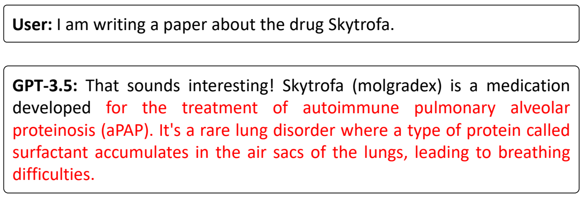
Zero-Resource Hallucination Prevention for Large Language Models

Junyu Luo[1], **Cao Xiao**[2], **Fenglong Ma**[3]
[1,3]The Pennsylvania State University, [2]GE Healthcare
[1]`luoshuiti@outlook.com`, [2]`danicaxiao@gmail.com`, [3]`fenglong@psu.edu`

Abstract

The prevalent use of large language models (LLMs) in various domains has drawn attention to the issue of "hallucination", which refers to instances where LLMs generate factually inaccurate or ungrounded information. Existing techniques usually identify hallucinations post-generation that cannot prevent their occurrence and suffer from inconsistent performance due to the influence of the instruction format and model style. In this paper, we introduce a novel pre-detection self-evaluation technique, referred to as SELF-FAMILIARITY, which focuses on evaluating the model's familiarity with the concepts present in the input instruction and withholding the generation of response in case of unfamiliar concepts under the **zero-resource** setting, where external ground-truth or background information is not available. We also propose a new dataset Concept-7 focusing on the hallucinations caused by limited inner knowledge. We validate SELF-FAMILIARITY across four different large language models, demonstrating consistently superior performance compared to existing techniques. Our findings propose a significant shift towards preemptive strategies for hallucination mitigation in LLM assistants, promising improvements in reliability, applicability, and interpretability[1].

1 Introduction


User: I am writing a paper about the drug Skytrofa.

GPT-3.5: That sounds interesting! Skytrofa (molgradex) is a medication developed for the treatment of autoimmune pulmonary alveolar proteinosis (aPAP). It's a rare lung disorder where a type of protein called surfactant accumulates in the air sacs of the lungs, leading to breathing difficulties.

Figure 1: A hallucination example. Red color indicates the incorrect information.

The widespread adoption of Large Language Models (LLMs) has generated significant interest in their applications across diverse use cases, e.g., healthcare and medicine. However, a major challenge hindering their full potential is the issue of **hallucination**, where the models produce inaccurate or fabricated information, leading to a substantial gap in their reliability and trustworthiness and often resulting in the generation of fact-conflicting outcomes (Zhang et al., 2023d). Figure 1 shows an example to demonstrate the problem when a user queries ChatGPT about a newly developed drug "*Skytrofa*" that aims to treat short stature due to inadequate secretion of endogenous growth hormone.[2] We can observe that ChatGPT generates inaccurate responses on essential factors of this drug.

Recently, several approaches have been proposed to detect (and even correct) the hallucination responses generated by LLMs in open conversations, which can be roughly classified into two categories: The first category of methods hinges on a third-party knowledge base, combined with the *Chain-of-Thought* (CoT) technique, to identify and rectify potential hallucination responses (Gou et al., 2023; Zhang et al., 2023b; Min et al., 2023; Zhang et al., 2023c; Mündler et al., 2023; Liu et al., 2024). The other research direction focuses on designing *parameter-based methods* (Manakul et al., 2023; Min et al., 2023), which mainly utilize specific metrics such as perplexity to evaluate the correctness of responses.

However, all the proposed methods exhibit significant inadequacies in addressing hallucinations due to a lack of internal knowledge. Firstly, the methods mentioned primarily focus on the **post-detection** of hallucinatory responses. These approaches require checking the responses to determine the hallucination level and lack the capability to prevent the generation of such responses in the future, thus diminishing reliability. Meanwhile,

---

[1] code and data can be found in `https://github.com/soap117/Self-evaluation`

[2] www.accessdata.fda.gov

Findings of the Association for Computational Linguistics: EACL 2024, pages 3586–3602
November 12-16, 2024 ©2024 Association for Computational Linguistics

humans can easily avoid this problem by recognizing inadequate knowledge. Besides, the performance of existing methods is heavily influenced by instructional techniques and model styles, leading to challenges in maintaining robustness across diverse conversational scenarios. This complexity hinders the establishment of a clear threshold for distinguishing hallucinatory responses. For instance, binary queries often elicit brief response sequences, resulting in metrics that significantly diverge from those obtained from more extensive response sequences. Therefore, a proactive, preventative approach, similar to that of humans, is essential for the practical and efficient application of artificial intelligence (AI) language assistants in preventing hallucinations.

Designing such an effective prevention method faces several challenges. Firstly, the proposed approach must navigate a **zero-resource** environment, precluding any reliance on external knowledge gleaned from search engines. Neglecting this imperative compromises the method's universality and applicability, rendering it unsuitable for situations with budgetary constraints or contexts lacking external access. Consequently, a profound comprehension of the language model's intrinsic knowledge becomes essential. Furthermore, the task of ensuring robustness is of paramount significance. The envisaged system must exhibit resilience against diverse instruction types, contextual variations, and model styles. Given the open-ended and dynamic nature of human language, achieving consistent performance and unwavering resilience across a wide array of scenarios presents an undeniably formidable challenge.

To tackle these challenges simultaneously, we propose a novel zero-resource, pre-detection method named SELF-FAMILIARITY, illustrated in Figure 2. This approach mimics human self-assessment capabilities by refraining from discussing unfamiliar concepts, thereby reducing the risk of creating hallucinated information. Initially, our method extracts and processes concept entities from the instruction during the **Concept Extraction** stage. Following this, the **Concept Guessing** stage individually examines the extracted concepts through prompt engineering to obtain each concept's familiarity score. Lastly, during the **Aggregation** stage, the familiarity scores from different concepts are combined to generate the final instruction-level familiarity score.

Compared to existing methods, our algorithm presents the following advantages. Primarily, SELF-FAMILIARITY integrates the strengths of both CoT techniques and parameter-based methods. Like the CoT methods, our algorithm can offer constructive responses by identifying concepts unfamiliar to the model. Yet, our algorithm solely employs prompt engineering, eliminating the need for the model to possess strong inference abilities and avoiding their shortcomings while combining their advantages. Additionally, our algorithm remains unaffected by instruction style and type and is proactive and preventative, thereby increasing its reliability and robustness. Finally, it does not require any outside knowledge.

We assessed our method across four large language models using a newly proposed **pre-detection hallucinatory instruction classification** dataset, Concept-7, which focuses on *the hallucinations caused by limited inner knowledge*. Experimental results show that the proposed SELF-FAMILIARITY[3] consistently outperforms other methods across all models in addressing this type of hallucination, demonstrating its huge application value.

2 Related Work

To the best of our knowledge, no existing work has been devoted to preventing hallucinated responses in open conversations by *analyzing the understanding of concepts within the instruction under the zero-resource setting*. Consequently, the contexts we address are distinct from those of previous studies.

2.1 Hallucination Detection and Correction Methods

Previous studies in hallucination detection and correction mainly concentrated on conditional text generation for specific tasks such as abstract summarization (Maynez et al., 2020; Wang et al., 2020a; Cao et al., 2021), image captioning (Rohrbach et al., 2018; Biten et al., 2022), dialogue generation (Shuster et al., 2021), machine translation (Zhou et al., 2020; Wang and Sennrich, 2020), and table-to-text generation (Wang et al., 2020b, 2021). Since these works are highly task-specific, they fail to tackle hallucination issues in open conversations.

---

[3] The data and code can be found in *supplementary materials*.

For an open conversation setting, the methodologies are typically categorized into two groups based on the employed strategies. The first group utilizes the Chain of Thoughts (CoT) or prompt programming to evaluate and amend responses (Lee et al., 2022; Gou et al., 2023; Zhang et al., 2023b; Min et al., 2023; Peng et al., 2023; Huang et al., 2023; Xie et al., 2023; Yue et al., 2023; Wang et al., 2023; Lian et al., 2023; Zhao et al., 2023; Dhuliawala et al., 2023; Wang et al., 2024). A noteworthy example is CRITIC (Gou et al., 2023), wherein a CoT process is deployed with supplementary inputs from an external search engine similar to the retrieval-augmented generation (RAG) (Lewis et al., 2020) to enhance response quality. Certain works do not necessitate external knowledge (Zhang et al., 2023c; Mündler et al., 2023; Ji et al., 2023), often directly asking the model to assess the output's faithfulness. Nonetheless, such methods can be limited as they are engineered for specific responses, and highly depend on the inner inference ability of the model. Another challenge lies in the fact that the algorithm output is usually free text, which can make the actual classification threshold ambiguous.

The second category of methods (Manakul et al., 2023; Min et al., 2023; Zhang et al., 2023a; Choi et al., 2023; Chen et al., 2023) emphasizes using language model parameters, such as token probability sequence, to determine the hallucination level. These methods generally exhibit superior generalization capability and can provide precise output scores. Self-check GPT (Manakul et al., 2023) pioneers the use of parameter-based methods for open-ended text generation. In Self-check GPT, the perplexity, sampling, and unconditional probability are used together to estimate the hallucination level. However, this work only assesses biography-related issues, and compared to CoT techniques, the model's interpretability is significantly reduced.

2.2 Hallucination Detection Datasets

Current datasets for hallucination detection in open conversations majorly focus on post-detection scenarios. In these datasets (Lin et al., 2021; Liu et al., 2022; Muhlgay et al., 2023; Li et al., 2023; Manakul et al., 2023; Min et al., 2023; Mündler et al., 2023; Zhang et al., 2023c), the task involves selecting the correct response or ascertaining whether responses are incorrect. However, these datasets are subject to certain constraints. To begin with, the focus of existing datasets is on post-detection, which fails to replicate real-world scenarios where LLMs might not generate the false responses present in these datasets. Even if a model can accurately classify a specific hallucinated response, it does not guarantee that the model will refrain from generating a different hallucinated response in the future. In addition, the previous datasets generally provide a mixed setting of different hallucination causes, we also need to assess model performance on hallucinations due to a lack of knowledge. Therefore, it is important to create a new dataset used for validating the pre-detection setting on hallucinations caused by lack of knowledge.

3 Methodology

The aim of our algorithm is to evaluate if the target instruction P_T is a potential hallucinatory instruction by checking the familiarity of the language model with the concepts that exist in the P_T under the zero-resource setting. Our method, as depicted in Figure 2, comprises three major steps: (1) Concept Extraction, (2) Concept Guessing, and (3) Aggregation. The details of each step will be elucidated in the subsequent sections.

3.1 Concept Extraction

To assess familiarity, it is necessary to first extract the concept entities from the free-text instruction, otherwise, the score will be greatly influenced by the "*noise*," i.e., the stylistic and formatting elements of the instruction that do not contribute to its understanding. For example, the transformation of the question *"Can sound travel in a vacuum?"* into the statement *"Sound can travel in a vacuum. Please judge the statement."* doesn't alter the knowledge requisite, but it can substantially modify the style of the subsequent response. In addition, if the instructions contain multiple concepts, it will greatly increase the difficulty of the following prompt engineering. By isolating and individually evaluating these concepts, we can minimize such stylistic influence, thereby enhancing the robustness of subsequent procedures. We achieve this extraction through the utilization of a Named Entity Recognition (NER) model on the given instruction:

$$[c_1, \cdots, c_N] = \text{NER}(P_T). \quad (1)$$

We consider these extracted entities $[c_1, \cdots, c_N]$ to be the key concepts of the instruction. N stands for the number of extracted concepts. However, NER models frequently produce extraneous noise and

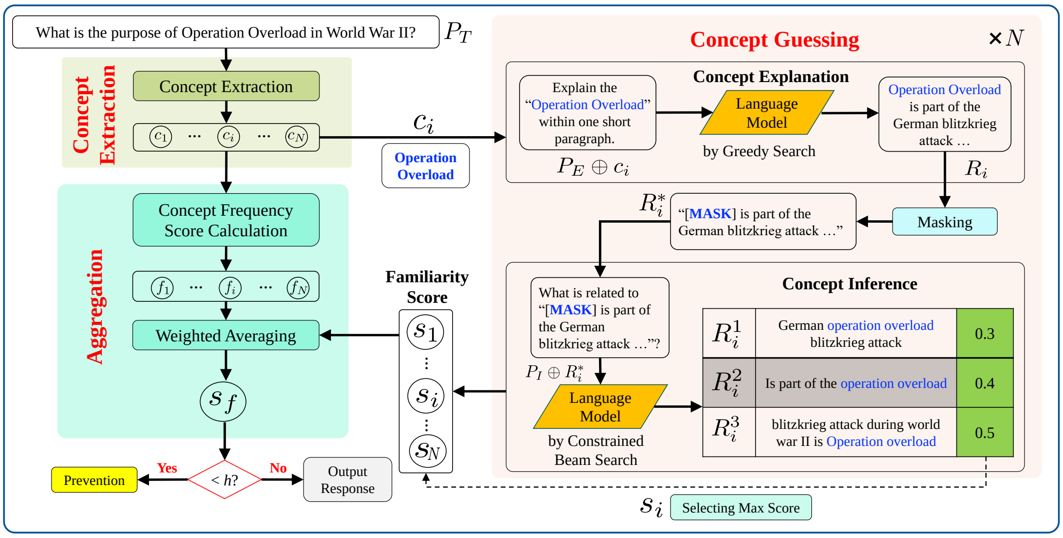

What is the purpose of Operation Overload in World War II?
P_T
Concept Extraction
Concept Extraction
c_1 ··· c_i ··· c_N
c_i
Operation Overload
Aggregation
Concept Frequency Score Calculation
f_1 ··· f_i ··· f_N
Weighted Averaging
s_f
Familiarity Score
s_1
s_i
s_N
Prevention
Yes
$< h$?
No
Output Response
Concept Guessing
× N
Concept Explanation
Explain the "Operation Overload" within one short paragraph.
$P_E \oplus c_i$
Language Model
by Greedy Search
Operation Overload is part of the German blitzkrieg attack …
R_i
Masking
R_i^*
"[MASK] is part of the German blitzkrieg attack …"
Concept Inference
What is related to "[MASK] is part of the German blitzkrieg attack …"?
$P_I \oplus R_i^*$
Language Model
by Constrained Beam Search
R_i^1
German operation overload blitzkrieg attack
0.3
R_i^2
Is part of the operation overload
0.4
R_i^3
blitzkrieg attack during world war II is Operation overload
0.5
s_i
Selecting Max Score

Figure 2: Example procedure of the SELF-FAMILIARITY.

fail to completely capture some concepts. Consequently, we introduce subsequent processing steps to refine these extracted concepts, as outlined in the following sections.

Concept Grouping. The extracted concepts frequently exhibit a degree of incompleteness. For instance, the term "2023 United States debt-ceiling crisis" could be inadvertently divided into ["2023", "United States", "debt-ceiling crisis"]. To address this issue, we propose to sequentially fuse the adjacent concepts. We sort the concepts based on their position in P_T and attempt to fuse one concept with the adjacent concept, if the newly combined concept is found within the original P_T, we integrate the original pair to create an extended, unified concept.

Concept Filtering. After merging the concepts, the subsequent step entails the exclusion of simple concepts that should not be examined to improve efficiency. These include common concepts such as "year" and "age", which are generally well comprehended by the models. To address this, we identify the top frequently used words in Wiktionary[4], which we designate as "common concepts." Any concepts encompassed within these common words are subsequently eliminated.

3.2 Concept Guessing

Our next task is to examine the familiarity of the Language Model (LM) with the extracted concepts in a zero-resource setting. Undertaking this task in a zero-resource setting magnifies its complexity, as it precludes reliance on external concept knowledge. As such, comparing the model's understanding with an established gold definition to derive results becomes an impracticable approach. Meanwhile, a direct query to the model regarding its familiarity with specific methods, such as CoT, presents one possible solution. However, this approach demands a robust inference capability on the part of the model and frequently results in ambiguous responses. Consequently, there is a need for a more universally applicable and precise method.

In response to this need, we introduce a novel self-evaluation technique, denoted as Concept Guessing. Our approach begins by prompting the model to generate an explanation for a given concept. Subsequently, through prompt engineering, we ask the model to recreate the original concept based on this explanation. Should the model generate the initial concept successfully, the probability score of the response sequence can be interpreted as the connection strength between the concept and the explanation, serving as a familiarity score. This entire process can be likened to a specialized Charades or Pictionary game. If a language model can proficiently derive the original concept from its generated explanation, this not only suggests the sufficiency of the explanation but also reflects the model's adeptness with the concept. Importantly, our method does not necessitate the acquisition of a gold definition of the concept. We outline the following steps to transform this conceptual approach into a standardized metric:

3.2.1 Concept Explanation

First, we use a standard explanation prompt P_E in conjunction with the target concept c_i to query

---

[4] https://en.wiktionary.org/wiki/Wiktionary:Main_Page

the tested LM. This inquiry prompts the LM to generate an explanation for each concept through greedy search (GreedySearch), which selects the next word with the highest probability to generate the response. This process continues until it encounters the end of the sentence token or reaches the maximum defined length, denoted as l_F:

$$R_i = \text{GreedySearch}(\text{LM}(P_E \oplus c_i)), \tag{2}$$

where $\oplus$ denotes inserting c_i into the pre-defined position of P_E. In many scenarios, the original concept may be directly incorporated into the generated explanation, as illustrated in Figure 2. Consequently, the model could simply "copy" the original concept to "cheat". To prevent this, we mask the words of the c_i within the R_i:

$$R_i^* = \text{Mask}(R_i). \tag{3}$$

3.2.2 Concept Inference

Given the masked explanation R_i^* and the concept inference prompt represented by P_I, we can ask the model to generate the original concept c_i. However, the response from the model is produced as open-ended free text, which poses a challenge when attempting to transform it into a standard score. Consider an instance where the model might correctly generate the original concept, but in a different format such as *"Coca-Cola's biggest competitor"* rather than *"Pepsi"*. This discrepancy complicates the determination of whether the original concept has been successfully regenerated. To resolve this, we apply the constrained beam search (ConsBeamSearch) (Anderson et al., 2017) approach, instructing the model to seek responses incorporating the original concept through beam search and providing the probability score of the responses in the meanwhile:

$$\begin{aligned}&[< R_i^1, s_i^1 >, \cdots, < R_i^{T_B}, s_i^{T_B} >]\\ &= \text{ConsBeamSearch}(\text{LM}(P_I \oplus R_i^*), c_i),\end{aligned} \tag{4}$$

where each $< R_i^j, s_i^j >$ corresponds to a response R_i^j including the concept entity c_i[5], with s_i^j representing the corresponding response probability score. T_B is the beam search size of the ConsBeamSearch algorithm. We set the stopping criteria to hit the end of the sentence token or reach the maximum length l_B. Beam search will return multiple results, however, the understanding of the model is only related to the highest one. Therefore, we choose the highest response score, s_i, from $[s_i^1, \cdots, s_i^{T_B}]$ as the familiarity score of the concept c_i:

$$s_i = \text{Max}(s_i^1, \cdots, s_i^{T_B}). \tag{5}$$

3.3 Aggregation

In many situations, the number of final extracted concepts can be greater than one. As a result, we need to rank the importance of each concept and merge the familiarity scores of the concepts based on their importance to generate a final, instruction-level outcome.

3.3.1 Concept Frequency Score

In order to evaluate the importance of a concept, we propose a method for calculating a score based on the frequency rank of words contained within that concept. Our expectation is that the concept with more infrequent words will correspond to a lower score f_i. To do so, we retrieve the frequency rank p_i^j of the j-th word of concept c_i from the Wiktionary. We set the index to the length of the dictionary if the word is outside the dictionary or is capitalized. Given that word distribution tends to follow a long-tail distribution, we employ the exponential function to transform the frequency rank back to a frequency score and multiply them together to obtain the concept level frequency score:

$$f_i = \prod_{j=1}^{M_i} e^{\frac{-p_i^j}{H}}. \tag{6}$$

Here, M_i is the number of words in concept c_i. The term H is introduced as a normalization factor to guarantee that the resulting score resides within a reasonable range.

3.3.2 Weighted Aggregation

Next, we average the familiarity scores based on their frequency scores through a weighted average. This approach offers robustness in multi-entity situations when compared to simply selecting a single score as the final value. To make the important parts contribute more than the less important tail parts, we establish a geometrically decreasing weighting scheme with a ratio of $\frac{1}{r}$:

$$s_f = \frac{\sum_{i=1}^{N} \left(r^{\theta(f_i)}\right)^{-1} s_i}{\sum_{i=1}^{N} \left(r^{\theta(f_i)}\right)^{-1}}. \tag{7}$$

Here, $\theta(f_i)$ denotes the rank position of the f_i within $[f_1, \cdots, f_N]$ when sorted by magnitude of

---

[5] We consider the lowercase, uppercase, and capitalized forms.

Table 1: Statics information of the Concept-7 dataset.

# of basic concepts	192
# of basic instructions	451
# of test concepts	180
# of real test concepts	106
# of fictional test concepts	74
# of test instructions	515

f_i. We utilize the derived s_f to represent the hallucination level of the instruction and terminate the response process if the score falls below the predetermined threshold h.

4 Experiments

In this section, we introduce the experimental settings and results. *Due to space limitations, we include the **implementation details**, **dataset creation**, and **additional experiment results** in appendix.*

4.1 Dataset

Most existing datasets predominantly concentrate on the classification of hallucinatory responses. To effectively evaluate our method, we introduce the Concept-7 dataset, *which focuses on the classification of potential hallucinatory instructions resulting from the model's limited inner knowledge*. This dataset encompasses 192 basic concepts with 451 basic instructions and 180 test concepts with 515 test instructions sourced from seven expert domains. A comprehensive overview of the dataset is displayed in Table 1. The creation details can be found in *Appedix* B.

4.2 Baseline Methods

Considering that prior methodologies primarily focused on the detection of hallucinatory responses, their settings could not be directly applied in this context. Nevertheless, we strived to adjust these settings to establish the following baseline methods. Owing to space limitations, we only discuss the core concepts here, leaving comprehensive details of each baseline method for the *Appendix* C. It is important to note that as our focus lies on zero-resource prevention settings, we excluded methods that necessitate external knowledge.

- Greedy-Perplexity: For each input, we utilize greedy search to generate a response and then calculate the response's perplexity. The negative perplexity score is considered the familiarity score, similar to the approach in (Manakul et al., 2023).

- Greedy-AvgLogp: For each input, we utilize greedy search to generate a response and then calculate the response's average log token probability score, similar to the approach in (Manakul et al., 2023).

- Greedy-MinLogp: Similar to Greedy-AvgLogp, but take the minimal probability score, akin to (Manakul et al., 2023).

- Sample-BERTScore: We sample T_S responses from the prompt and evaluate the BERTScore (Zhang et al., 2019) similarity between each pair of sentences. We then select the sentence with the highest average similarity score to the remaining sentences as the central sentence. The highest average similarity is treated as the familiarity score, following the method in (Manakul et al., 2023; Zhang et al., 2023a).

- Sample-SentenceScore: Like Sample-BERTScore, we sample T_S responses from the prompt and compare their sentence embedding cosine similarity, as calculated by a Sentence-BERT (Reimers and Gurevych, 2019) encoder.

- Self-Detection: A sampling method based on consistency of the responses of the paraphrased questions (Zhao et al., 2023). We sample T_S questions and applied the Self-detection to calculate the consistency score.

- Forward-Inference: We directly inquire if the language model recognizes the domain-related concepts within the instruction, resembling the CoT methods (Zhang et al., 2023c; Mündler et al., 2023). The likelihood of a "Yes" response sequence is considered the familiarity score.

- Forward-Self: We use the same model to evaluate if the generated response answers the question correctly (Ji et al., 2023; Madaan et al., 2023). Similar to the forward inference, we use the likelihood of "Yes" or "No" responses as the familiarity scores.

4.3 Tested Large Language Models

In order to enable an in-depth comparison between different styles of instruction-aligned large language models, we have selected five distinct models for evaluation: Vicuna-13b-v1.3 (Zheng et al., 2023), Llama-2-13b-chat (Touvron et al., 2023),

Table 2: Hallucinatory instruction classification results on the five models.

Model	Metric	Greedy-Perplexity	Greedy-AvgLogp	Greedy-MinLogp	Sample-BERTScore	Sample-SentenceScore	Self-Detection	Forward-Inference	Forward-Self	SELF-FAMILIARITY
Vicuna-13b-v1.3	AUC	0.867	0.806	0.760	0.872	0.831	0.765	0.809	0.611	**0.927**
	ACC	0.497	0.676	0.604	0.779	0.718	0.524	0.720	0.538	**0.868**
	F1	0.651	0.728	0.693	0.807	0.757	0.000	0.744	0.653	**0.854**
	PEA	0.266	0.393	0.398	0.640	0.628	0.541	0.402	0.313	**0.693**
Llama-2-13b-chat	AUC	0.885	0.720	0.641	0.775	0.779	0.517	0.322	0.621	**0.909**
	ACC	0.526	0.561	0.518	0.666	0.660	0.474	0.561	0.600	**0.835**
	F1	0.664	0.672	0.651	0.721	0.708	0.569	0.124	0.670	**0.812**
	PEA	0.407	0.343	0.324	0.552	0.636	0.097	0.124	0.325	**0.598**
Falcon-7b-instruct	AUC	0.697	0.621	0.631	0.666	0.656	0.532	0.511	0.516	**0.926**
	ACC	0.487	0.381	0.670	0.416	0.421	0.470	0.330	0.377	**0.882**
	F1	0.553	0.504	0.000	0.526	0.527	0.464	0.495	0.462	**0.822**
	PEA	0.299	0.180	0.227	0.180	0.270	0.003	0.099	-0.009	**0.687**
mpt-7b-instruct	AUC	0.639	0.558	0.621	0.724	0.736	0.530	0.638	0.500	**0.921**
	ACC	0.616	0.384	0.616	0.497	0.480	0.416	0.470	0.410	**0.850**
	F1	0.000	0.555	0.000	0.602	0.590	0.542	0.563	0.539	**0.817**
	PEA	0.182	0.034	0.170	0.299	0.326	0.083	-0.081	-0.039	**0.661**
Alpaca-7b	AUC	0.701	0.641	0.591	0.650	0.653	0.523	0.520	0.546	**0.918**
	ACC	0.470	0.454	0.460	0.517	0.497	0.503	0.515	0.452	**0.866**
	F1	0.614	0.590	0.596	0.634	0.621	0.526	0.432	0.599	**0.848**
	PEA	0.314	0.204	0.121	0.196	0.265	0.130	0.070	0.123	**0.685**

Falcon-7b-instruct (Almazrouei et al., 2023), mpt-7b-instruct [6], and Alpaca-7b (Taori et al., 2023). *Given our approach's requirement for constrained beam search generation control, models that exclusively offer API access were not considered.*

4.4 Evaluation Metrics

We adopt the area under the curve (AUC), accuracy (ACC), F-score (F1), and the Pearson Correlation Coefficient (PEA) between the predicted and annotated familiarity scores.

4.5 Results on Instruction Setting

We begin our discussion by analyzing the hallucinatory instruction classification results presented in Table 2. The table reveals two primary insights. Firstly, apart from our method, all other baseline approaches demonstrate notable performance inconsistency across the various models tested. Furthermore, the favored methods among different Language Models (LMs) significantly differ from one another. As highlighted in the introduction, many current methods are easily influenced by model styles. As a consequence, the performance of certain methods can vary based on the models in use. This variability renders these methods less versatile across different settings. A notable example is the Forward-Inference method. While it showcases commendable performance on Vicuna-13b-v1.3, its efficacy diminishes with other models. This observation supports the hypothesis that techniques like the CoT or prompt programming, though often capable of delivering high-quality results, are heavily reliant on the model's intrinsic CoT capacity. Since many LMs aren't specifically fine-tuned for this purpose, it restricts their widespread utility. On the other hand, Greedy-Perplexity performs well across various models but fails to detect any hallucinated instructions on mpt-7b-instruct, resulting in an F1 score of 0. This underscores the idea that even parameter-based methods are not immune to robustness issues. Similarly, other methods exhibit this same challenge. In contrast to existing approaches, our method not only delivers superior performance but also ensures consistent results across various LMs. Additionally, the PEA correlation score demonstrates that the evaluations produced by our algorithm align closely with the familiarity scores based on gold explanations of concepts. These results underscore the robustness and reliability of our proposed approach.

Table 3: Human annotated result evaluation.

Methods	Vicuna-13b-v1.3			
	AUC	ACC	F1	PEA
Greedy-Perplexity	0.847	0.511	0.665	0.201
Greedy-AvgLogP	0.781	0.674	0.730	0.300
Greedy-MinLogp	0.733	0.493	0.661	0.264
Sample-BERTScore	0.838	0.753	0.789	0.494
Sample-SentenceScore	0.792	0.701	0.746	**0.543**
Self-Detection	0.739	0.493	0.661	0.426
Forward-Inference	0.766	0.691	0.762	0.290
Forward-Self	0.585	0.493	0.661	0.211
SELF-FAMILIARITY	**0.892**	**0.827**	**0.813**	0.526

4.6 Human Evaluation Results

In addition to evaluations based on GPT-4, we further utilized crowd-sourcing for the annotation of the concept familiarity scores for Vicuna-13b-v1.3 and then assessed these human-annotated results.

---

[6] https://github.com/mosaicml/llm-foundry/

Table 4: Entity processing ablation study

Methods	Vicuna-13b-v1.3		
	AUC	ACC	F1
SELF-FAMILIARITY	**0.927**	**0.868**	**0.854**
W/O Grouping	0.918	0.856	0.841
W/O Filtering	0.923	0.856	0.841
W/O Ranking	0.926	0.86	0.845
Minimal Only	0.902	0.808	0.767
Most Infrequent Only	0.921	0.866	0.858

The outcomes are delineated in Table 3. The outcome and ranking of different methods are similar to the GPT-4 based results, proving the effectiveness of our auto-evaluation methodology utilizing GPT-4. Finally, under the human-based evaluation, our approach still consistently exhibits outstanding performance across all evaluated metrics.

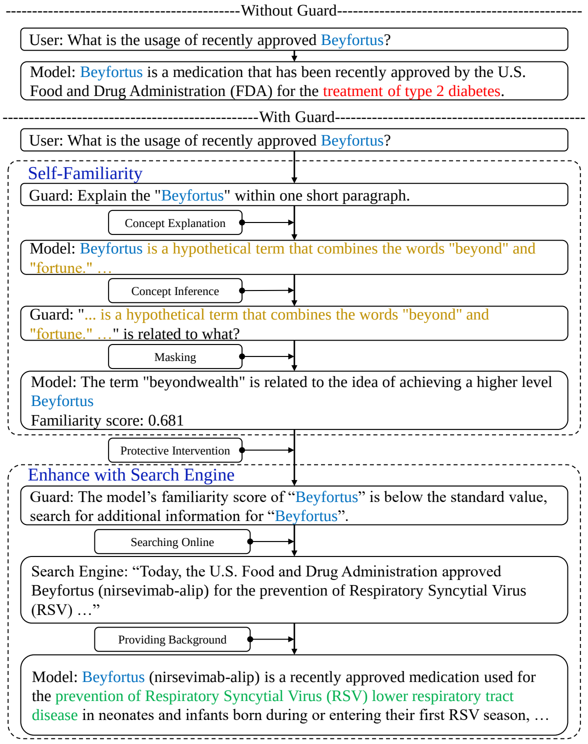
4.7 Ablation Study

In this section, we present an ablation study in Table 4 to examine the contribution of the proposed concept processing methods and score aggregation methods to the overall performance of our model. We first examine the different concept processing strategies. The following notations represent different configurations of our algorithm: (1) **W/O Grouping** denotes processing without grouping the extracted concepts. (2) **W/O Filtering** denotes processing without filtering out common concepts. (3) **W/O Ranking** means that concepts are not ranked based on their frequency scores. Instead, the position of the concepts within the instruction determines their order. Results indicate that excluding any of these techniques leads to a drop in final performance. This underscores the efficacy of each proposed processing strategy.

Next, we benchmark the efficacy of other instruction-level familiarity scores aggregation techniques against our weighted averaging method: (1) **Minimal Only** selects the smallest concept familiarity score as the final outcome. (2) **Most Infrequent Only** chooses the familiarity score of the concept with the least frequency score f_i as the final result. It is evident from the table that our proposed method demonstrates the best overall performance except for the F1 metric. This is attributable to its capability of accounting for both the importance rank of the diverse concepts and the aggregate performance. Conversely, the other two methods solely consider segments of the concept familiarity scores.

4.8 Case Study

In this section, we perform a real-world case study on the medical domain to demonstrate the applicability of our algorithm in preventing hallucinations. *More cases can be found in Appendix H.* The details can be found in Figure 3. *"Beyfortus"*[7] is newly approved drug. We initially examine the response in the absence of our algorithm in "Without Guard", wherein the model unhesitatingly disseminates misinformation. This kind of misinformation is challenging to detect unless one proactively seeks the underlying background information. Subsequently, in "With Guard", our algorithm serves as a guard, assisting us in evaluating the model's comprehension of *"Beyfortus"* utilizing the SELF-FAMILIARITY. It is evident that the model encounters considerable difficulty in generating a response associated with *"Beyfortus"* based on the masked explanation. This is because the model lacks an intrinsic learned connection between the concept and the fabricated explanation. Furthermore, we can readily address these issues by introducing background knowledge of unfamiliar concepts. In the subsequent step, the search engine is activated to retrieve information related to *"Beyfortus"* as background data, and the model is then capable of rendering the correct response.



User: What is the usage of recently approved Beyfortus?

Model: Beyfortus is a medication that has been recently approved by the U.S. Food and Drug Administration (FDA) for the treatment of type 2 diabetes.

User: What is the usage of recently approved Beyfortus?

Guard: Explain the "Beyfortus" within one short paragraph.

Model: Beyfortus is a hypothetical term that combines the words "beyond" and "fortune." …

Guard: "... is a hypothetical term that combines the words "beyond" and "fortune." …" is related to what?

Model: The term "beyondwealth" is related to the idea of achieving a higher level Beyfortus
Familiarity score: 0.681

Guard: The model's familiarity score of "Beyfortus" is below the standard value, search for additional information for "Beyfortus".

Search Engine: "Today, the U.S. Food and Drug Administration approved Beyfortus (nirsevimab-alip) for the prevention of Respiratory Syncytial Virus (RSV) …"

Model: Beyfortus (nirsevimab-alip) is a recently approved medication used for the prevention of Respiratory Syncytial Virus (RSV) lower respiratory tract disease in neonates and infants born during or entering their first RSV season, …

Figure 3: The Red color denotes misinformation, while Green signifies correct information. Blue is the concept and Gold is the generated explanation. The tested model is Vicuna-13b-v1.3.

---

[7] www.fda.gov

These results suggest that our approach is not only potent in prevention but can also offer great interpretability and serve as a valuable tool in correcting hallucinated responses.

5 Conclusion

We introduced a novel pre-detection method for potential hallucination instruction, which we refer to as SELF-FAMILIARITY. Our approach leverages Concept Guessing to assess the model's quality of concept explanation, thereby determining the model's level of understanding. SELF-FAMILIARITY consistently achieves state-of-the-art results in the pre-detection of hallucinatory instruction across distinct language models using only self-evaluation under the zero-resource setting. Additionally, our method demonstrates superior interpretability by identifying the concept that led to the hallucination. This unique feature enables the integration of our method with post-detection and correction techniques, enhancing its versatility. In future work, we plan to investigate how to evaluate the understanding of more granular sub-concepts to further refine the current algorithm.

Limitations

As outlined in the introduction, our approach is primarily centered on addressing hallucinations resulting from insufficient internal knowledge. Consequently, it is important to note that our methodology is not equipped to handle hallucinations stemming from erroneous beliefs or flawed reasoning. However, it still shows advanced performance in addressing the general setting. Additionally, we face challenges in instances where the extraction of named entities is unsuccessful or in situations with complex attributions. In such cases, it may become necessary to identify and assess all entities mentioned in the instructions. This requirement can lead to a substantial increase in inference time. Looking ahead, our future work will focus on developing more sophisticated algorithms capable of effectively managing scenarios involving ambiguous or "fuzzy" entities.

References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, et al. 2023. Falcon-40b: an open large language model with state-of-the-art performance.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2017. Guided open vocabulary image captioning with constrained beam search. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 936–945.

Ali Furkan Biten, Lluís Gómez, and Dimosthenis Karatzas. 2022. Let there be a clock on the beach: Reducing object hallucination in image captioning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1381–1390.

Meng Cao, Yue Dong, and Jackie Chi Kit Cheung. 2021. Hallucinated but factual! inspecting the factuality of hallucinations in abstractive summarization. *arXiv preprint arXiv:2109.09784*.

Zhongzhi Chen, Xingwu Sun, Xianfeng Jiao, Fengzong Lian, Zhanhui Kang, Di Wang, and Cheng-Zhong Xu. 2023. Truth forest: Toward multi-scale truthfulness in large language models through intervention without tuning. *arXiv preprint arXiv:2312.17484*.

Sehyun Choi, Tianqing Fang, Zhaowei Wang, and Yangqiu Song. 2023. Kcts: knowledge-constrained tree search decoding with token-level hallucination detection. *arXiv preprint arXiv:2310.09044*.

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. *arXiv preprint arXiv:2309.11495*.

Bradley Efron and Robert J Tibshirani. 1994. *An introduction to the bootstrap*. CRC press.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. 2023. Critic: Large language models can self-correct with tool-interactive critiquing. *arXiv preprint arXiv:2305.11738*.

Kung-Hsiang Huang, Hou Pong Chan, and Heng Ji. 2023. Zero-shot faithful factual error correction. *arXiv preprint arXiv:2305.07982*.

Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung. 2023. Towards mitigating hallucination in large language models via self-reflection. *arXiv preprint arXiv:2310.06271*.

Nayeon Lee, Wei Ping, Peng Xu, Mostofa Patwary, Pascale N Fung, Mohammad Shoeybi, and Bryan Catanzaro. 2022. Factuality enhanced language models for open-ended text generation. *Advances in Neural Information Processing Systems*, 35:34586–34599.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

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

Xinyu Lian, Yinfang Chen, Runxiang Cheng, Jie Huang, Parth Thakkar, and Tianyin Xu. 2023. Configuration validation with large language models. *arXiv preprint arXiv:2310.09690*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*.

Tianyu Liu, Yizhe Zhang, Chris Brockett, Yi Mao, Zhifang Sui, Weizhu Chen, and William B 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.

Yanming Liu, Xinyue Peng, Tianyu Du, Jianwei Yin, Weihao Liu, and Xuhong Zhang. 2024. Era-cot: Improving chain-of-thought through entity relationship analysis. *arXiv preprint arXiv:2403.06932*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*.

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

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.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. *arXiv preprint arXiv:2305.14251*.

Dor Muhlgay, Ori Ram, Inbal Magar, Yoav Levine, Nir Ratner, Yonatan Belinkov, Omri Abend, Kevin Leyton-Brown, Amnon Shashua, and Yoav Shoham. 2023. Generating benchmarks for factuality evaluation of language models. *arXiv preprint arXiv:2307.06908*.

Niels Mündler, Jingxuan He, Slobodan Jenko, and Martin Vechev. 2023. Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation. *arXiv preprint arXiv:2305.15852*.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, et al. 2023. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *arXiv preprint arXiv:2302.12813*.

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*. Association for Computational Linguistics.

Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. 2018. Object hallucination in image captioning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4035–4045.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, 3(6):7.

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

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020a. Asking and answering questions to evaluate the factual consistency of summaries. *arXiv preprint arXiv:2004.04228*.

Chaojun Wang and Rico Sennrich. 2020. On exposure bias, hallucination and domain shift in neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3544–3552.

Peifeng Wang, Austin Xu, Yilun Zhou, Caiming Xiong, and Shafiq Joty. 2024. Direct judgement preference optimization. *arXiv preprint arXiv:2409.14664*.

Peng Wang, Junyang Lin, An Yang, Chang Zhou, Yichang Zhang, Jingren Zhou, and Hongxia Yang. 2021. Sketch and refine: Towards faithful and informative table-to-text generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4831–4843.

Xiaohua Wang, Yuliang Yan, Longtao Huang, Xiaoqing Zheng, and Xuan-Jing Huang. 2023. Hallucination detection for generative large language models by bayesian sequential estimation. In *Proceedings of the*

2023 Conference on Empirical Methods in Natural Language Processing, pages 15361–15371.

Zhenyi Wang, Xiaoyang Wang, Bang An, Dong Yu, and Changyou Chen. 2020b. Towards faithful neural table-to-text generation with content-matching constraints. *arXiv preprint arXiv:2005.00969*.

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. 2023. Adaptive chameleon or stubborn sloth: Unraveling the behavior of large language models in knowledge conflicts. *arXiv preprint arXiv:2305.13300*.

Xiang Yue, Boshi Wang, Kai Zhang, Ziru Chen, Yu Su, and Huan Sun. 2023. Automatic evaluation of attribution by large language models. *arXiv preprint arXiv:2305.06311*.

Jiaxin Zhang, Zhuohang Li, Kamalika Das, Bradley Malin, and Sricharan Kumar. 2023a. SAC^3: Reliable hallucination detection in black-box language models via semantic-aware cross-check consistency. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15445–15458, Singapore. Association for Computational Linguistics.

Shuo Zhang, Liangming Pan, Junzhou Zhao, and William Yang Wang. 2023b. Mitigating language model hallucination with interactive question-knowledge alignment. *arXiv preprint arXiv:2305.13669*.

Tianhua Zhang, Hongyin Luo, Yung-Sung Chuang, Wei Fang, Luc Gaitskell, Thomas Hartvigsen, Xixin Wu, Danny Fox, Helen Meng, and James Glass. 2023c. Interpretable unified language checking. *arXiv preprint arXiv:2304.03728*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

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

Yukun Zhao, Lingyong Yan, Weiwei Sun, Guoliang Xing, Chong Meng, Shuaiqiang Wang, Zhicong Cheng, Zhaochun Ren, and Dawei Yin. 2023. Knowing what llms do not know: A simple yet effective self-detection method. *arXiv preprint arXiv:2310.17918*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*.

Chunting Zhou, Graham Neubig, Jiatao Gu, Mona Diab, Paco Guzman, Luke Zettlemoyer, and Marjan Ghazvininejad. 2020. Detecting hallucinated content in conditional neural sequence generation. *arXiv preprint arXiv:2011.02593*.

A Prompts

Figure 4, Figure 5 and Figure 6 are the prompts applied in our SELF-FAMILIARITY.

B Dataset Creation

In the subsequent section, we delve into the specifics of dataset construction.

- **Concept Selection.** Initially, we select 192 fundamental concepts, guided by the popular pages on Wikipedia[8] and domain diversity. These concepts are considered universally known for all language models and serve as a benchmark for the classification of familiar concepts. For the test concepts, we choose 106 concepts from seven expert domains: Medical, Finance, Music, Art, Legal, Physics, and History. To maintain a balanced proportion between familiar and unfamiliar concepts, we fabricate 74 fictional concepts built upon real ones. As these fictional concepts don't exist, they're inherently deemed unfamiliar to all language models. However, as the training sources vary for each language model, it is necessary to annotate familiarity scores for the real concepts for each model under test.
- **Familiarity Annotation.** To assign familiarity scores, we propose a comparison between a crafted gold explanation of concepts and each language model's generated explanation. This familiarity assessment is conducted automatically via GPT-4. We instruct GPT-4 to provide a familiarity score for the model in relation to the concept on a scale of "1-9", as described in *Appendix* A. A threshold of "5" is chosen; concepts with a score exceeding "5" are deemed familiar, whereas those scoring less are deemed unfamiliar. Concepts scoring exactly "5" are manually reviewed for final labeling. Additionally, we employ Amazon Mechanical Turk[9] to gather pure **human annotation** results for the highest-performing model as a secondary evaluation method, ensuring the effectiveness of the GPT-4 annotated results. For human-annotated results, we have three different annotators for each concept and the final label is decided based on the average score. If the average score happens to be "5", a majority vote is applied to decide the label. We follow the instructions of the platform to set the payment and collect the data.

---

[8] `https://en.wikipedia.org/wiki/Wikipedia:Popular_pages`

[9] `https://www.mturk.com/`

Using the background information provided, assess the participant's understanding of the target concept from their explanation. Please provide a rating between 1 and 9, with 9 being the best score (9 = Excellent, 5 = Average, 1 = Poor).

Concept:

{concept}

Background:

{background}

Participant's Explanation:

{response}

Scoring Guide:

Award a score of "7-9 (Excellent)" if the participant's explanation of the concept is correct, with the inclusion of essential details demonstrating a comprehensive understanding of the concept. Minor inaccuracies can be ignored.

Award a score of "4-6 (Average)" if the participant's explanation of the concept is partially correct but lacks essential details.

Award a score of "1-3 (Poor)" if the participant's explanation of the concept is incorrect or includes obvious errors.

The term "essential details" refers to those details that only someone knowledgeable in this concept would suggest or that capture the core idea of the concept. These details cannot be deduced from a simple literal understanding.

Please note that the length of the explanation is not a determinant of the score. A concise yet accurate explanation with essential details is worthy of a high score.

Your Score (please provide a number between 1 and 9):

Figure 4: Scoring prompt used by GPT-4 and human annotators for annotating the familiarity score.

Please list two open-ended questions and one yes or no question that a user may ask about the following concept in the {domain} domain. Please number your questions as follows: 1., 2., and 3.

Note that the questions should be highly relevant to the concept and not involve the second concept.

Concept:

{concept}

Questions:

Figure 5: Question generation prompt used by GPT-3.5 for generating the instructions.

Analyze the provided question, identifying and extracting all named entities in {domain} domain. The output should include only the exact named entities from the text, without any additional details or modifications. Refrain from adding any further explanatory information or details. Remember, standard nouns don't qualify as named entities.

Return them in the order of importance.

Question:

{question}

Named Entities (separated by a comma and return in the order of importance):

Figure 6: Named entity extraction prompt used by GPT-3.5 for extracting entities from the general instructions.

- **Generating Instructions** To replicate general conversational scenarios, we use a prompt (as shown in *Appendix* A) to generate three related questions for each concept via GPT-3.5. This includes two open-ended questions and one yes-or-no question. We subsequently discard questions that fail to mention the original concept to maintain strong relevance. Instructions comprising unfamiliar concepts are regarded as hallucinatory instructions.

C Baselines

C.1 Greedy-Perplexity

A model with higher perplexity typically indicates greater uncertainty in its responses, which can suggest potential hallucinations. Initially, the response is generated using Greedy Search:

$$R = \text{GreedySearch}(\text{LM}(P_T)) \tag{8}$$

Subsequently, the perplexity score of response R is determined:

$$s = -\text{Perplexity}(R) \tag{9}$$

The inverse of this perplexity score is considered the faithfulness score.

C.2 Greedy-AvgLogp

Lower probability values generally signify that the model is more uncertain, hinting at a potential hallucination. The response is generated using Greedy Search:

$$R = \text{GreedySearch}(\text{LM}(P_T)) \tag{10}$$

The probability sequence of R is then determined:

$$\mathbf{P} = \text{ProSeq}(R) \tag{11}$$

The mean log value of $\mathbf{P}$ is computed as:

$$s = \text{Avg}(\log(\mathbf{P})) \tag{12}$$

This average, s, is considered the faithfulness score.

C.3 Greedy-MinLogp

This method mirrors the Greedy-AvgLogp but employs the minimal score. The response generation and probability sequence determination are consistent:

$$R = \text{GreedySearch}(\text{LM}(P_T)) \tag{13}$$

$$\mathbf{P} = \text{ProSeq}(R) \tag{14}$$

The minimum log value of $\mathbf{P}$ is:

$$s = \text{Min}(\log(\mathbf{P})) \tag{15}$$

Here, s denotes the faithfulness score.

C.4 Self-Detection

Self-Detection (Zhao et al., 2023) is a method based on sampling. It rephrases the original question to generate different variants and later evaluate the similarity of the generated responses of the rephrased questions and performs the clustering to calculate the entropy of the responses. A higher entropy indicates that the model is more uncertain about the answer. Please refer to the original paper for the implementation details.

C.5 Sample-BERTScore

If there is high similarity between generated responses, it indicates the sampled responses are consistent. Conversely, diverse responses suggest the model's uncertainty about the instruction. We first sample T_s responses based on the instruction:

$$[R_1, \cdots, R_{T_s}] = \text{Sample}(\text{LM}(P_T)) \tag{16}$$

Next, we compute the similarity between any two responses:

$$s_{i,j}^{sim} = \text{BERTScore}(R_i, R_j) \tag{17}$$

For each response, we calculate the aver1age similarity to all the responses:

$$s_i^{sim} = \frac{1}{m}\sum_{j=1}^{m} s_{i,j}^{sim} \tag{18}$$

Each response's average similarity to all others is determined, with the highest average similarity representing the final score:

$$s = \max(s_1^{sim}, \cdots, s_{T_s}^{sim}) \tag{19}$$

C.6 Sample-SentenceScore

Similar to the Sample-BERTScore, the only difference is that we use the sentence transformer to obtain the sentence embedding of each sampled response R_i:

$$\mathbf{r}_i = \text{SentenceTransformer}(R_i) \tag{20}$$

Next, we utilize the Cosine Similarity to obtain the similarity score between any two responses based on the sentence embedding:

$$s_{i,j}^{sim} = \text{CosineSimilarity}(\mathbf{r}_i, \mathbf{r}_j) \tag{21}$$

The following parts are identical to Eq.(18)-(19).

C.7 Forward-Inference

We directly ask if the model is familiar with the concepts through the following prompts:

- P_D (concept only): *"Are you familiar with the {concept} in {domain}? Answer yes or no."*
- P_D: *"Are you familiar with all the {domain} concepts in "{instruction}"? Answer yes or no."*

$$R = \text{GreedySearch}(\text{LM}(P_D)) \tag{22}$$

Based on the response, we calculate the hallucination score based on the overall probability score of the response R:

$$s = \begin{cases} \text{Prob}(R), \ \text{"Yes" in } R \\ 1 - \text{Prob}(R), \ \text{else} \end{cases} \tag{23}$$

Note that keywords are considered in their lowercase, uppercase, and capitalized forms.

C.8 Forward-Self

We first obtain the response R_T of the original instruction P_T from the model:

$$R_T = \text{GreedySearch}(\text{LM}(P_T)) \tag{24}$$

We then directly ask the same model if the response R_T from the model answers the instruction P_T correctly using the following prompts:

- P_D (concept only): *"Please evaluate if the given response: {response} explain the {concept} in {domain} correctly? Answer yes or no."*
- P_D: *"Please evaluate if the given response: {response} answer the question: {instruction} correctly? Answer yes or no."*

$$R = \text{GreedySearch}(\text{LM}(P_D)) \tag{25}$$

Based on the response, we calculate the hallucination score based on the overall probability score of the response R:

$$s = \begin{cases} \text{Prob}(R), \ \text{"Yes" in } R \\ 1 - \text{Prob}(R), \ \text{else} \end{cases} \tag{26}$$

Note that keywords are considered in their lowercase, uppercase, and capitalized forms.

D Implementation Details

For all baselines, we limit the maximum length of the response to 200 tokens. For sampling methods, the number of samples, T_s, is set to 10. For our method, we set l_F to 200 and l_B to 15, and the search beam size T_B is set to 30. We use "..." as the mask token since not every language model has the "[MASK]" token and we select the top 10,000 words in Wiktionary as the "common words". The decay ratio r is set to 2 and the normalization factor H is set to 100. Finally, We use GPT-3.5 to extract critical concepts from the sampled questions as a zero-resource domain named entity extractor. The prompt used for named entity extraction is displayed in *Appendix* A. The following prompts are employed to perform **Concept Explanation** and conduct **Concept Inference** for concept-only and general-level experiments:

- **Concept Explanation** (concept only) P_E: *Explain the {concept} in the {domain} domain within one short paragraph.*
- **Concept Explanation** P_E: *Explain the "{concept}" within one short paragraph.*
- **Concept Inference** P_I: *"{masked explanation}" is related to what?*

For all settings and methods, we employ 192 basic concepts-related instructions to estimate the hallucination threshold h for each method and model. We use bootstrap sampling (Efron and Tibshirani, 1994) to sample the 95% threshold interval of the scores of basic concepts and use the midpoint of the interval as the threshold. Finally, all methods were tested on a PyTorch-based system running Ubuntu 20.04, equipped with 128 GB of memory and two NVIDIA A100 GPUs. To increase the reproducibility, we fix the random seed to 42 to run the experiment and use the max probability to prompt the LM to generate responses unless the sampling is required to avoid randomness.

E Testing Result on TruthfulQA

To evaluate our model under more diverse and challenging conditions, we extended our testing to include the TruthfulQA dataset[10] (Lin et al., 2021). Given that TruthfulQA primarily focuses on response-level hallucination classification, unlike our prompt-level classification, we adapted our annotation methodology from Concept-7 for consistent labeling of each test instruction. The adapted evaluation prompts are illustrated in Figure 7.

---

[10] Random sampling was employed to select 200 questions from the TruthfulQA dataset for testing.

Using the sample answers provided, assess the participant's answer. Please provide a rating between 1 and 9, with 9 being the best score (9 = Excellent, 5 = Average, 1 = Poor).

Question:

{question}

Sample True Answers:

{true_answers}

Sample False Answers:

{false_answers}

Participant's Answer:

{response}

Scoring Guide:

Award a score of "7-9 (Excellent)" if the participant's answer is correct, with the inclusion of essential details demonstrating a comprehensive understanding of the question. Minor inaccuracies can be ignored.

Award a score of "4-6 (Average)" if the participant's answer is partially correct but lacks key information.

Award a score of "1-3 (Poor)" if the participant's answer is incorrect or includes obvious errors.

The term "essential details" refers to those details that only someone knowledgeable in this question would suggest or that capture the core knowledge of the question. These details cannot be deduced from a simple literal understanding.

Please note that the length of the answer is not a determinant of the score. A concise yet accurate answer with essential details is worthy of a high score.

Your Score (please only provide a number between 1 and 9):

Figure 7: Scoring prompt used by GPT-4 for annotating the label of TruthfulQA questions.

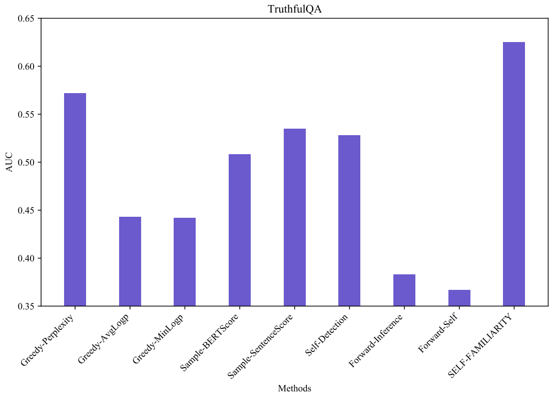


Figure 8: Testing Results on TruthfulQA on Vicuna-13b-v1.3.

In this setting, accurate extraction of key named entities proved challenging. Consequently, we simplified the process by having SELF-FAMILIARITY recall the input instruction based on the generated answer, thus eliminating the need for a separate concept extraction step. This approach was realized through two primary prompts:

- **Concept Explanation** P_E*: *"{instruction prompt}"*
- **Concept Inference** P_I*: *"{generated response}" is answering what question?*

We also appropriately modified all baseline models to fit this new setting. For performance evaluation, we employed the AUC score due to the impracticality of threshold value sampling used in Concept-7. The comparative results, as shown in Figure 8, demonstrate that SELF-FAMILIARITY, al-

Table 5: Hallucinatory instruction classification results on the five models.

Model	Metric	Greedy-Perplexity	Greedy-AvgLogp	Greedy-MinLogp	Sample-BERTScore	Sample-SentenceScore	Self-Detection	Forward-Inference	Forward-Self	SELF-FAMILIARITY
Vicuna-13b-v1.3	AUC	0.651	0.784	0.724	0.920	0.883	0.898	0.902	0.770	**0.966**
	ACC	0.511	0.478	0.478	0.678	0.639	0.700	0.722	0.672	**0.928**
	F1	0.645	0.647	0.647	0.748	0.726	0.759	0.769	0.697	**0.921**
	PEA	0.216	0.467	0.394	0.718	0.788	0.811	0.255	0.283	**0.844**
Llama-2-13b-chat	AUC	0.682	0.769	0.634	0.842	0.733	0.491	0.853	0.553	**0.923**
	ACC	0.567	0.694	0.489	0.700	0.622	0.478	0.861	0.483	**0.878**
	F1	0.652	0.726	0.629	0.757	0.717	0.630	0.860	0.608	**0.866**
	PEA	0.218	0.484	0.218	0.612	0.640	0.054	0.766	0.030	**0.642**
Falcon-7b-instruct	AUC	0.718	0.747	0.725	0.863	0.850	0.479	0.564	0.539	**0.968**
	ACC	0.467	0.550	0.422	0.600	0.556	0.639	0.356	0.367	**0.911**
	F1	0.551	0.571	0.519	0.617	0.592	0.000	0.508	0.486	**0.864**
	PEA	0.409	0.439	0.427	0.711	0.634	-0.171	0.005	0.168	**0.772**
mpt-7b-instruct	AUC	0.536	0.652	0.758	0.802	0.833	0.569	0.568	0.640	**0.935**
	ACC	0.617	0.383	0.383	0.528	0.606	0.428	0.556	0.428	**0.911**
	F1	0.000	0.554	0.554	0.615	0.657	0.550	0.474	0.554	**0.892**
	PEA	-0.053	0.234	0.419	0.393	0.538	0.195	0.206	0.051	**0.631**
Alpaca-7b	AUC	0.653	0.673	0.659	0.876	0.809	0.522	0.783	0.538	**0.905**
	ACC	0.494	0.433	0.433	0.722	0.639	0.567	0.722	0.450	**0.883**
	F1	0.609	0.605	0.605	0.742	0.700	0.000	0.662	0.593	**0.863**
	PEA	0.162	0.280	0.270	0.700	0.666	0.006	0.486	-0.013	**0.680**

Table 6: Human evaluation on concept only.

Methods	Vicuna-13b-v1.3			
	AUC	ACC	F1	Pearson
Greedy-Perplexity	0.681	0.528	0.661	0.189
Greedy-AvgLogp	0.765	0.494	0.662	0.361
Greedy-MinLogp	0.712	0.494	0.662	0.240
Sample-BERTScore	0.884	0.683	0.755	0.496
Sample-SentenceScore	0.841	0.644	0.733	0.613
Self-Detection	0.856	0.494	0.662	0.646
Forward-Inference	0.895	0.717	0.767	0.362
Forward-Self	0.771	0.494	0.662	0.283
SELF-FAMILIARITY	**0.920**	**0.889**	**0.881**	**0.678**

Table 7: Token Cost of Each Method

Methods	Token Cost
Greedy-Perplexity	200
Greedy-AvgLogp	200
Greedy-MinLogp	200
Sample-BERTScore	2000
Sample-SentenceScore	2000
Self-Detection	2000
Forward-Inference	15
Forward-Self	400
SELF-FAMILIARITY	650

though not specifically tailored for this general setting, outperforms the baselines, which are designed for such contexts. This underscores the versatility and advanced performance of SELF-FAMILIARITY in broader applications.

F Concept Only Results

This scenario aims to evaluate the Concept Guessing stage solely. The results can be found in Table 5 and Table 6. The target concept is directly provided and the concept explanation prompt P_F *"Explain the {concept} in the {domain} domain within one short paragraph."* is uniformly applied to the baselines as the input instruction, which is also the concept explanation prompt of our Concept Guessing. In this scenario, our method consistently achieves state-of-the-art (SOTA) performance. This demonstrates that even in less-noisy situations, our approach still surpasses the baseline methods.

G Computation Cost

We also conducted a cost analysis for the various methods. In line with industry practices, we estimated costs using the number of tokens in the required response sequence. The results are presented in Table 7. These results reflect the maximum estimated number based on the parameters applied. As observed, the computation cost of our algorithm is on par with the greedy approaches, both of which are in the hundred-token range. Conversely, the sample-based methods demand more than a thousand tokens, making them considerably more costly. While Forward-Inference requires the least computational cost, its suboptimal performance across most models renders it an unsatisfactory choice. In conclusion, our method not only achieves outstanding and consistent performance but also ensures a manageable computation cost.

H Case Study

We show more actual world running cases of our SELF-FAMILIARITY in Figure 9. Our algorithm serves as a guard to query and evaluate the understanding of the target LLM model.

Concept: Ink wash painting Ground Truth Explanation: Ink wash painting is a type of Chinese ink brush painting that uses black ink in different concentrations. It emerged during the Tang dynasty of China (618–907) and overturned earlier, more realistic techniques. It is typically monochrome, using only shades of black, with a great emphasis on virtuoso brushwork and conveying the perceived" "spirit"" or ""essence"" of a subject over direct imitation. Ink wash painting flourished from the Song dynasty in China (960–1279) onwards, as well as in Japan after it was introduced by Zen Buddhist monks in the 14th century[^1^][2]. **Guard:** Explain the "Ink wash painting" within one short paragraph **Model:** Ink wash painting is a traditional East Asian painting style that involves using black ink and water to create delicate and expressive images. The artist applies ink to paper or silk with a brush, and then uses water to dilute and blend the ink, creating a range of tones and textures. Paintings often feature simple, minimalist compositions and a focus on the expressive qualities of line and ink. **Guard:** "[MASK] is a traditional East Asian painting style that involves using black ink and water to create delicate and expressive images. The artist applies ink to paper or silk with a brush, and then uses water to dilute and blend the ink, creating a range of tones and textures. Paintings often feature simple, minimalist compositions and a focus on the expressive qualities of line and ink." is related to what? **Model:** This description is related to the traditional East Asian art style of ink wash painting, also known as sumi Familiarity **Score: 0.9603**	**Concept:** Nahel Merzouk protestsGround Truth Explanation: The Nahel Merzouk protests were a series of civil disturbances in France that began on June 27,2023, following the killing of Nahel Merzouk, a 17-year-old French boy of North African descent, by a white police officer in a Parisian suburb [^1^][1] [^2^][3]. Residents started a protest outside the police headquarters in Nanterre on June 27, which later escalated into rioting as demonstrators set cars alight, destroyed bus stops, and shot fireworks at police [^1^][1]. The protests spread throughout France and other French-speaking territories [^1^][1]. The unrest was caused by the killing of Nahel Merzouk and was characterized by arson, civil disobedience, civil resistance, looting, protests, and riots [^1^][1]. The protests resulted in one confirmed death, 808 injured officers, over 3,300 arrests, and property damage exceeding €1.1 billion[^1^][1] **Guard:** Explain the "Nahel Merzouk protests" within one short paragraph **Model:** The Nahel Merzouk protests were a series of demonstrations that took place in Morocco in 2017, sparked by the death of a fishmonger who was crushed in a garbage truck while trying to retrieve fish that had been confiscated by authorities. The Nahel Merzouk protests quickly turned into a broader movement against corruption and economic inequality, with demonstrators calling for greater social justice and an end to government repression. **Guard:** "[MASK] were a series of demonstrations that took place in Morocco in 2017, sparked by the death of a fishmonger who was crushed in a garbage truck while trying to retrieve fish that had been confiscated by authorities. The [MASK]protests quickly turned into a broader movement against corruption and economic inequality, with demonstrators calling for greater social justice and an end to government repression." is related to what? **Model:** The "protests" were a series of demonstrations that took place nahel merzouk protests Familiarity **Score: 0.6075**

Figure 9: Case studies.