keepitsimple at SemEval-2025 Task 3: LLM-Uncertainty based Approach for Multilingual Hallucination Span Detection

Saketh Reddy Vemula
IIIT Hyderabad
saketh.vemula@research.iiit.ac.in

Parameswari Krishnamurthy
IIIT Hyderabad
param.krishna@iiit.ac.in

Abstract

Identification of hallucination spans in black-box language model generated text is essential for applications in the real world. A recent attempt at this direction is SemEval-2025 Task 3, Mu-SHROOM—a Multilingual Shared Task on Hallucinations and Related Observable Overgeneration Errors. In this work, we present our solution to this problem, which capitalizes on the variability of stochastically-sampled responses in order to identify hallucinated spans. Our hypothesis is that if a language model is certain of a fact, its sampled responses will be uniform, while hallucinated facts will yield different and conflicting results. We measure this divergence through entropy-based analysis, allowing for accurate identification of hallucinated segments. Our method is not dependent on additional training and hence is cost-effective and adaptable. In addition, we conduct extensive hyperparameter tuning and perform error analysis, giving us crucial insights into model behavior.[1]

1 Introduction

Hallucination is a situation where Large Language Models (LLMs) produce outputs that are inconsistent with real-world facts or unverifiable, posing challenges to the trustworthiness of AI systems (Huang et al., 2025). Hallucination Detection is the process of identifying such sections of text where a model generates content that is untrue, misleading, or unverifiable by any source. As LLMs are used to generate massive texts in all applications, it is essential to make sure their output is accurate (Bommasani et al., 2022). Undetected hallucinations can propagate misinformation, lower confidence in AI systems, and have severe implications in applications such as healthcare and law. Identification of particular spans of hallucinated text, as opposed to merely marking whole outputs, is critical for real-world application, as it enables accurate corrections and improved comprehension of where and why a model hallucinate.

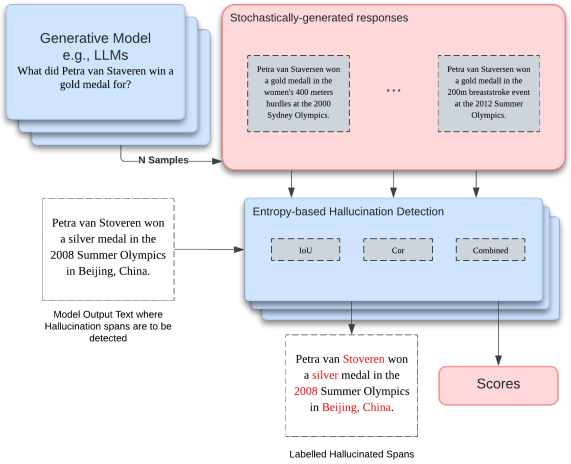

Figure 1: Architecture Diagram describing proposed method for detecting hallucination spans.(Manakul et al., 2023)

In this paper, we describe an LLM-uncertainty based method for Hallucination span detection. Our hypothesis builds upon Manakul et al. (2023) that if an LLM is certain of a given concept, stochastically-sampled responses are likely to be similar and contain consistent facts. However, for hallucinated facts, these sampled responses are likely to diverge and contradict one another. We utilize entropy information to identify the precise spans of hallucinated text using sampled responses (Xiao and Wang, 2021), allowing us to effectively identify inconsistencies that signal hallucination.

Our approach works well in zero-resource and black-box environments without any extra training. In addition, since our approach is language-independent, it works equally well in a variety of languages. Our model ranks 18th on average among over 40 submissions, achieving its best rank

[1]The code is available at https://github.com/SakethReddyVemula/semeval-2025_Mu-SHROOM

of 10th in Chinese (Mandarian).[2]

2 Related Work

The problem of hallucination detection in Large Language Models (LLMs) has been a focus of much attention recently. Hallucinations are defined as cases when LLMs produce outputs that sound plausible but are factually false or unsupported, compromising their validity for real-world usage. Farquhar et al. (2024) proposed a technique employing semantic entropy to identify such confabulations through uncertainty estimation in the semantic space of model outputs. This method calculates uncertainty at the meaning level as opposed to actual word sequences and allows for recognizing arbitrary and poor-quality generations for different datasets and tasks without explicit domain knowledge.

Following this, Kossen et al. (2024) introduced Semantic Entropy Probes (SEPs), which estimate semantic entropy directly from one generation's hidden states. SEPs are efficient in computation, avoiding repeated model samplings at inference time. Their experiments showed that SEPs have high performance in hallucination detection and generalize well to out-of-distribution test sets, indicating that model hidden states contain semantic uncertainty relevant to hallucinations.

In parallel, Manakul et al. (2023) introduced SelfCheckGPT, a zero-resource black-box method for fact-checking LLM responses independent of external databases. The technique exploits the consistency of stochastically generated responses by assuming that when an LLM has knowledge about a concept, its sampled responses will be consistent and similar in content while hallucinated facts result in diverse and contradictory responses. Their results show that SelfCheckGPT efficiently identifies non-factual sentences and evaluates the factuality of passages, providing an efficient solution for situations where model internals are not available.

These studies together highlight the need to create effective and efficient techniques for hallucination detection in LLMs. Methods based on semantic entropy, model hidden states, and response consistency provide promising directions for improving the reliability of LLM outputs in different applications.

[2] https://mushroomeval.pythonanywhere.com/submission/

3 Task Description

Mu-SHROOM [3] (Multilingual Shared-task on Hallucinations and Related Observable Overgeneration Mistakes) focuses on detecting hallucinated spans in text output from instruction-tuned LLMs. The task includes 14 languages: Arabic (Modern standard), Basque, Catalan, Chinese (Mandarin), Czech, English, Farsi, Finnish, French, German, Hindi, Italian, Spanish, and Swedish. (Vázquez et al., 2025)

Evaluation is conducted separately for each language and is based on the following two character-level metrics:

- **Intersection-over-Union (IoU):** Measures the overlap between predicted and reference hallucination spans.

$$\text{IoU} = \frac{|P \cap G|}{|P \cup G|}$$

  where P is the set of predicted hallucination characters and G is the set of gold reference hallucination characters.

- **Probability Correlation (Cor):** Evaluates how well the predicted hallucination probabilities match empirical annotator probabilities.

$$\rho = \text{corr}(\hat{p}, p)$$

  where $\hat{p}$ are the predicted probabilities and p are the human-annotated probabilities.

Data format is described in Table 1. The `hard_labels` are used for intersection-over-union accuracy, while the `soft_labels` are used for correlation evaluation. Table 5 shows the number of samples in the task dataset.

4 Methodology

In this section we describe our methodology for detecting hallucination spans. Given generated text $\mathcal{G}$ and stochastically-sampled responses $\mathcal{S} = s'_1, s'_2, ..., s'_n$ from models, our method predicts hallucination spans as follows:

Given a generated text $\mathcal{G}$, we segment it into overlapping spans using a sliding window approach. Each span s_i is extracted using a window size w and stride t such that:

$$s_i = \mathcal{G}[(i-1)t : (i-1)t + w] \quad (1)$$

[3] https://helsinki-nlp.github.io/shroom/

Field	Description
`lang`	Language of the text.
`model_input`	Input query provided to the LLM.
`model_output_text`	Generated text from the LLM.
`hard_labels`	List of pairs (s_i, e_i) representing hallucination spans (start-inclusive, end-exclusive).
`soft_labels`	List of dictionaries, each containing: • `start`: Start index of hallucination span. • `end`: End index of hallucination span. • `prob`: Probability of the span being a hallucination.

Table 1: Data fields used from Mu-SHROOM Dataset.

for all valid indices i with step size t. This ensures each part of the text is analyzed with sufficient context.

For each span s_i, we retrieve the most similar spans from a set of sampled responses $\mathcal{S} = s'_1, s'_2, ..., s'_n$ using a lexical matching function based on sequence similarity. The matching spans $\mathcal{M}_i$ are defined as:

$$\mathcal{M}_i = s'_j \in \mathcal{S} \mid \text{Similarity}(s_i, s'_j) > \tau \quad (2)$$

where τ is a threshold for similarity.

We compute the hallucination score for each span s_i using a combination of semantic entropy, lexical entropy, and frequency-based scoring.

Semantic Entropy To measure semantic inconsistency, we compute cosine similarity between the span s_i and each matched span s'_j, using a pretrained sentence embedding model:

$$\text{sim}(s_i, s'_j) = \frac{E(s_i) \cdot E(s'_j)}{|E(s_i)||E(s'_j)|} \quad (3)$$

where $E(s)$ denotes the embedding representation of span s. The probability distribution over similarities is given by:

$$P(s'_j \mid s_i) = \frac{e^{\text{sim}(s_i, s'_j)}}{\sum_k e^{\text{sim}(s_i, s'_k)}} \quad (4)$$

The semantic entropy is then computed as:

$$H_s(s_i) = -\sum_{s'_j \in \mathcal{M}_i} P(s'_j \mid s_i) \log P(s'_j \mid s_i) \quad (5)$$

Higher entropy values indicate greater semantic inconsistency.

Lexical Entropy To measure lexical variability, we compute the Shannon entropy over the frequency distribution of matched spans:

$$H_l(s_i) = -\sum_{s'_j \in \mathcal{M}_i} p(s'_j) \log p(s'_j) \quad (6)$$

where $p(s'_j)$ is the probability of span s'_j appearing in the matched set $\mathcal{M}_i$.

Frequency Score The frequency-based confidence score is computed as:

$$F(s_i) = 1 - \frac{|\mathcal{M}_i|}{|\mathcal{S}|} \quad (7)$$

where a lower $|\mathcal{M}_i|$ suggests fewer matches and a higher likelihood of hallucination.

The final hallucination score for each span s_i is computed as a weighted sum:

$$S_h(s_i) = \alpha H_s(s_i) + \beta H_l(s_i) + \gamma F(s_i) \quad (8)$$

where α, β, γ are hyperparameters controlling the contribution of each component. For our submission, we heuristically choose α = 0.4, β = 0.4 and γ = 0.2. We plan to tune these parameters in our future work.

To ensure hallucination spans align with meaningful text units, we refine span boundaries using:

- **Token boundaries**: Adjusting span edges to align with word boundaries.
- **Phrase boundaries**: Ensuring spans do not split meaningful phrases.
- **Named entity boundaries**: Avoiding incorrect segmentation of entity names.

The refined spans are selected by maximizing the entropy gradient at span boundaries.

Detected hallucination spans that overlap significantly are merged into a single span with an updated score:

$$S'_h(s) = \frac{\sum_{i \in \mathcal{O}} S_h(s_i) \cdot |s_i|}{\sum_{i \in \mathcal{O}} |s_i|} \quad (9)$$

where $\mathcal{O}$ is the set of overlapping spans.

The final output is a set of hallucination spans $\mathcal{H}$:

$$\mathcal{H} = (s_i, S_h(s_i)) \mid S_h(s_i) > \lambda \quad (10)$$

where λ is a threshold for hallucination detection.

5 Experiments

5.1 Models

Our experiments utilize Llama-3.2-3B-Instruct model (Dubey et al., 2024), a 3 billion parameter instruction-tuned language model. We generate responses using a temperature of 0.1 to maintain relatively deterministic outputs while allowing for some diversity, along with top-p sampling (nucleus sampling) set to 0.9 and top-k sampling with k=50. To avoid repetitive patterns of text, we use a 3-gram repetition penalty. We produce 20 candidate responses with a maximum of 64 tokens per input query. The model is executed in mixed-precision using FP16 to save memory, with memory consumption limited to 6GB GPU memory and 8GB CPU memory via gradient offloading.

5.2 Hyperparameter Tuning

Considering the presence of various hyperparameters in our methodology, we perform extensive hyperparameter tuning on validation split for each language. We observe that, while many languages have same set of hyperparameters performing the best on evaluation, there exist few languages where notable differences exist. We summarize our hyperparameters choice in Table 2

Language	w	t	λ	MSL	BT
arabic	4	2	0.6	3	0.3
german	4	2	0.6	3	0.3
english	5	3	0.5	3	0.3
spanish	4	2	0.6	3	0.3
finnish	4	3	0.6	3	0.3
french	4	2	0.6	3	0.3
hindi	5	2	0.6	3	0.3
italian	4	2	0.7	3	0.3
sweden	4	2	0.5	3	0.3
chinese	7	3	0.6	3	0.3

Table 2: Hyperparameters choosen for different languages. Notations include w: Window Size, t: Stride, λ: Entropy Threshold, MSL: Minimum Span Length, BT: Boundary Threshold

6 Results and Analysis

Our submission demonstrated consistent performance across multiple languages as shown in Table 3, achieving similar Intersection over Union (IoU) and Correlation (Cor) scores across various languages. The system performed particularly well in Basque (IoU: 0.4193, Cor: 0.3525), Finnish (IoU: 0.4554, Cor: 0.3323), Italian (IoU: 0.4009, Cor: 0.386) and Hindi (IoU: 0.3598, Cor: 0.3508), indicating its effectiveness in identifying and handling hallucinated text. Similarly, for languages such as English (IoU: 0.3466, Cor: 0.2104), German (IoU: 0.3651, Cor: 0.2199), and Chinese (IoU: 0.4703, Cor: 0.1601), the system maintained consistent performance, demonstrating its adaptability to different linguistic structures.

The findings reveal that our model is aptly suitable for detecting hallucinations for a wide variety of languages that possess intricate morphological and syntactic features. The high correlation scores across numerous languages confirm that our system makes good predictions which correlate well with ground truth annotation. Further, the high IoU values verify its capacity for good localization of hallucinated text, which enables it to be a trustworthy model in addressing the problems of hallucinations in multilingual environments.

6.1 Error Analysis

Table 4 reports a sample data point from test split, where our model's prediction successfully detects the hallucination span. But, it also labels other spans as hallucinated due to noise in generated responses. This behavior of false positives poses significant challenge and it must be handled. We plan to pinpoint why this happens and potentially fix this in our future work.

7 Conclusion

In this paper, we utilized an LLM-uncertainty-based method for hallucination span detection which works equally well in multiple languages. By using entropy-based uncertainty measures from sample responses, our approach accurately detects hallucinated spans without the need for further training. Our model performed competitively in various languages, ranking highly in Basque, Italian, and Hindi. The experiments emphasize the strength of our method, as they show its effectiveness in coping with varied linguistic forms and in yielding precise hallucination span detection. Our error analysis also informs on typical failure instances, presenting potential for additional refinements.

Although our approach is strong, it has limitations, specifically in exploiting supervised learning to achieve better span prediction. Our future re-

Language	Arabic		Catalan		Czech		German		English	
System	IoU	Cor	IoU	Cor	IoU	Cor	IoU	Cor	IoU	Cor
Baseline (neural)	0.0418	0.119	0.0524	0.0645	0.0957	0.0533	0.0318	0.1073	0.031	0.119
Baseline (mark none)	0.0467	0.0067	0.08	0.06	0.13	0.1	0.0267	0.0133	0.0325	0
Baseline (mark all)	0.3614	0.0067	0.2423	0.06	0.2632	0.1	0.3451	0.0133	0.3489	0
Our Submission	**0.3631**	**0.2499**	**0.3161**	**0.3377**	**0.2895**	**0.2423**	**0.3651**	**0.2199**	**0.366**	**0.2104**

Language	Spanish		Basque		Farsi		Finnish		French	
System	IoU	Cor	IoU	Cor	IoU	Cor	IoU	Cor	IoU	Cor
Baseline (neural)	0.0724	0.0359	0.0208	0.1004	0.0001	0.1078	0.0042	0.0924	0.0022	0.0208
Baseline (mark none)	0.0855	0.0132	0.0101	0	0	0.01	0	0	0	0
Baseline (mark all)	0.1853	0.0132	0.3671	0	0.2028	0.01	**0.4857**	0	0.4543	0
Our Submission	**0.2131**	**0.2335**	**0.4193**	**0.3525**	**0.3132**	**0.357**	0.4554	**0.3323**	**0.4651**	**0.2756**

Language	Hindi		Italian		Swedish		Chinese	
System	IoU	Cor	IoU	Cor	IoU	Cor	IoU	Cor
Baseline (neural)	0.0029	0.1429	0.0104	0.08	0.0308	0.0968	0.0236	0.0884
Baseline (mark none)	0	0	0	0	0.0204	0.0136	0.02	0
Baseline (mark all)	0.2711	0	0.2826	0	**0.5373**	0.0136	**0.4772**	0
Our Submission	**0.3598**	**0.3508**	**0.4009**	**0.386**	0.3967	**0.217**	0.4703	**0.1601**

Table 3: Performance comparison across different languages. IoU (⇑) : Intersection over Union. Cor (⇑) : Correlation. *Baseline (neural)* represents the baseline provided in participant kit, while *Baseline (mark none)* and *Baseline (mark all)* represents no characters labelled hallucinated and all characters labelled as hallucinated respecitively. ⇑ denotes higher is better.

Input Text	Chi ha doppiato in italiano l'attrice Catherine McCormack nel film Il sarto di Panama?
Ground Truth	L'attrice Catherine McCormack è stata doppiata in italiano da Elisa Di Stefano nel film "Il sarto di Panama"
Predicted	L'attrice Catherine McCormack è stata doppiata in italiano da Elisa Di Stefano nel film "Il sartodi Panama"

Table 4: Hallucinated spans highlighted in red for a sample datapoint in Italian.

search might consider fine-tuning over accessible training data in order to make performance even better while keeping our zero-resource model flexible. More context and fact-based verification methods can be incorporated to improve hallucination detection even further. With LLMs still evolving, creating scalable and accurate methods of hallucination detection remains a critical step to maintain the integrity of AI-produced text across real-world use cases.

Limitations

Our method does not employ supervised learning for predicting the exact spans. Under-utilization of training splits of the task is a major drawback of our system. Utilizing the training split for any kind of supervised learning could potentially improve the performance. Moreover, failing to incorporate contextual and factual verification techniques poses a major challenge to our approach.

Acknowledgments

We would like to thank Mu-SHROOM shared task organizers, Raúl Vázquez, Timothee Mickus, and their team, for their effort and commitment to organizing this task.

References

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2022. On the opportunities and risks of foundation models. *Preprint*, arXiv:2108.07258.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and

open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

Jannik Kossen, Jiatong Han, Muhammed Razzak, Lisa Schut, Shreshth Malik, and Yarin Gal. 2024. Semantic entropy probes: Robust and cheap hallucination detection in llms. *arXiv preprint arXiv:2406.15927*.

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

Raúl Vázquez, Timothee Mickus, Elaine Zosa, Teemu Vahtola, Jörg Tiedemann, Aman Sinha, Vincent Segonne, Fernando Sánchez-Vega, Alessandro Raganato, Jindřich Libovický, Jussi Karlgren, Shaoxiong Ji, Jindřich Helcl, Liane Guillou, Ona de Gibert, Jaione Bengoetxea, Joseph Attieh, and Marianna Apidianaki. 2025. SemEval-2025 Task 3: Mu-SHROOM, the multilingual shared-task on hallucinations and related observable overgeneration mistakes.

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

A Mu-SHROOM Dataset Statistics

Language	Validation	Test
ar	50	150
ca	-	100
cs	-	100
de	50	150
en	50	154
es	50	152
eu	-	100
fa	-	100
fi	50	150
fr	50	150
hi	50	150
it	50	150
sv	50	150
zh	50	150

Table 5: Number of Samples in Validation and Test data in Mu-SHROOM. For Hyperparameter Tuning, we considered validation split for languages containing validation data points. For others, we heuristically approximate the parameters.