HalluCheck: An Efficient & Effective Fact-Based Approach Towards Factual Hallucination Detection Of LLMs Through Self-Consistency

Anonymous EMNLP submission

Abstract

Large language models (LLMs) frequently generate inaccurate responses – this can be particularly dangerous in sensitive areas like medicine and healthcare. Current methods for detecting hallucinations involve sampling answers multiple times, making them computationally intensive. In this study, we introduce HalluCheck, a novel hallucination detection module that identifies factual elements or atomic facts within a text. HalluCheck operates on the premise that responses to questions probing factual answers should be consistent both within a single LLM and across different LLMs. To improve system robustness, we incorporate a token-probability-based double-check mechanism. For hallucinated facts, inconsistencies or a lack of model confidence during generation will be evident. We evaluate our detection module on fact-based datasets such as NQ_Open, HotpotQA, and WebQ, by building upon open-source LLMs such as LLaMa-2 (7B)-Instruct and Mistral-7B-Instruct. Finally, we compare the generated output with the correct answers to determine sentence-level AUC-ROC scores for hallucination detection. Our results demonstrate that HalluCheck can (i) detect hallucinated facts and (ii) achieve significantly higher AUC-ROC scores compared to existing baselines that operate under similar conditions, specifically those that do not utilize external databases for hallucination detection.

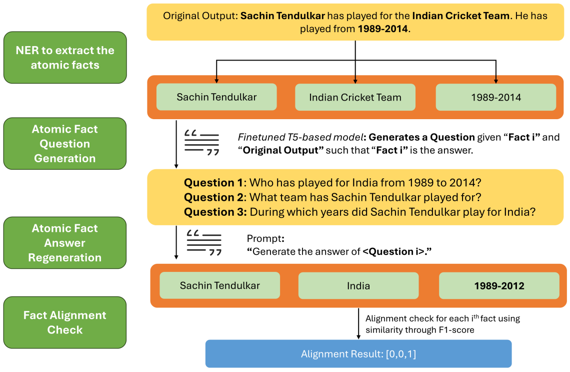

Figure 1: Atomic fact-based hallucination detection through the Fact Alignment check of our pipeline. Each fact is used to generate a question and the fact is regenerated by prompting the question to the LLM.

1 Introduction

Large Language Models (LLMs) (like GPT-4 (OpenAI et al., 2024), PALM (Chowdhery et al., 2022) among others) are well known for their excellent text generation capabilities and are at the forefront of NLP research (Zhao et al., 2023). However, these models often produce information that appears plausible but is actually factually incorrect or nonsensical termed as hallucination (Xu et al., 2024). Studies ((Ji et al., 2023), (Huang et al., 2023)) have categorized hallucination in multiple ways. For example, Ji et al. (2023) state that hallucinations can be of two types, (i) intrinsic (outputs that contradict source information) and (ii) extrinsic (outputs that are left unverified from the source information). Moreover, extrinsic hallucination is approached with caution due to its unverifiable nature, which heightens the risk from a factual safety perspective (Ji et al., 2023). Another way to categorize hallucination is in terms of faithful or factual hallucinations (Huang et al., 2023). Factuality in hallucination highlights the gap between generated content and verifiable real-world facts, usually appearing as factual inconsistencies or fabrications. On the other hand, faithfulness in hallucination refers to the deviation of generated content from the user instructions or the context provided by the input, as well as a lack of self-consistency within the generated content. As studied by Xu et al. (2024), hallucinations in LLMs are inevitable. Therefore, it is imperative to detect hallucinations when they occur, in order to minimize misinformation from reaching the user accessing the LLM.

Several methods for detecting hallucinations have been developed, falling into one or more of these categories. Traditional approaches involve intrinsic uncertainty metrics to identify the parts of the output sequence where the model has the least confidence (Yuan et al., 2021; Fu et al., 2023). However, metrics such as token-probabilities or information about the model's internal parameters might not be available to the user while using closed-source models such as ChatGPT[1], Gemini[2] *etc.* Other approaches include accessing databases to verify the truthfulness of facts (Thorne et al., 2018b; Guo et al., 2022). However, facts can only be evaluated in relation to the knowledge contained within the database. The veracity of the facts outside the database would not be checked. Additionally, some hallucination detection pipelines are restricted to particular tasks such as abstractive summarization (Maynez et al., 2020), machine translation (Dong et al., 2020) among others. More recently, hallucination detection has been performed through sampling-based approaches (Manakul et al., 2023; Mündler et al., 2024). While Manakul et al. (2023) rely on stochastic sampling, Mündler et al. (2024) check for internal-contradiction in the output provided by the model. Given the stochastic nature of the approach, the hallucination metric tends to be inconsistent and, therefore, unreliable. Furthermore, sampling multiple responses is computationally expensive (Manakul et al., 2023) as well. Moreover, internal-consistency addresses only the faithfulness aspect of hallucination and does not account for factual hallucinations.

In this work, we propose a fact-based hallucination detection method for LLMs. The method leverages the LLM's output itself to identify factual inconsistencies without relying on external knowledge sources. It combines the model's internal consistency and confidence scores to assess factuality without requiring repeated sampling of the same response. The approach focuses on capturing factual information within the LLM's response and dynamically regenerates queries based on these factual claims to verify their accuracy. Moreover, the pipeline is customized for each response and does not require any training, making it user-friendly and enhances ease of use. We illustrate the use of our approach via the example in Figure 1, where the steps are highlighted to show how we perform the Fact Alignment check to be able to detect hallucinations of facts in the output at an atomic level. This method is evaluated on an open-domain question-answering (QA) task where inputs to the LLM lack any additional context. Finally, the performance of this approach is compared to existing self-check, self-consistency-based hallucination detection baselines. We conducted our experiments using the NQ Open (Kwiatkowski et al., 2019), HotpotQA (Yang et al., 2018), and WebQA (Berant et al., 2013) datasets, evaluating responses generated by open-source LLMs. As illustrated in Table 2, our method performs comparably to existing hallucination detection baselines while being computationally less demanding. When built on Mistral-7B (Jiang et al., 2023), we surpass other baselines in AUC-ROC scores by 12% on NQ_Open and by 8% on HotpotQA, while providing comparable results on Web Questions. Similarly, for LLaMA2-7B (Touvron et al., 2023) as the choice of our LLM, we exceed other baselines by 7% on NQ_Open and perform on par with them on HotpotQA and Web Questions.

Our primary contribution is HALLUCHECK, a novel hallucination detection module (*c.f.*, Section 3) that is based on the premise that questions probing factual answers should provide consistent responses. This consistency check leverages a token probability-based double-check mechanism. Since HALLUCHECK does not require any training, it can generalize well as evident in our experiments on multiple combinations of datasets and LLMs (*c.f.*, Section 4). We empirically demonstrate across such settings (*c.f.*, Section 5) that HALLUCHECK identifies factual elements or atomic facts within a text with accuracy that is comparable with erstwhile sophisticated approaches. Further, we show that we achieve significantly higher AUC-ROC scores compared to existing baselines that do not utilize external databases for hallucination detection.

2 Related Work

Hallucination in LLMs. Hallucinations are an unwanted phenomenon occurring during text generation by Natural Language Generation (NLG) models. It refers to the erroneous or unfaithful text generated by these models (Ji et al., 2023). Recently, extensive research has been conducted to discuss its principles and challenges (Huang et al., 2023), analysis in various domains such as multimodal LLMs (Bai et al., 2024) and visual models (Liu et al., 2024), detection and mitigation

[1]chat.openai.com
[2]gemini.google.com

techniques, *etc.* (Zhang et al., 2023b; Tonmoy et al., 2024).

Detection of Hallucinations. Zhang et al. (2023a) propose Semantic-Aware Cross-Check Consistency (SAC^3), which is a sampling-based method aimed at addressing hallucinations at the question and model levels, dealing with self-consistency of model generation. Similarly, Manakul et al. (2023) present SelfCheckGPT, another sampling-based detection method for fact-checking LLMs. It uses an LLM to generate stochastically similar outputs and scores the similarity of sampled responses with the original to self-check the LLM's confidence over the original generation. Such self-refining approaches often rely on the target LMs themselves, which is also demonstrated in Self-Refine (Madaan et al., 2023), an iterative mitigation-based approach for hallucinations.

Mündler et al. (2023) analyze self-contradiction in instruction-tuned LMs by employing two separate LLMs for text generation and contradiction analysis for hallucination detection. Their method achieves significant results across various LLMs for their own synthetically LLM-generated text description dataset, providing valuable insights into addressing inconsistencies in the generated text.

Honovich et al. (2022) introduce TRUE, an evaluation of factual consistency measures on pre-existing texts manually annotated for factual consistency. Their study employs a range of metrics, including n-gram-based, model-based, and NLI-based evaluations, conducted on the FEVER dataset (Thorne et al., 2018a). Similarly, among techniques with additional benchmarks, Liu et al. (2022) propose a reference-free, token-level method for detecting hallucinations. The work is supported by a novelly- curated Hallucination Detection dataset (HaDes), with raw web text being perturbed and then annotated by humans to design it for hallucination detection as a classification task.

Finetuning of LLMs is another aspect that can improve hallucination detection and factual output generation. Tian et al. (2023) propose a simple method for optimizing language models in long-form text generation without human annotation for improving the factuality of LLMs. They demonstrate how learning from automatically produced factuality preference rankings—created using their method or by using current retrieval systems—significantly increases the factuality.

A few other detection approaches deal with internal state analysis in LLMs. Azaria and Mitchell (2023) suggest a method to assess the veracity of outputs and detect hallucinations by passing the internal states/activations of an LLM through a trained classifier to output its probabilities of truthfulness. Similarly, some algorithms such as Decoding by Contrasting Layers (Chuang et al., 2023) are developed to handle differences between output token probabilities in the final states or hidden intermediate states of LLMs for detecting hallucinations while also proposing it further as a mitigation strategy. Shi et al. (2023) propose a similar decoding strategy that appends question-based inputs with external context and then deals with the output token probability differences for detection and, subsequently, mitigation. These approaches do not specifically deal with the contextual information in the inputs, which are utilized by other detection approaches to aid in dealing with factual information. Context, in several cases, plays out as a major factor in improving hallucination detection baselines.

3 HalluCheck

Our proposed method, HALLUCHECK, aims to tackle the occurrence of factual hallucinations in Large Language Models (LLMs). HALLUCHECK identifies hallucinations through the utilization of only the text provided for which hallucination has to be detected.

To check whether a piece of text, $\mathcal{A}$, generated by an LLM $\mathcal{M}$ is hallucinated, we start with the assumption that the generated text is correct. We then generate questions that can be answered based on the information in $\mathcal{A}$. Subsequently, we employ the LLM to answer the questions and see if the answers match the information in $\mathcal{A}$, a mismatch indicating hallucinations. The initial step is to identify the factual components within a sentence. According to Kai et al. (2024), factual information in a sentence is typically conveyed through specific parts of speech, *viz.*, nouns, pronouns, cardinal numbers, and adjectives. This information can be extracted by performing part-of-speech (POS) tagging on the sentence. Mathematically, given $\mathcal{A}$, we perform coreferencing and decompose $\mathcal{A}$ into sentences $S_1, S_2, \ldots, S_N$, where N is the total number of sentences, such that $\sum_{i=1}^{N} S_i = \mathcal{A}$. Each sentence is tagged to extract atomic facts

a_{ij}, where $i \in \{1, \ldots, N\}$ and j depends on the number of tagged entities in a sentence. The tagging can be either POS-based or NER-based, as discussed in Section 6.1.3. For example, given the original sentence "*Sachin Tendulkar has played for the Indian Cricket Team. He has played from 1989-2014.*", in Figure 1 the atomic facts consist of $a = [a_{11} = \text{Sachin Tendulkar}, a_{12} = \text{Indian Cricket Team}, a_{21} = \text{1989-2014}]$.

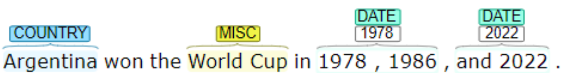

Figure 2: NER tagged sentence. As can be seen, the atomic facts required in the sentence are Argentina, the World Cup, and the years (1978, 1986, and 2022)

After identifying the atomic facts, the next step involves verifying whether each fact is hallucinated within the context of the sentence. Unlike previous methodologies that assign a hallucination score to each sentence, HALLUCHECK focuses on atomic facts, thereby enhancing explainability by pinpointing the exact parts of a sentence that are hallucinated and providing reasons for this determination, as detailed in Section **??**. Specifically, for each atomic fact a_{ij} given sentence S_i, a corresponding question q_{ij} is generated (using a T5-based fine-tuned model), with a_{ij} as the target answer and S_i as the context, expressed as $q_{ij} = \mathcal{Q}(a_{ij}|S_i)$, where $\mathcal{Q}$ represents the question generation module. In Figure 1 each atomic fact provides one question $q = [q_{11} = \text{Question 1}, q_{12} = \text{Question 2}, q_{13} = \text{Question 3}]$. These questions are then evaluated by the LLM $\mathcal{M}'$ at a low temperature to ensure response consistency (refer to Section **??**). Note that $\mathcal{M}'$ may not be the same as $\mathcal{M}$ as detailed in section 6.1.2.

The responses from $\mathcal{M}'$ yield regenerated facts f_{ij}, which are subsequently checked for consistency with a_{ij}. The f for figure 1 being $f = [f_{11} = \text{Sachin Tendulkar}, f_{12} = \text{India}, f_{21} = \text{1989-2012}]$. It should also be noted that the number of atomic facts varies per sentence based on factual content present per sentence. Therefore, the number of questions generated also varies. Loosely speaking, this approach allows us to break down the information in a sentence into discrete elements. This approach assumes that the LLM's answers will be consistent for factual information when sampled at a low temperature.

If f_{ij} and a_{ij} are not consistent (as is the case of f_{21} in figure 1), then a_{ij} is tagged as hallucinated. In the second scenario, while generating f_{ij}, we also record the probabilities associated with its generation. Given that f_{ij} and a_{ij} are consistent (f_{11}, f_{12} in figure 1), we hypothesize that the probabilities used to generate f_{ij} can serve as a proxy for a_{ij}. Let p_{ij} refer to the token probabilities of f_{ij}. For each f_{ij}, a Kolmogorov–Smirnov test is performed between the top-5 tokens (heuristically chosen) to check whether f_{ij} has indeed been sampled from a non-uniform distribution. If it is indeed sampled from a non-uniform distribution, then a_{ij} is tagged as non-hallucinated; otherwise, hallucinated.

The final hallucination score for a sentence S_i is calculated by averaging the individual scores of a_{ij} present in it to give a probability of how likely a sentence has been hallucinated.

4 Task and Datasets

Open-domain Question Answering. Open-domain question answering (QA) is a task where large language models (LLMs) are particularly susceptible to factual hallucinations, especially when no external context or information is provided for the input questions. In such scenarios, if the LLM lacks the correct information within its parameters and pretraining data for the specific inputs, it is likely to generate factually inaccurate answers, resulting in hallucinations.

Datasets. To evaluate our approach, we utilize three publicly available datasets curated for open-domain QA tasks. These tasks are designed to answer factual questions from a large knowledge corpus without providing any explicit evidence.

- **Natural Questions (NQ)-open dataset** (Kwiatkowski et al., 2019): The NQ-Open task, introduced by Lee et al. (2019), is an open-domain question-answering benchmark derived from the Natural Questions dataset. Its objective is to generate an English answer string in response to an English input question, with all questions answerable using content from English Wikipedia. The validation split of this dataset comprises 3,610 samples featuring open-domain questions (unsupported by any explicit evidence) across a wide range of topics, along with their factual answers. We use these questions as inputs for the LLM to

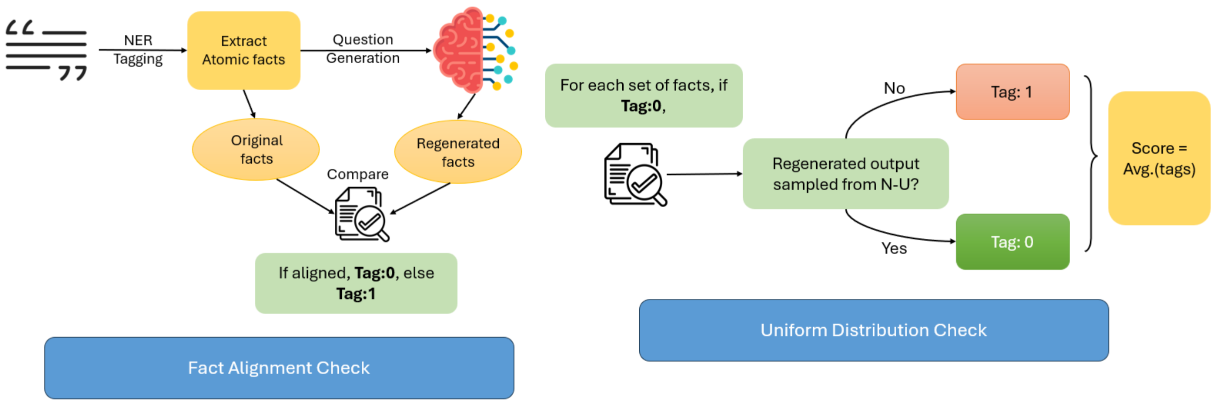

Figure 3: Pipeline of the HALLUCHECK approach, with NER tagging of outputs followed by the comparison-based Fact Alignment check and additional probability-based check, for tagging hallucinations.

generate answers, upon which our detection approach will be applied.

- **HotpotQA** (Yang et al., 2018): HotpotQA is a question-answering dataset that features natural, multi-hop questions and provides strong supervision for supporting facts. Due to the nature of the dataset, the LLM responses necessitate multiple hops, resulting in the generation of numerous facts. Consequently, verifying the correctness of each generated fact becomes essential. For our experimentation, we employ the validation split of this dataset, similar to NQ-open, which contains 7,405 question samples. Therefore, responses were generated from both LLaMA2-7B and Mistral-7B models to check for hallucinations.

- **Web Questions** (Berant et al., 2013): This dataset comprises 6,642 question/answer pairs, with questions designed to be answerable using Freebase, a comprehensive knowledge graph. The questions predominantly focus on a single named entity. For our experimentation, we use the test set, which comprises approximately 2,000 samples. As this dataset contains samples without context, it is specifically used for open-domain QA.

5 Experiments

Models Used. The generative LLMs used to generate responses for our dataset are Mistral-7B-Instruct (Jiang et al., 2023) and LLaMA2-7B (Touvron et al., 2023), which are state-of-the-art open-source models at the time of dataset creation. To obtain the responses, we set the temperature

Model Name	% Atomic Facts/Output		
	NQ-Open	HotpotQA	Web Questions
Mistral-7B	27.53%	13.10%	21.61%
LLaMA2-7B	10.23%	10.22%	8.4%

Table 1: Factuality in generated outputs, highlighted by the percentage of average atomic facts per total generated tokens for each of the samples in the three datasets.

to 0.0. Our primary focus is on utilizing LLMs that are robust in text generation and have been pretrained on extensive datasets, enabling them to perform well on open-domain question-answering tasks in settings without external context.

Experimentation details. The models utilized are open-source, with their associated model weights accessible for inference via the Huggingface web platform[3]. Baseline implementations utilized identical models (Mistral-7B and LLaMA2-7B) and datasets to present comparative outcomes. For SAC^3 (Zhang et al., 2023a), we compute the question-level consistency SAC^3-Q score and employ predetermined thresholds to discern the presence of hallucinated outputs. The other baselines are used in the same setting for the evaluation on the datasets. Our methodology involves assessing diverse decoding methodologies (see Section 6.1.1). Additionally, we integrate outcomes from experiments employing the verifier LLM LLaMA2-7B-Inst (Touvron et al., 2023) as a complementary measure to identify hallucinatory content in the responses generated by Mistral-7B.
Metrics for Analysis. The experiments using the baselines and our approach are analyzed as binary

[3]https://huggingface.co/models

classification tasks for hallucination detection, to classify the original output text generated by the LLM for each instance in the datasets. We compare the baselines with our approach (see Table 2) and report the AUC-ROC and Average Precision scores on the three datasets used for open-domain Question-Answering. AUC-ROC accounts for both the True Positive and True Negative Rates, providing a balanced view of the model's ability to distinguish between the two classes. Average Precision is particularly useful in such a hallucination detection task, where the positive class (i.e., hallucinated text) is more important as it emphasizes performance on the positive class, especially in imbalanced datasets.

6 Results

We test our pipeline on factual datasets mentioned in Section 4. The results for the classification of hallucinated texts have been formulated in Table 2. We see that our models outperform the current best self-consistency-based hallucination detection frameworks in the NQ_Open dataset. For the HotpotQA dataset, HaDeS has slightly better Avg. precision. Overall, Alignment solely is a strong signal for detecting hallucinations occurring in the models. Fact Alignment Check (w/ Greedy Decoding): This baseline refers to only checking the consistency of the model, ignoring the confidence on which the regenerated facts were generated from. This method is completely black-box taking into account none of the model's internal parameters either during the original generation of the answer or during regeneration. This model is the best overall, giving competitive precision results compared to other hallucination detection frameworks.

6.1 Study of different parameters utilized in the pipeline

6.1.1 Decoding strategies

Regardless of how the original response, subject to hallucination assessment, was generated, we examine the variations in regenerated factual responses when decoding strategies are varied. The following decoding strategies were utilized:

1. **Greedy Decoding**: Greedy decoding involves selecting the token from the vocabulary V with the highest conditional probability. This suggests prioritizing atomic facts for which the model has the highest immediate confidence.
2. **Beam Decoding**: Beam decoding represents an enhancement over greedy decoding. In Beam decoding, a parameter known as beam_size determines the number of tokens with the highest conditional probabilities considered at each time step t. For our experiments, we considered the beam size to be 5.

Greedy decoding improves the detection of hallucinations during fact regeneration compared to beam search. This advantage likely arises because greedy decoding prioritizes immediate model confidence. Consequently, decoding strategies that improve the factuality of the models are likely to do better in the pipeline (Li et al., 2023). As a result, when generating atomic facts, it maximizes confidence at each step as can be seen in Table 3. This is further corroborated by the findings of Lee et al. (2023), which indicate that greedy decoding is more factual. Greedy decoding selects the word with the highest probability, thereby minimizing randomness and maximizing the utilization of the language model's parametric knowledge. However, this decoding strategy does sacrifice generation diversity and quality.

6.1.2 Evaluator LLMs

We focus on model-level self-consistency as examined by Zhang et al. (2023a), employing different models of approximately the same size to generate responses for the datasets. This cross-verification uses different LLMs to leverage their diverse knowledge bases for the same factual query. Since the questions probe for factual knowledge, any deviation between the original fact and the regenerated fact indicates hallucination due to the lack of consistency between the models' answers. However, as shown in Table 2, cross-evaluation performs comparably to fact alignment, suggesting that alignment with the same LLM is preferable to using a different LLM. The original LLM used for generating responses was Mistral-7B, while the verifier LLM was LLaMA2-7B.

6.1.3 Tagging of atomic-facts

Kai et al. (2024) suggests that factual information in a sentence can be identified using POS tagging. In our pipeline, we also incorporate NER tagging, as it identifies tags that contain the most factual information, specifically 'NNP' or 'NNPS'. We selected the tags 'NNP', 'NNPS', 'CD', and 'RB' to be considered atomic facts. Additionally, we sampled random tokens from the sentence, ensuring

Model	NQ Open				HotpotQA				WebQA			
	Mistral-7B		LLaMA2-7B		Mistral-7B		LLaMA2-7B		Mistral-7B		LLaMA2-7B	
	AUC-ROC	AP	AUC-ROC	AP	AUC-ROC	AP	AUC-ROC	AP	AUC-ROC	AP	AUC-ROC	AP
SelfCheckGPT (Manakul et al., 2023)	0.46	0.79	0.52	0.88	0.54	0.83	0.51	0.82	**0.72**	**0.89**	0.54	0.83
SAC^3 (Zhang et al., 2023a)	0.54	0.83	0.56	0.89	0.53	0.83	0.54	0.82	0.51	0.78	0.51	0.82
HaDes (Liu et al., 2022)	0.55	0.84	0.49	**0.92**	0.48	**0.92**	0.51	**0.85**	0.56	0.86	0.58	**0.88**
HALLUCHECK (Fact alignment)	**0.67**	**0.88**	**0.63**	0.91	**0.56**	0.84	**0.54**	0.83	0.67	0.86	**0.61**	0.85
HALLUCHECK (Cross Eval)	0.61	0.85	0.56	0.89	0.51	0.82	0.53	0.83	0.65	0.84	0.6	0.85

Table 2: Model Evaluation Metrics for NQ Open, HotpotQA, and Web Questions. AP refers to the average precision obtained while varying the threshold. We compare HalluCheck in the same settings as the baselines to report the results, with Mistral-7B-Inst and LLaMa2-7B-Inst as the base models. Results for HalluCheck are provided when using Fact Alignment check, and where LLaMa2-7B-Inst and correspondingly Mistral-7B-Inst are used as Cross evaluator models.

Model	Decoding Method	NQ Open	HotpotQA	WebQA
Mistral-7B	Greedy	**0.64**	**0.53**	**0.66**
Mistral-7B	Beam	0.56	0.53	0.60
LLaMA2-7B	Greedy	**0.54**	**0.51**	0.51
LLaMA2-7B	Beam	0.53	0.51	**0.52**

Table 3: The AUC-ROC scores of Mistral and LLaMA models using different decoding strategies for fact regeneration on three datasets.

the number of sampled tokens equaled the number of NER tags present. Our results show that NER outperforms both POS tagging and random token sampling in identifying which tokens contribute to the factuality of a sentence or paragraph.

Tagging	NQ Open	HotpotQA	WebQA
NER	**0.67**	**0.56**	**0.67**
POS	0.62	0.52	0.61
Random	0.58	0.56	0.49

Table 4: The AUC-ROC scores of Mistral models using different tagging strategies for identifying atomic facts in the sentence.

6.1.4 Effects of changing threshold

For additional evaluation, we use threshold-based analysis to classify the averaged scores of each sample, i.e. for different thresholds between 0 and 1, we classify the output as hallucinated if the score lies above the threshold. We use this to plot precision values in Figure 4 for the different settings of our approach. The results indicate a gradual increase for each of the settings on all 3 datasets as the threshold increases between 0 and 1. We observe that Fact Alignment performs consistently better than the other settings, indicating that alignment without probability check performs better in hallucination detection.

6.2 Comparison with Baselines

We compare the results obtained when testing the baselines on the datasets with those obtained from our experimented approaches. Our methods (with Greedy and Beam decoding or LLaMa2-7B as a cross evaluator) outperform the baselines, primarily because our approach relies directly on a factuality-based check whether the target LLM contains the correct factual information in its original outputs. As opposed to this, the baselines tend to use stochastic sampling-based approaches, which do not directly compare the pinpointed facts in the outputs to regenerated answers, and hence our approach performs well on these open-domain QA datasets where generated outputs are concise and compact. In such cases, pinpointed fact-checking is a simpler and more direct way of detecting hallucinations.

7 Strengths

Consistent Scoring of Samples. In contrast to previous stochastic methods for hallucination detection (Manakul et al., 2023), our approach does not depend on the randomness or multiple outputs of the LLM. Consequently, our scores remain consistent across multiple runs of the same sample. Furthermore, our method avoids generating multiple responses from the same LLM for the same , instead concentrating on extracting diverse facts from sentences. This results in lower computational overhead compared to previous approaches.

Explainability of Scores. We provide fact-level scoring, enabling users to discern which specific facts are hallucinated and which are not. Furthermore, because our pipeline operates on fine-grained facts rather than entire sentences, we provide a level of explainability absent in previous approaches such as Zhang et al. (2023a), clarifying the rea-

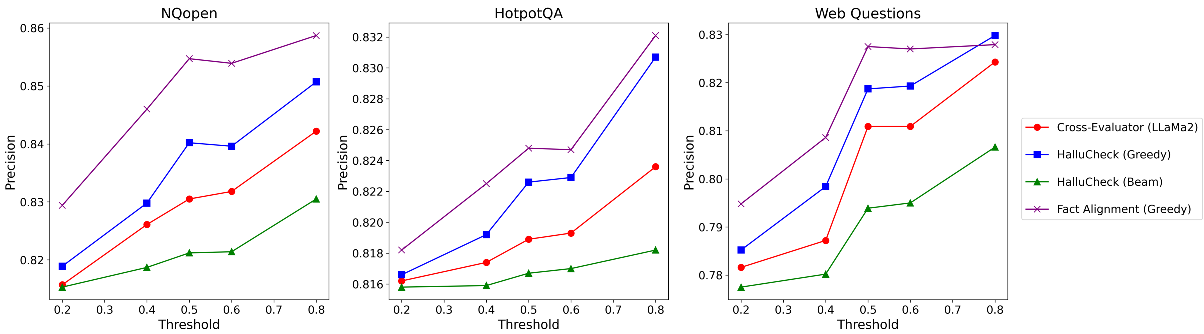

Figure 4: Precision values at varying thresholds (ranging from 0.2 to 0.8) of hallucination classification on generated outputs of the three datasets by Mistral-7B, for each of our HALLUCHECK-based experiments. We observe that the Fact Alignment Greedy significantly outperforms others across all thresholds.

sons behind tagging a sentence as hallucinated.

No Training Required. Our approach operates on a general framework applicable to any sentence, rather than being tailored to specific datasets or sentence types. The number of generated questions is determined by the factual content within the sentence, rather than being selected heuristically. Moreover, our approach provides users with a rapid assessment of whether the information generated by the LLM is factual.

Ease of use. Our approach does not rely on access to any API and requires only the model's output (which needs to be verified) and the LLM used to generate the response. This allows our method to function on the same device where the response was generated, whether through a web interface, API or a model running on a local device.

8 Conclusions

In this work, we propose HALLUCHECK, a novel fact-based hallucination detection pipeline, and compare it to existing approaches. We evaluate our method on three factuality measurement datasets, providing the first comparison between existing factual hallucination detection methods. Our findings reveal that despite being less computationally expensive and not requiring any training, our method performs on par with other approaches.

9 Limitations

We also analyze scenarios where our model fails to correctly categorize the output. Take for example the following question from HotpotQA:

Question: Which of the office buildings used to staff the White House used to be known as the State, War, and Navy Building?

Answer provided by Mistral-7B-Instruct: The office building used to staff the White House that was once known as the State, War, and Navy Building is now known as the **Eisenhower Executive Office Building**. This building was constructed in 1952 and was named after President Dwight D. Eisenhower. It was originally built to house the executive branch of the U.S. government, including the White House staff. The building was designed by architects Cass Gilbert and James A. Moore.

We make the following observations:

1. Although the Eisenhower Executive Office Building is factually correct, our pipeline categorizes the entire paragraph as hallucinated. This discrepancy arises because our model identifies the fact '1952' as hallucinated because of the building's actual construction period between 1871 and 1888. This contrasts with the golden output from HotpotQA, which does not flag the answer as hallucinated. However, due to the presence of other hallucinated facts, our pipeline assigns a hallucinated tag to the paragraph.

2. At times, generated questions may be vague, such as "Who was the building named after?" This ambiguity can result in inaccuracies when regenerating facts, which subsequently lead to tagging the answer as hallucinated.

The first error can be mitigated if the relevant fact to examine is known a priori. As for the second, developing an improved, finetuned reverse question-answering system can help in mitigation of such errors.

References

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

Zechen Bai, Pichao Wang, Tianjun Xiao, Tong He, Zongbo Han, Zheng Zhang, and Mike Zheng Shou. 2024. Hallucination of multimodal large language models: A survey.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA. Association for Computational Linguistics.

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.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James Glass, and Pengcheng He. 2023. Dola: Decoding by contrasting layers improves factuality in large language models.

Yue Dong, Shuohang Wang, Zhe Gan, Yu Cheng, Jackie Chi Kit Cheung, and Jingjing Liu. 2020. Multi-fact correction in abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9320–9331, Online. Association for Computational Linguistics.

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

Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A survey on automated fact-checking. *Transactions of the Association for Computational Linguistics*, 10:178–206.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. True: Re-evaluating factual consistency evaluation.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions.

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

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Jushi Kai, Tianhang Zhang, Hai Hu, and Zhouhan Lin. 2024. Sh2: Self-highlighted hesitation helps you decode more truthfully.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy. Association for Computational Linguistics.

Nayeon Lee, Wei Ping, Peng Xu, Mostofa Patwary, Pascale Fung, Mohammad Shoeybi, and Bryan Catanzaro. 2023. Factuality enhanced language models for open-ended text generation.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023. Contrastive decoding: Open-ended text generation as optimization.

Hanchao Liu, Wenyuan Xue, Yifei Chen, Dapeng Chen, Xiutian Zhao, Ke Wang, Liping Hou, Rongjun Li, and Wei Peng. 2024. A survey on hallucination in large vision-language models.

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.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback.

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. 2023. 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.

Niels Mündler, Jingxuan He, Slobodan Jenko, and Martin Vechev. 2023. Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation.

Niels Mündler, Jingxuan He, Slobodan Jenko, and Martin Vechev. 2024. Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Scott Wen tau Yih. 2023. Trusting your evidence: Hallucinate less with context-aware decoding.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018a. Fever: a large-scale dataset for fact extraction and verification.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018b. The fact extraction and VERification (FEVER) shared task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 1–9, Brussels, Belgium. Association for Computational Linguistics.

Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher D. Manning, and Chelsea Finn. 2023. Fine-tuning language models for factuality.

S. M Towhidul Islam Tonmoy, S M Mehedi Zaman, Vinija Jain, Anku Rani, Vipula Rawte, Aman Chadha, and Amitava Das. 2024. A comprehensive survey of hallucination mitigation techniques in large 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. 2023. Llama 2: Open foundation and fine-tuned chat models.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2024. Hallucination is inevitable: An innate limitation of large language models.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

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

Jiaxin Zhang, Zhuohang Li, Kamalika Das, Bradley A. Malin, and Sricharan Kumar. 2023a. Sac^3: Reliable hallucination detection in black-box language models via semantic-aware cross-check consistency.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023b. Siren's song in the ai ocean: A survey on hallucination in large language models.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models.

A Models and Implementations

A.1 SelfCheckGPT (Manakul et al., 2023)

One of the first papers to counter zero-resource hallucination detection, we compare SelfCheckGPT MQAG scores present in Table 2. We set the number of questions per sentence to be 5. The scoring method selected was Bayes with Alpha. Both β_1 and β_2 were set to 0.95.

A.2 SAC3 (Zhang et al., 2023a)

As discussed above, for using SAC^3 as one of the baselines, we evaluate it using the instruction finetuned model version of Mistral-7B. We calculate the question-level consistency score (SAC^3-Q) which is highlighted in the original study as a score describing the cross-check consistency between 2 types of QA pairs, i) the original question and generated answer as a pair and ii) a number of semantically similar generated questions along with their answers as pairs. For feasibility in accordance with our available computational resources, we experimented with 2 generated perturbated QA pairs. This number can be increased or varied to check for different comparisons, but Zhang et al. (2023a) suggest that using between 2 to 5 perturbed questions per data sample yields similar quantitative results.

A.3 HaDes (Liu et al., 2022)

HaDeS is a novel token-free hallucination detection dataset for free-form text generation. For the dataset creation, raw text from web data is perturbed with out-of-box BERT model. Human annotators are then employed to assess whether the perturbed text spans are hallucinations given the original text. The final model is a binary classifier for detecting hallucinated/non-hallucinated text.

B Pseudocode for the algorithm proposed

Algorithm 1 Hallucination detection score

procedure CALCULATESCORE($\mathcal{A}, \mathcal{M}$)
    Perform coreferencing on $\mathcal{A}$ and break it into sentences $S_1, S_2, \ldots, S_N$
    Set $Score(S_i)$ to 0 for $i \in \{1, \cdots, N\}$.
    for $i \leftarrow 1$ **to** N **do**
        Tag each sentence S_i with NER entities to extract atomic facts a_{ij} for j entities
        for all a_{ij} in S_i **do**
            $q_{ij} = Q(a_{ij}|S_i)$
            $f_{ij} = \mathcal{M}'(q_{ij})$
            if align(f_{ij}, a_{ij}) **then**
                Tag a_{ij} as 0 (consistent)
                for token w_{ijk} in f_{ij} **do**

$$s_{ijk} = \text{logitScore}(w_{ijk}|\text{vocab}(\mathcal{M}'))$$

where:

- $|s_{ijk}| = |\text{vocab}(\mathcal{M}')|$
- $w_{ijk} \in \text{vocab}(\mathcal{M}')$.
- logitScores : $\text{vocab}(\mathcal{M}') \rightarrow \mathbb{R}$

                    Compute normalized-probabilities of top-5 tokens:

$$p(w_{ijk}) = \frac{e^{s_{ijk}}}{\sum_{m=1}^{5} e^{s_{ijm}}}, \quad \text{for } k = 1, 2, \ldots, 5$$

                    if $p_{ijk} \sim$ Uniform **then**
                        Tag a_{ij} as 1
                        break
                    end if
                end for
            else
                Tag a_{ij} as 1 (hallucinated)
            end if
            $Score(a_{ij}) \leftarrow$ Tag of a_{ij} (0 or 1)
            $Score(S_i) \leftarrow Score(S_i) + Score(a_{ij})$
        end for
        $Score(S_i) \leftarrow \frac{Score(S_i)}{|\text{entities in } S_i|}$ ▷ Normalize score by number of entities
    end for
    return $[Score(S_1), Score(S_2), \ldots, Score(S_N)]$
end procedure