataset. While GPT aggressively classifies most data as positive samples, Gemini shows improved performance with higher accuracy and F1-scores for both positive and negative samples. However, Gemini still exhibits a tendency to over-predict samples as non-hallucinated (positive.

When comparing LLMs' performance on the *long* and *short* settings, we observe distinct patterns. For GPT models, recall for positive samples decreased to 0.602 while recall for negative samples increased to 0.610, achieving balanced performance, which indicates GPT now handles both sample types more equitably, significantly mitigating its previous tendency to over-predict positive samples. Consequently, both accuracy and F1-score for negative samples improved, making this the best-performing setting among the four experiments.

Setting	Acc	Prec	Rec	F1	NegPrec	NegRec	NegF1
GPT+long	0.432	0.294	**0.932**	0.448	**0.923**	0.269	0.416
Gemini-2.0+long	0.530	0.326	0.851	**0.472**	0.897	0.425	0.577
GPT+short	**0.608**	**0.335**	0.602	0.431	0.824	**0.610**	**0.701**
Gemini-2.0+short	0.536	0.310	0.719	0.433	0.838	0.476	0.607

Table 2: Evaluating LLMs on hallucination detection. Average results from Repeated Experiments(3 Runs). "+short" indicates using dataset contains short answers. NegPrec, NegRec and NegF1 represent the precision, recall, and F1-score for the negative class. Standard errors of Acc, F1 and NegF1 all within (0.001 - 0.005)

In contrast, Gemini showed no significant performance differences between dataset settings. This suggests GPT is more sensitive to text length variations, while Gemini maintains stronger inherent robustness in predictions. Although Gemini showed slight improvements in the *short* setting, the differences were not significant.

5.2 Analyzing the Role of Evidence Length

Setting	Evd Length	Acc	F1	NegF1
long	1	0.475	0.454	0.495
	5	0.470	0.453	0.487
	all	0.470	0.449	0.490
short	1	0.543	0.457	0.605
	5	**0.550**	**0.461**	**0.614**
	all	0.544	0.453	0.609

Table 3: Performance of `Gemini-2.0` on evidence-based hallucination detection with varying evidence lengths (Evd Length = number of sentences in evidence).

Table 3 reveals that in both the *long* and *short* settings, the metrics remain stable regardless of variations in evidence length. The best performing configuration in our experiments was the 5-sentence evidence setting within the short group, but its performance advantage over other settings in the same group was marginal (all metric differences < 0.03). In particular, even the theoretically simplest 1-sentence evidence configuration did not show significant performance differences compared to other settings.

This indicates that the model simply lacks the ability to solve the problem we proposed, regardless of the length of the evidence. Moreover, this also shows that the model can maintain consistent results with short texts even under long-text conditions, demonstrating the model's effectiveness in processing long texts.

Setting	Orig Pos Acc	Gen Pos Acc
GPT+long	0.863	0.680
Gemini-2.0+long	0.466	0.330
GPT+short	0.967	0.947
Gemini-2.0+short	0.523	0.464

Table 4: Comparative Performance on Original and Generative Positive Samples.

5.3 Comparison between Original and Generative Positive Sample

The experimental results in Table 4 demonstrate that the performance of LLMs on original positive samples consistently surpasses that on generated positive samples across all experimental settings.

We note that our dataset may not be entirely clean. Although we filtered the data by computing semantic similarity between Original Short Answers and Synthetic Short Answers, a subset of Synthetic Short Answers classified as positive samples may exhibit high semantic similarity to Original Short Answers without conveying the same meaning. For instance, in date-related short answers (e.g.,"March 30th" and "January 1st"), the semantic similarity score might be high despite referring to distinct dates.

However, the performance gap remains marginal. Particularly in the GPT+short experimental group, the accuracy difference is merely 0.02, indicating generated positive samples exhibit high similarity to original positive samples and serve as effective augmented data.

5.4 Consistency augmentation

We first compared the voting results (Table 5) with the single-run performance of Gemini-2.0 (Table 2). The results show that voting, whether using multi-prompt or same-prompt sampling, consistently outperformed single-run outputs, which demonstrates that the voting approach can enhance result robustness by reducing the impact of occasional errors in LLM generations. However, the overall per-

Setting	Threshold	Acc	Prec	Rec	F1	U+ Rate	NegPrec	NegRec	NegF1	U- Rate
multi prompt+voting	–	0.647	0.392	0.782	0.522	–	0.894	0.602	0.720	–
same prompt+voting	–	0.526	0.326	0.866	0.474	–	0.904	0.415	0.569	–
multi prompt+unknown	6	**0.599**	0.413	**0.754**	0.534	0.046	0.894	**0.549**	**0.680**	0.101
	7	0.549	0.431	0.708	0.536	0.117	0.897	0.497	0.640	0.196
	8	0.493	0.468	0.668	0.550	0.189	0.903	0.436	0.588	0.316
	9	0.416	**0.540**	0.611	**0.573**	0.291	**0.916**	0.352	0.508	0.478
same prompt+unknown	6	**0.506**	0.337	**0.855**	0.483	0.022	0.906	**0.392**	**0.547**	0.057
	7	0.483	0.345	0.839	0.489	0.051	0.911	0.367	0.523	0.112
	8	0.453	0.356	0.815	0.495	0.091	**0.916**	0.335	0.491	0.181
	9	0.408	**0.382**	0.787	**0.514**	0.130	0.912	0.284	0.433	0.298

Table 5: Gemini-2.0 Performance on Hallucination Detection: 9-Run Voting Results. Shows performance across different thresholds in unknown experiments. U+ Rate, U- Rate represent the ratio of sample predicted as unknown in all positive samples and negative samples.

formance of voting remained unsatisfactory. This indicates that while voting improves reliability, it does not fundamentally address the core challenge of hallucination detection in LLMs.

As the threshold increases, more instances are predicted as "unknown", with both U+ Rate and U- Rate showing an upward trend. Notably, across all experimental groups, U- Rate consistently remains higher than U+ Rate, typically by a factor of approximately two. This finding demonstrates that LLMs exhibit significantly greater uncertainty when judging negative samples compared to positive ones.

The introduction of the “unknown” label leads to decreases in accuracy, recall, and negative recall. This occurs because some instances that should have been correctly classified are instead categorized as “unknown”, resulting in uncontrolled degradation of these metrics. Meanwhile, precision improves with increasing thresholds. This indicates that under the influence of the “unknown” mechanism, LLMs become more conservative in predicting True labels: some instances that would have been classified as True but with low confidence are now assigned to the unknown category.

From the perspective of F1-score, the F1 on positive samples increases, while the F1 on negative samples decreases. Since the U- Rate is significantly higher than the U+ Rate, the introduction of the “unknown” class enable negative samples more likely to be classified as “unknown” leading to a marked drop in recall for negative samples. Although the precision for negative samples improves, the F1-score (which tends to be closer to the lower value between precision and recall) decreases due to the sharp decline in recall. In contrast, the recall for positive samples remains relatively stable, while their precision increases, resulting in an overall improvement in F1 for positive samples.

Overall, even though part of metrics improved after adding the “unknown” label, LLMs still struggle to reliably complete this task. However, the sampling and aggregation method helps LLMs provide answers more cautiously. For example, during sampling, LLMs generate answers multiple times and use voting to avoid rare errors. In aggregation, when uncertain, LLMs output “unknown” instead of randomly choosing “True” or “False”. However, a significant portion of the data contains hallucinations that LLMs lack sufficient information to detect. Even though sampling and aggregation improves reasoning performance, LLMs still find it hard to judge these cases correctly.

6 Conclusion

In this work, we explore the capabilities of LLMs in performing fact checking tasks on synthetic data. A dataset has been constructed from synthetic data, along with a pipeline for building the dataset, which can easily facilitate the large-scale creation of fact-checking datasets. In our experiments, we verify the limitations of LLMs in this task, thereby highlighting the uniqueness and necessity of the dataset we have proposed. To address the shortcomings of LLMs, we introduce a method that leverages consistency to enhance the precision of LLMs’ predictive outcomes, essentially gauging the confidence of LLMs in their results through the consistency of their multiple output iterations. In terms of results, this approach has effectively improved the accuracy of LLMs in identifying false facts.

7 Limitations

In this study, we exclusively focused on evaluating the hallucination detection capabilities of LLMs on English text, without considering their performance in other languages. Regarding the selection of LLMs for experimentation, due to the rapid development of LLMs, several popular LLMs were not included in our experiments. Additionally, in the automated dataset generation process, we did not fully incorporate all existing generation methodologies. This partial selection approach may result in insufficient diversity within the automatically generated dataset, potentially impacting the generalizability of our experimental findings.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher 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.

Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. 2024. Universal self-consistency for large language models.

Yuyan Chen, Qiang Fu, Yichen Yuan, Zhihao Wen, Ge Fan, Dayiheng Liu, Dongmei Zhang, Zhixu Li, and Yanghua Xiao. 2023. Hallucination detection: Robustly discerning reliable answers in large language models. In *International Conference on Information and Knowledge Management, Proceedings*, pages 245–255. Association for Computing Machinery.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2023. Chain-of-verification reduces hallucination in large language models.

Robert Friel and Atindriyo Sanyal. 2023. Chainpoll: A high efficacy method for llm hallucination detection.

Baizhou Huang, Shuai Lu, Weizhu Chen, Xiaojun Wan, and Nan Duan. 2023. Enhancing large language models in coding through multi-perspective self-consistency.

Seongmin Lee, Hsiang Hsu, and Chun-Fu Chen. 2024. Llm hallucination reasoning with zero-shot knowledge test.

Mosh Levy, Alon Jacoby, and Yoav Goldberg. 2024. Same task, more tokens: the impact of input length on the reasoning performance of large language models.

Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. 2024. Long-context llms struggle with long in-context learning.

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

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, and Jianfeng Gao. 2023. Check your facts and try again: Improving large language models with external knowledge and automated feedback.

Dorian Quelle and Alexandre Bovet. 2024. The perils and promises of fact-checking with large language models. *Frontiers in Artificial Intelligence*, 7.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks.

Han Wang, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. 2024a. Soft self-consistency improves language model agents.

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.

Yuxia Wang, Minghan Wang, Muhammad Arslan Manzoor, Fei Liu, Georgi Georgiev, Rocktim Jyoti Das, and Preslav Nakov. 2024b. Factuality of large language models: A survey.

Gian Wiher, Clara Meister, and Ryan Cotterell. 2022. On decoding strategies for neural text generators.

Sam Wiseman, Stuart M. Shieber, and Alexander M. Rush. 2017. Challenges in data-to-document generation.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018, Lisbon, Portugal. Association for Computational Linguistics.

Yakir Yehuda, Itzik Malkiel, Oren Barkan, Jonathan Weill, Royi Ronen, and Noam Koenigstein. 2024. Interrogatellm: Zero-resource hallucination detection in llm-generated answers.

Xiaoying Zhang, Baolin Peng, Ye Tian, Jingyan Zhou, Lifeng Jin, Linfeng Song, Haitao Mi, and Helen Meng. 2024. Self-alignment for factuality: Mitigating hallucinations in llms via self-evaluation.