UZH at SemEval-2025 Task 3: Token-Level Self-Consistency for Hallucination Detection

Michelle Wastl **Jannis Vamvas** **Rico Sennrich**
Department of Computational Linguistics, University of Zurich
{wastl,vamvas,sennrich}@cl.uzh.ch

Abstract

This paper presents our system developed for the *SemEval-2025 Task 3: Mu-SHROOM, the Multilingual Shared-task on Hallucinations and Related Observable Overgeneration Mistakes*. The objective of this task is to identify spans of hallucinated text in the output of large language models across 14 high- and low-resource languages. To address this challenge, we propose two consistency-based approaches: (a) token-level consistency with a superior LLM and (b) token-level self-consistency with the underlying model of the sequence that is to be evaluated. Our results show effectiveness when compared to simple mark-all baselines, competitiveness to other submissions of the shared task and for some languages to GPT4o-mini prompt-based approaches.[1]

1 Introduction

Large language models (LLMs) have transcended the boundaries of the natural language processing community and hold significant potential for automating tasks across different industries. However, their adoption is often hindered by concerns regarding their trustworthiness and factual reliability, particularly due to hallucinations — the phenomenon where generative systems produce incorrect or fabricated output. Thus, developing methods to detect hallucinations is an essential step towards ensuring the safe and effective deployment of LLMs in real-world applications.

Various approaches have been undertaken that showed the effectiveness of detecting hallucinations through retrieval augmentation and referring to external knowledge (Wang et al., 2023; Ji et al., 2023; Zhang et al., 2024; Mishra et al., 2024) or leveraging the internal states of the LLM (Xiao and Wang, 2021; Varshney et al., 2023; Farquhar et al., 2024). These methods are restricted by the fact that they rely on access to external knowledge and the openness of the LLM. A further method to detect hallucination that is not impacted by these factors leverages the self- and cross-model consistency for a given task for the same or across different LLMs (Zhang et al., 2023; Manakul et al., 2023). While effective, these methods have been almost exclusively tested at the sentence level, however, hallucinations often occur at the sub-sentential level, with incorrect or fabricated information appearing within specific spans of text rather than entire sentences. Less work has gone into detecting hallucinations at a more fine-grained level (Zhou et al., 2021; Liu et al., 2022; Fadeeva et al., 2024).

This year's *SemEval Task 3 MuSHROOM: the Multilingual Shared-task on Hallucinations and Related Observable Overgeneration Mistakes* (Vázquez et al., 2025) calls for research to fill this gap, by challenging the community on a hallucination span detection task, providing a human-annotated dataset of question-answer pairs, in which the answers have been generated by various open-source language models. The dataset contains labelled data for 10 languages (German, English, French, Italian, Spanish, Hindi, Chinese, Arabic, Swedish, and Finnish) and unlabelled data for 4 additional languages (Farsi, Basque, Czech, and Catalan) – bringing multilingual and low-resource components to the challenge.

We approach this challenge by adapting self- and cross-model consistency approaches to the token level. Our proposed approach compares alternative responses by the same underlying model that has been used to generate the answer that is to be evaluated as a way of self-consistency or by a superior LLM by aligning each token of the answer that is to be evaluated with a token from each alternative answer as a way of cross-model consistency (GPT-consistency). The median similarity score between the aligned tokens is then used as an indicator for token-level consistency, or hallucinations.

[1]The code is available at https://github.com/ZurichNLP/sc-hallucination-detection.

Proceedings of the The 19th International Workshop on Semantic Evaluation (SemEval-2025), pages 257–270

Settings	ar	zh	en	fi	fr	de	hi	it	es	sv	Avg.
Token-level labels inter alignment	44.0%	23.8%	26.7%	47.4%	36.7%	37.7%	43.4%	42.8%	16.1%	50.5%	36.9%
Token-level labels forward alignment	38.9%	42.2%	23.7%	51.5%	39.2%	36.6%	44.2%	42.6%	20.4%	46.7%	38.6%
+ median token consistency score	40.9%	42.6%	23.5%	52.0%	36.9%	41.0%	46.4%	44.5%	19.0%	48.6%	39.5%
+ word-level labels	42.1%	43.9%	25.8%	50.5%	35.9%	42.8%	52.4%	42.2%	18.3%	50.0%	40.4%

Table 1: Ablation for different settings across different languages for self-consistency on the validation set.

We find that simply prompting a strong LLM results in a strong baseline, reaching the highest results on average across languages with an overlap with gold annotations of up to 51.7%, consistency-based approaches can outperform it or achieve comparable results for languages like Chinese, French, Finnish or Swedish, reaching overlaps with human gold annotations of 47.9%–60.1%

2 Background

This work takes inspiration in self-consistency (Wang et al., 2023), a framework in which Chain-of-Thought (CoT) prompting is improved by sampling more than one possible continuation and choosing the continuation that is generated most consistently. Self-consistency has also been adapted to the hallucination detection task under the assumption that inconsistency is an indicator for counter-factuality and model uncertainty. It is proven effective at the sentence-level hallucination detection task in English: Manakul et al. (2023) compare an LLM-generated response against stochastically-generated responses and let an LLM judge over the factuality of the response. Mündler et al. (2024) investigate self-contradiction as an indicator for hallucinations. Zhang et al. (2023) introduce the concept of cross-model consistency, building on the observation that one model can hallucinate consistently across multiple continuations and therefore, an additional LLM is needed to break the consistency. In this work, we continue to follow the intuition that inconsistency indicates model uncertainty and that, especially in a cross-model setting with a superior LLM, inconsistency can be used for hallucination detection at sub-sentential level.

3 System Overview

3.1 Self- and cross-model consistency

We generate k alternative responses to the question of the original question answer pair, where $k = 5$ for self-consistency and $k = 20$ for GPT-consistency per sampling configuration. The number of sampling configurations for self-consistency is based on the provided scripts by the shared task organizers in order to stay as close as possible to the setup used to generate the hallucinated answers. This number varies, but at the minimum includes all combinations of temperatures `t = {0.1, 0.2, 0.3}` and nucleus sampling probability `p = {0.90, 0.95}`. The process of generating alternative responses is done using a superior, multilingual model `gpt-4o-mini-2024-07-18` with the set of minimum configurations. During the first iteration of experiments, all alternative responses are included in the threshold calibration and prediction, afterwards, we experiment with fewer alternative responses and analyzing the optimal configuration setting, showing the stability of our approach with down to 5 alternative responses in total (see Appendix C).

3.2 Token consistency scores

We use SimAlign (Jalili Sabet et al., 2020) with XLM-R (Conneau et al., 2020), transformer layer 8, and SimAlign variant *forward* to calculate the token alignments and their corresponding similarity scores $s_{i,1\ldots k}$ for each token i in the answer that is to be evaluated and the aligned token in each alternative answer k. The similarity scores are aggregated across alternative answers by taking their median:

$$s_i = \text{median}\left(s_{i,1}, s_{i,2}, \ldots, s_{i,k}\right).$$

We call s_i the *token consistency score*.

Table 1 shows that we found the *forward* variant of SimAlign to be superior to the *inter* variant for our purposes. Additionally, we experimented with using the mean (first two rows in Table 1) as an aggregation method for the similarity scores, but found the median to be a better fit.

3.3 Threshold calibration

The token consistency scores are then compared to a model-specific threshold to decide whether it is hallucinated or not. The threshold is calibrated based on a calibration set that we split from the shared task's validation dataset. The threshold

value that maximizes the F1-score for detecting hallucinated spans for the training set is chosen for each individual underlying model. All tokens with median similarity score below the threshold are labelled as a hallucination.

3.4 Word-level label derivation

Once the token-level label has been attained, we derive a word-level label by taking the majority vote of the labels of the subword tokens within a word and extrapolate it to the word. Although the gold label annotations are at the character level, we find that word-level predictions outperform predictions below the word level (token) (Table 1). The hard labels are then derived by matching each separate word in order with the whole answer string to find the start and end indices of the hallucinated substrings.

3.5 Baselines

Mark-all As a lower baseline, we use a mark-all approach, where we mark every single character in the answer as a hallucination, as do the organizers of the shared task (Vázquez et al., 2025).

GPT4o zero-shot classification As another baseline, we prompt `gpt-4o-mini-2024-07-18` to detect the hallucinated spans in the answer that is to be evaluated. We experiment with two different prompting techniques: a simple direct prompt and a more elaborate two-step prompt[2] . For all GPT4o-mini generations, OpenAI's structured outputs are used, and all prompts are written in English no matter what the target language is[3].

3.6 Handling of previously unseen languages

The test set contains 4 languages that are not part of the validation set and for which it was not possible to calibrate the threshold directly beforehand. In order to make a prediction for these languages nonetheless, we used the following approaches of adapting thresholds by models and languages that were seen during validation:

1. Taking the threshold of the same model in a different language. If the same model was used for multiple languages, the typologically closer language was chosen. Applied to Catalan, Basque, and Czech.

2. If the model was not seen in the validation set, the threshold was derived by averaging all model-specific thresholds for the typologically closest language. Applied to Farsi.

4 Experimental Setup

4.1 Data

All question–answer pairs containing hallucination spans were provided by the shared task's organizers in the form of a validation and test set.

The validation set contains ~50 question-answer pairs each for 10 languages (Arabic (ar), Chinese (zh), English (en), French (fr), Finnish (fi), German (de), Hindi (hi), Italian (it), Spanish (es), Swedish (sv)). Each data point comes annotated with hard labels that give the start and the end index of a hallucinated span, soft labels that includes the same indices in addition to a hallucination probability value[4], the name of the model used to generate the answer, and the token logits. We split the validation set 50/50 for a train/val split. This split is balanced according to the underlying models.

The test set contains the same annotations apart from the hard and soft labels, which were only provided after the shared task evaluation phase. It contains the same 10 and 4 additional (low-resource) languages that were not part of the validation set (Basque (eu), Catalan (ca), Czech (cs), Farsi (fa)). For each of the original 10 languages, ~150 samples were provided, while 100 samples were provided for the 4 additional languages. More detailed information on the full dataset and the models used to generate the answers is given in Table 4 in Appendix A.

4.2 Evaluation

Two evaluation metrics are considered for this task. Intersection over Union (IoU) is applied to calculate the overlap between the predicted hard labels and the gold annotations, where S_p is the predicted span and S_g is the gold span:

$$\text{IoU} = \begin{cases} 1, & \text{if both } S_g \text{ and } S_p \text{ are empty} \\ \frac{|S_g \cap S_p|}{|S_g \cup S_p|}, & \text{otherwise} \end{cases}$$

The second metric, Spearman's correlation coefficient, is used to evaluate the probability assigned to each hallucinated span in the soft labels. Since we focus on the hard labels in our experiments,

[2]The prompts are attached in Appendix D.

[3]Chollampatt et al. (2025) find that GPT models almost consistently perform better if prompts for multilingual tasks are held in English.

[4]The handling of the soft labels is described in Appendix B.2.

Method	ar	zh	en	fi	fr	de	hi	it	es	sv	Avg.
mark-all	36.1%	47.7%	34.9%	48.6%	45.4%	34.5%	27.1%	28.3%	18.5%	53.7%	37.5%
direct prompt	50.3%	39.9%	47.0%	51.7%	45.3%	**51.2%**	**63.8%**	61.7%	**40.5%**	52.6%	50.4%
two step prompt	**52.5%**	42.3%	**48.5%**	53.8%	48.6%	48.9%	61.8%	**68.3%**	36.4%	**56.1%**	**51.7%**
GPT-consistency	40.7%	42.4%	41.8%	**60.1%**	57.1%	42.2%	50.9%	51.5%	27.2%	55.4%	46.9%
GPT-consistency-fv	40.4%	**47.9%**	44.1%	51.4%	**57.7%**	41.6%	51.7%	52.8%	23.4%	55.9%	46.7%
self-consistency	36.4%	44.6%	36.3%	51.9%	47.5%	39.4%	38.8%	47.1%	20.9%	46.5%	40.9%
self-consistency-fv	35.0%	43.3%	34.2%	51.4%	51.9%	37.2%	36.4%	46.8%	18.5%	52.6%	40.7%
rank	12/32	6/29	14/44	7/30	9/33	14/31	9/27	11/31	11/35	9/30	

Table 2: IoU scores for our best-performing approaches per system for each language seen in the validation set. The rank indicates the rank achieved by the bolded approach in the shared task compared to the best submissions by all participants. -fv indicates that the full validation set was used to calibrate the threshold. These results are based on the test set.

Method	ca	cs	eu	fa	Avg.
mark-all	24.2%	26.3%	36.7%	20.3%	26.9%
direct prompt	55.6%	**39.3%**	46.5%	48.7%	47.5%
two step prompt	**58.6%**	39.2%	**50.7%**	**51.1%**	**49.9%**
GPT-consistency	41.3%	33.4%	46.6%	44.7%	41.5%
GPT-consistency-fv	39.8%	34.0%	48.6%	44.9%	41.8%
self-consistency	28.3%	30.7%	32.7%	20.7%	28.1%
self-consistency-fv	26.7%	30.5%	33.7%	21.0%	28.0%
rank	8/24	11/26	11/26	12/26	

Table 3: IoU scores for our best-performing approaches per system for each previously unseen language. The rank indicates the rank achieved by the bolded approach in the shared task compared to the best submissions by all participants. -fv indicates that the full validation set was used to calibrate the threshold.

all results for the soft labels are appended in Appendix B.2.

We first evaluate all hyperparameter settings per system on the validation set. Based on those results we choose the best-performing setting for the self- and GPT-consistency approach (Table 1). For each consistency-based approach, we additionally experiment with calibrating the threshold on the full validation set (denoted with -fv in Table 2 and 3).

5 Results and Discussion

5.1 Main results

Table 2 shows the results of our systems and baselines on the test set[5]. We find that on average, our GPT4o-mini prompt-based approaches outperform the rest. Prompt engineering led to a small improvement over a direct prompt. In cases like zh, fr (direct), and sv (direct), however, it fails to beat the lower mark-all baseline.

For zh, fi, and fr we see distinct improvement over prompting when applying GPT-consistency. This discrepancy to the prompt-based approach could indicate that for these languages, GPT4o-mini is able to generate the factual answers to the questions, but cannot locate the spans by itself just as well. While doubling the number of question-answer pairs to find a threshold does improve the performance in some cases (zh, en, fr, hi, it, sv), it performs on average worse than the system based on the original smaller split. The self-consistency results for zh, fi, and fr show competitiveness compared to the prompt-based approach. Furthermore, using the full validation for the self-consistency calculation only leads to an improvement for fr and sv. This indicates some stability to using a small number of samples that are needed for the consistency-based systems to be effective. We append a more detailed analysis of the effect of sample and alternative answer number in Appendix C.

5.2 Previously unseen languages

For the previously unseen, low-resource languages, the results are shown in Table 3. Similarly to the results for the previously seen languages, all of our systems outperform the mark-all baseline on average across all languages. A notable difference here is, however, the clear drop in performance of SC, which could be led back to the missing model and language specific threshold calibration.

The models used for ca and cs were all seen for other languages in the validation set. For eu the majority of answers have been generated with a model that has also been seen in other languages in the validation set. For Farsi a completely new set of models, none of which were seen in the vali-

[5]On average, all systems achieve higher scores on the test set compared to the validation set, while maintaining their relative ranking. There is some system ranking variation for individual languages. Most notably fr and fi, where GPT-consistency surprisingly outperforms the other approaches on the test set.

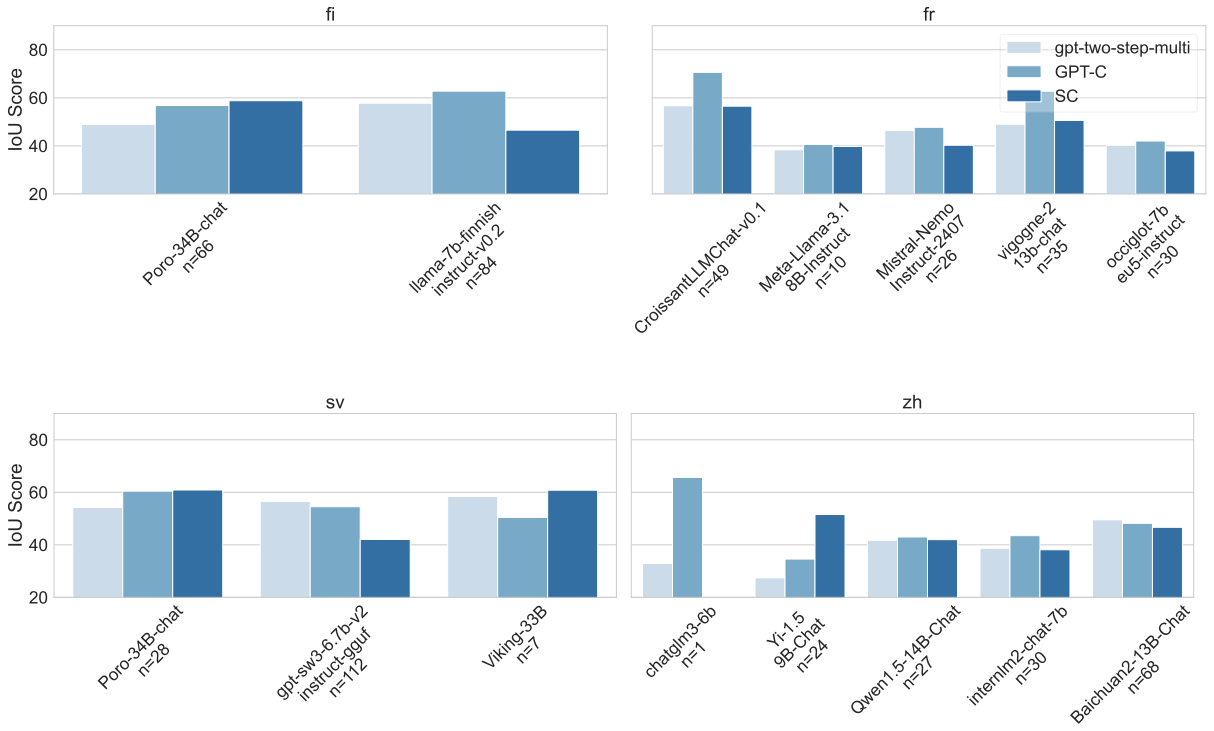


Figure 1: The performance of the zero-shot baseline, GPT-consistency and self-consistency for a subset of languages that were seen during calibration and some of whose underlying model had a parameter size larger than 9B. n indicates the number of samples/answers this statistic is based on.

dation set, have been used to generate the answers that are to be evaluated. Taking this into consideration, we can deduct that the self-consistency based approaches rely highly on the threshold being calibrated for each model specifically, and, to a lesser degree, for the specific language. GPT-consistency seems somewhat more stable in this regard.

5.3 Model-specific scores

An analysis of model-specific performance (Figure 1) reveals that the effectiveness of the above described approaches is not only determined by the given language, but it is also strongly dependent on the underlying model – more specifically, the underlying model's size. For models exceeding 8B parameters, self-consistency almost consistently outperforms the zero-shot baseline. This trend has a particularly strong effect on the overall performance of zh, where a substantial portion of the evaluated responses were generated by large models. Similar trends are seen in fi, fr and sv. In contrast, for smaller models (<8B), GPT-based zero-shot approaches generally perform better. Full insight into the model-specific results is given in Appendix E.

6 Conclusion

In this work, we introduced a (self-)consistency-based approach to hallucination span detection in LLM-generated responses for multilingual question-answering as part of the submission for the *SemEval 2025 Shared Task 3: Mu-SHROOM, the Multilingual Shared-task on Hallucinations and Related Observable Overgeneration Mistakes*.

Our method leverages both token-level self-consistency and cross-model consistency with a superior LLM (GPT4o-mini) to identify hallucinated spans. We demonstrated that our methods outperform naive baselines and remain comparable with state-of-the-art approaches for certain languages, even when operating under limited resource constraints. The results indicate that self-consistency is highly dependent on model-and language-specific threshold calibration and that it is most effective when applied to responses generated by larger models (>8B parameters), where it oftentimes outperforms the GPT-based zero-shot baseline. Future work could improve the stability across smaller models by combining self-consistency and GPT-consistency, or extend consistency-based methods to additional generative tasks beyond question-answering.

Acknowledgments

This work was funded by the Swiss National Science Foundation (project InvestigaDiff; no. 10000503).

References

Shamil Chollampatt, Minh Quang Pham, Sathish Reddy Indurthi, and Marco Turchi. 2025. Cross-lingual evaluation of multilingual text generation. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 7766–7777, Abu Dhabi, UAE. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Ekaterina Fadeeva, Aleksandr Rubashevskii, Artem Shelmanov, Sergey Petrakov, Haonan Li, Hamdy Mubarak, Evgenii Tsymbalov, Gleb Kuzmin, Alexander Panchenko, Timothy Baldwin, Preslav Nakov, and Maxim Panov. 2024. Fact-checking the output of large language models via token-level uncertainty quantification. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9367–9385, Bangkok, Thailand. Association for Computational Linguistics.

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

Masoud Jalili Sabet, Philipp Dufter, François Yvon, and Hinrich Schütze. 2020. SimAlign: High quality word alignments without parallel training data using static and contextualized embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1627–1643, Online. Association for Computational Linguistics.

Ziwei Ji, Zihan Liu, Nayeon Lee, Tiezheng Yu, Bryan Wilie, Min Zeng, and Pascale Fung. 2023. RHO: Reducing hallucination in open-domain dialogues with knowledge grounding. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4504–4522, Toronto, Canada. Association for Computational Linguistics.

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

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore. Association for Computational Linguistics.

Abhika Mishra, Akari Asai, Vidhisha Balachandran, Yizhong Wang, Graham Neubig, Yulia Tsvetkov, and Hannaneh Hajishirzi. 2024. Fine-grained hallucination detection and editing for language models. In *First Conference on Language Modeling*.

Niels Mündler, Jingxuan He, Slobodan Jenko, and Martin Vechev. 2024. Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation. In *The Twelfth International Conference on Learning Representations*.

Neeraj Varshney, Wenlin Yao, Hongming Zhang, Jianshu Chen, and Dong Yu. 2023. A stitch in time saves nine: Detecting and mitigating hallucinations of llms by validating low-confidence generation. *Preprint*, arXiv:2307.03987.

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.

Xiaohua Wang, Yuliang Yan, Longtao Huang, Xiaoqing Zheng, and Xuanjing 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, Singapore. Association for Computational Linguistics.

Yijun Xiao and William Yang Wang. 2021. On hallucination and predictive uncertainty in conditional language generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2734–2744, Online. Association for Computational Linguistics.

Jiaxin Zhang, Zhuohang Li, Kamalika Das, Bradley Malin, and Sricharan Kumar. 2023. 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. 2024. The knowledge alignment problem: Bridging human and external knowledge for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*,

pages 2025–2038, Bangkok, Thailand. Association for Computational Linguistics.

Chunting Zhou, Graham Neubig, Jiatao Gu, Mona Diab, Francisco Guzmán, Luke Zettlemoyer, and Marjan Ghazvininejad. 2021. Detecting hallucinated content in conditional neural sequence generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1393–1404, Online. Association for Computational Linguistics.

A Details on Dataset

Lang	# train	# val	# test	Model Names
Arabic (ar)	26	24	150	SeaLLM-7B-v2.5, openchat-3.5-0106-gemma, Arcee-Spark
Basque (eu)	0	0	99	Meta-Llama-3-8B-Instruct, gemma-7b-it
Catalan (ca)	0	0	100	Meta-Llama-3-8B-Instruct, occiglot-7b-es-en-instruct
Chinese (zh)	25	25	150	Qwen1.5-14B-Chat, Baichuan2-13B-Chat, Yi-1.5-9B-Chat, internlm2-chat-7b
Czech (cs)	0	0	100	Mistral-7B-Instruct-v0.3, Meta-Llama-3-8B-Instruct
English (en)	24	26	154	mistral, falcon-7b-instruct, Pythia-Chat-Base-7B
Farsi (fa)	0	0	100	PersianMind-v1.0, Meta-Llama-3.1-8B-Instruct, Llama-3.2-3B-Instruct, aya-23-35B, aya-23-8B
Finnish (fi)	26	24	150	Poro-34B-chat, llama-7b-finnish-instruct-v0.2
French (fr)	25	25	150	CroissantLLMChat-v0.1, vigogne-2-13b-chat, Mistral-Nemo-Instruct-2407, occiglot-7b-eu5-instruct, Meta-Llama-3.1-8B-Instruct
German (de)	24	26	150	bloom-6b4-clp-german-oasst-v0.1, SauerkrautLM-7B-v1-GGUF, occiglot-7b-de-en-instruct
Hindi (hi)	25	25	150	ProjectIndus, OpenHathi-7B-Hi-v0.1-Base, Meta-Llama-3-8B-Instruct
Italian (it)	25	25	150	modello-italia-9b, Meta-Llama-3.1-8B-Instruct, DanteLLM-7B-Instruct-Italian-v0.1, Qwen2-7B-Instruct
Spanish (es)	25	25	152	Llama-3-Instruct-Neurona-8b-v2, Meta-Llama-3-8B-Instruct, Qwen2-7B-Instruct
Swedish (sv)	25	24	150	gpt-sw3-6.7b-v2-instruct-gguf, Poro-34B-chat, Viking-33B

Table 4: Statistics on the dataset splits with model names.

B Correlation of Hallucination Probability to Gold Annotation

B.1 Predicting the hallucination probability

To predict the hallucination probability of each span we always assign a probability of 1. The correlation between the predicted probabilities and the gold labels is shown in Table 5 for the submitted systems on the validation set.

B.2 Evaluating the hallucination probability

To measure the correlation between the predicted probability values with the gold values for the soft labels, Spearman's correlation was applied, where p_i is the ranked position of the predicted and r_i of the gold probability:

$$\rho = \begin{cases} 1, & \text{if both } \mathbf{r} \text{ and } \mathbf{p} \text{ contain only a single unique value and match} \\ 0, & \text{if one contains a single unique value and the other does not match} \\ 1 - \frac{6\sum(r_i - p_i)^2}{n(n^2-1)}, & \text{otherwise} \end{cases}$$

Method	ar	zh	en	fi	fr	de	hi	it	es	sv	Avg.
GPT-consistency-allone	30.23%	20.52%	36.56%	37.66%	30.69%	30.23%	54.21%	34.02%	25.55%	22.13%	32.18%
self-consistency-allone	36.37%	20.89%	23.08%	43.43%	27.10%	33.94%	50.88%	32.94%	25.73%	18.90%	31.33%

Table 5: Spearman correlation scores for our best-performing approaches per system for each language seen in the validation set.

C Stability to a reduced number of alternative responses

One limitation of the consistency-based approaches in the main section is that they require a considerable number of alternative responses, and therefore, computation and time. Hence, we decided to experiment with 5 alternative responses during either the prediction, for threshold generation or both. Additionally, we look at the influence different configuration settings have on the performance. Table 6 shows the results on overlap with gold annotations on the validation split.

When comparing the averages across languages, a small performance degradation is noticeable everywhere. GPT-consistency seems to be more stable across the board, with weaker fluctuation across sampling configurations. None of the averages degrade more than 1.5%. In this regard, self-consistency shows on average stronger fluctuations when changing the sampling configurations for a reduced size of alternative responses, degrading as much as 5.1% (sc-fewer-both; p0.90 t0.1) and even gaining 0.1% (sc-fewer-pred; p0.90 t0.3).

	Config	ar	zh	en	fi	fr	de	hi	it	es	sv	Avg.
	sc-all	42.1%	43.9%	25.8%	50.5%	35.9%	42.8%	52.4%	42.2%	18.3%	50.1%	40.4%
sc-fewer-pred	p0.90 t0.1	42.5%	41.9%	21.5%	47.1%	33.9%	38.9%	40.9%	37.6%	14.9%	35.8%	35.5%
	p0.90 t0.2	40.6%	43.4%	26.7%	51.3%	33.9%	42.2%	53.2%	38.9%	17.9%	49.4%	39.7%
	p0.90 t0.3	39.0%	44.9%	27.7%	51.4%	35.8%	47.0%	51.8%	43.5%	17.2%	47.9%	40.6%
	p0.95 t0.1	42.0%	42.0%	25.9%	49.4%	31.9%	43.6%	42.0%	34.5%	16.2%	50.6%	37.8%
	p0.95 t0.2	44.8%	43.8%	25.2%	50.2%	34.6%	40.3%	51.0%	41.2%	16.7%	47.2%	39.5%
	p0.95 t0.3	41.5%	43.4%	23.8%	52.2%	37.0%	46.7%	53.2%	40.3%	15.7%	47.6%	40.1%
	one of each	42.7%	43.7%	24.3%	52.3%	31.9%	44.6%	47.5%	36.9%	17.1%	47.7%	38.9%
sc-fewer-th	p0.90 t0.1	41.4%	44.5%	21.3%	49.6%	38.6%	39.0%	40.7%	43.3%	12.3%	50.2%	38.1%
	p0.90 t0.2	43.9%	44.8%	23.3%	48.3%	38.1%	40.9%	43.9%	42.4%	18.9%	47.9%	39.3%
	p0.90 t0.3	40.7%	44.1%	24.8%	49.0%	35.8%	39.9%	43.2%	39.1%	18.5%	48.8%	38.4%
	p0.95 t0.1	43.1%	44.8%	23.8%	48.1%	36.5%	39.4%	36.9%	45.3%	16.0%	50.2%	38.4%
	p0.95 t0.2	43.2%	43.7%	23.7%	46.9%	40.5%	40.1%	43.2%	41.8%	17.2%	53.6%	39.4%
	p0.95 t0.3	43.8%	45.5%	23.7%	48.3%	38.2%	40.2%	44.9%	43.3%	18.7%	49.6%	39.6%
	one of each	38.8%	44.9%	21.2%	48.7%	40.6%	40.7%	43.6%	42.6%	16.1%	50.1%	38.7%
sc-fewer-both	p0.90 t0.1	39.0%	42.3%	20.2%	46.3%	34.3%	40.3%	38.8%	44.1%	11.3%	36.1%	35.3%
	p0.90 t0.2	40.9%	42.8%	27.0%	51.0%	35.7%	40.1%	52.9%	44.2%	19.4%	48.6%	40.3%
	p0.90 t0.3	39.9%	44.2%	25.7%	50.4%	33.2%	39.1%	51.7%	40.6%	17.7%	45.5%	38.8%
	p0.95 t0.1	39.4%	41.9%	26.6%	50.1%	34.3%	41.8%	33.9%	39.0%	14.8%	47.6%	36.9%
	p0.95 t0.2	40.1%	43.7%	25.0%	49.4%	34.1%	45.1%	50.7%	40.5%	18.2%	53.0%	40.0%
	p0.95 t0.3	40.2%	39.5%	23.7%	52.6%	37.1%	38.8%	53.6%	40.5%	17.1%	45.7%	38.9%
	one of each	40.7%	43.7%	23.2%	53.1%	34.0%	38.8%	46.0%	36.6%	15.9%	50.1%	38.2%
	gpt-all	41.5%	42.5%	31.7%	49.3%	41.2%	41.9%	55.6%	44.3%	27.5%	59.7%	43.5%
gpt-fewer-pred	p0.90 t0.1	40.4%	40.8%	31.4%	49.3%	41.0%	41.9%	55.7%	44.2%	27.3%	59.5%	43.1%
	p0.90 t0.2	40.3%	40.4%	32.1%	49.4%	41.3%	42.7%	56.0%	42.7%	23.2%	59.9%	42.8%
	p0.90 t0.3	40.9%	40.2%	31.0%	48.5%	40.9%	42.9%	55.9%	42.5%	23.4%	61.3%	42.8%
	p0.95 t0.1	41.1%	40.6%	31.0%	49.6%	41.3%	40.7%	55.5%	44.1%	27.2%	59.6%	43.1%
	p0.95 t0.2	40.6%	40.2%	32.1%	48.3%	41.2%	42.6%	55.5%	43.1%	27.5%	59.9%	43.1%
	p0.95 t0.3	40.4%	40.4%	31.8%	48.5%	41.2%	42.9%	56.0%	42.6%	23.4%	60.9%	42.8%
	one of each	39.5%	40.3%	31.5%	48.3%	41.1%	41.9%	55.8%	44.4%	23.6%	59.9%	42.6%
gpt-fewer-th	p0.90 t0.1	41.2%	40.5%	32.3%	49.6%	42.0%	40.4%	55.6%	42.1%	26.1%	58.7%	42.9%
	p0.90 t0.2	42.0%	40.1%	31.8%	49.4%	42.4%	40.1%	55.7%	43.0%	27.5%	60.1%	43.2%
	p0.90 t0.3	41.8%	41.0%	32.1%	48.3%	42.2%	40.2%	56.0%	44.4%	24.1%	59.2%	42.9%
	p0.95 t0.1	41.3%	40.5%	31.7%	48.8%	41.9%	38.3%	55.6%	43.0%	25.5%	59.5%	42.6%
	p0.95 t0.2	41.9%	40.1%	31.8%	48.9%	42.7%	37.5%	55.6%	44.7%	28.0%	59.3%	43.0%
	p0.95 t0.3	41.9%	40.9%	31.7%	49.3%	42.3%	40.8%	55.7%	44.5%	27.7%	59.6%	43.4%
	one of each	41.9%	40.7%	29.9%	47.8%	41.6%	41.1%	55.6%	44.2%	26.8%	59.2%	42.9%
gpt-fewer-both	p0.90 t0.1	39.9%	40.2%	31.9%	49.3%	41.5%	41.3%	55.7%	42.0%	25.9%	58.0%	42.6%
	p0.90 t0.2	39.2%	39.7%	32.2%	49.4%	42.3%	42.6%	56.1%	41.8%	23.3%	60.4%	42.7%
	p0.90 t0.3	40.0%	40.5%	33.7%	47.5%	42.0%	42.9%	56.0%	42.6%	19.9%	60.7%	42.6%
	p0.95 t0.1	38.3%	40.3%	31.0%	48.7%	41.9%	40.3%	55.5%	42.8%	25.0%	59.3%	42.3%
	p0.95 t0.2	37.5%	39.6%	32.1%	47.8%	42.5%	42.4%	55.7%	43.6%	27.9%	60.3%	42.9%
	p0.95 t0.3	39.8%	40.5%	31.6%	48.4%	42.3%	42.9%	56.0%	43.5%	23.3%	60.7%	42.9%
	one of each	39.2%	40.2%	29.8%	47.0%	41.5%	41.8%	55.7%	44.2%	22.4%	59.3%	42.1%

Table 6: IoU results for self-consistency with fewer alternative responses during prediction, thresholds calibrated with fewer alternative responses or both in comparison to using all alternative responses for both calibrating the threshold and prediction.

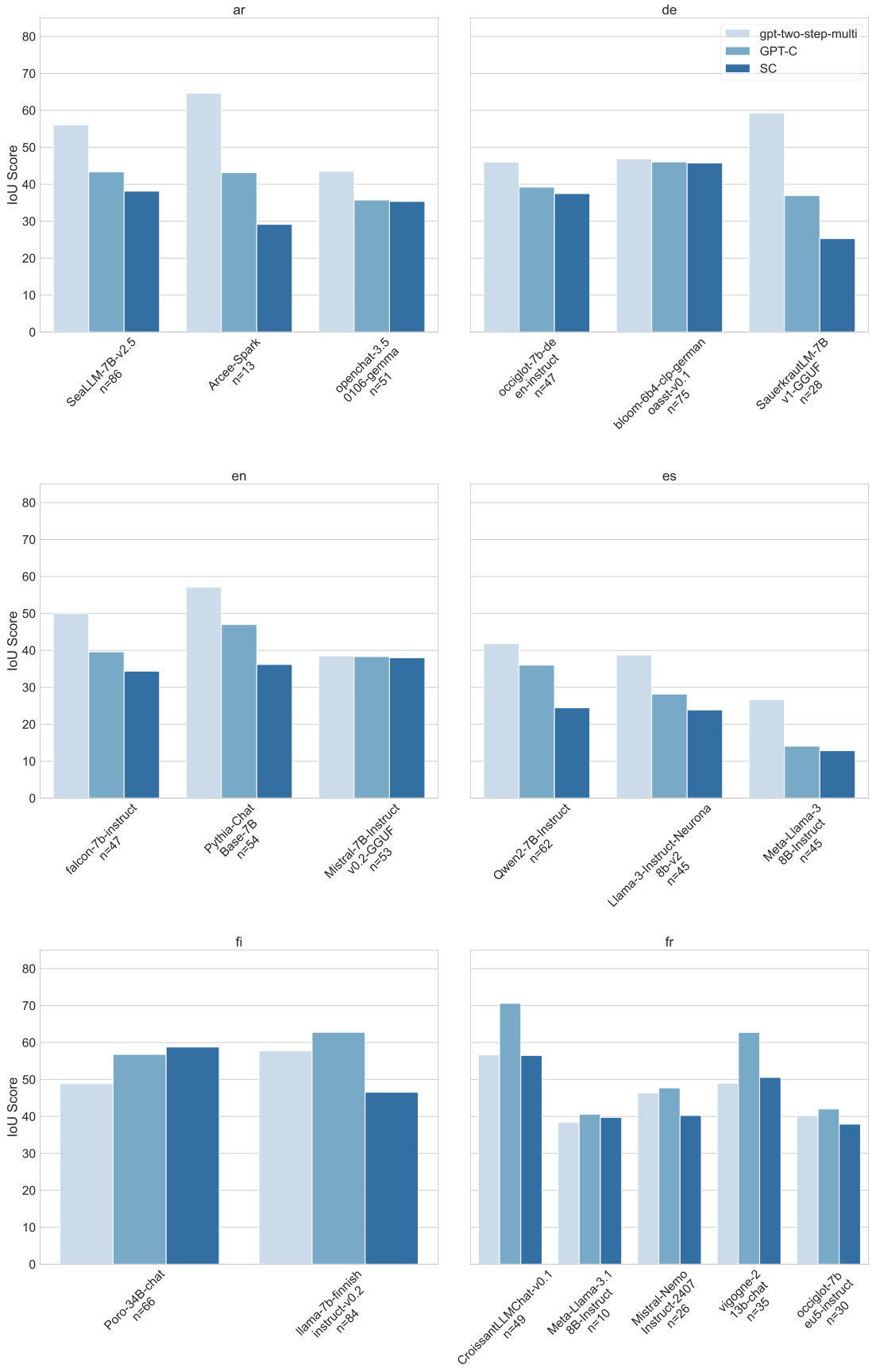
D Prompts for the Zero-Shot GPT4o-mini Baseline

We use GPT4o-mini as a zero-shot baseline with the following two prompts:

Method	Prompt
Direct prompt	**System Prompt:** "You will be given a question-answer pair. The answer might contain spans that are counterfactual. You will output the counterfactual spans. Note that the answers can also be fully counterfactual or not at all."
	User Prompt: "Question: {model_input}. Answer: {model_output}."
Two step prompt	**User Prompt (Alternative Answer):** "Answer the following question with five possible answers: {model_input}."
	System Prompt: "You will be given a question-answer pair. The answer might contain spans that are counterfactual. You will also be given multiple other possible answers. Based on these other possible answers, output the spans from the answer of the initial question-answer pair that are counterfactual. Note that the answers can also be fully counterfactual or not at all."
	User Prompt: "Question: {model_input} Answer: {model_output} Other possible answers: {alternative_answer}."

Table 7: System and user prompts used in our experiments.

E Model-specific analysis



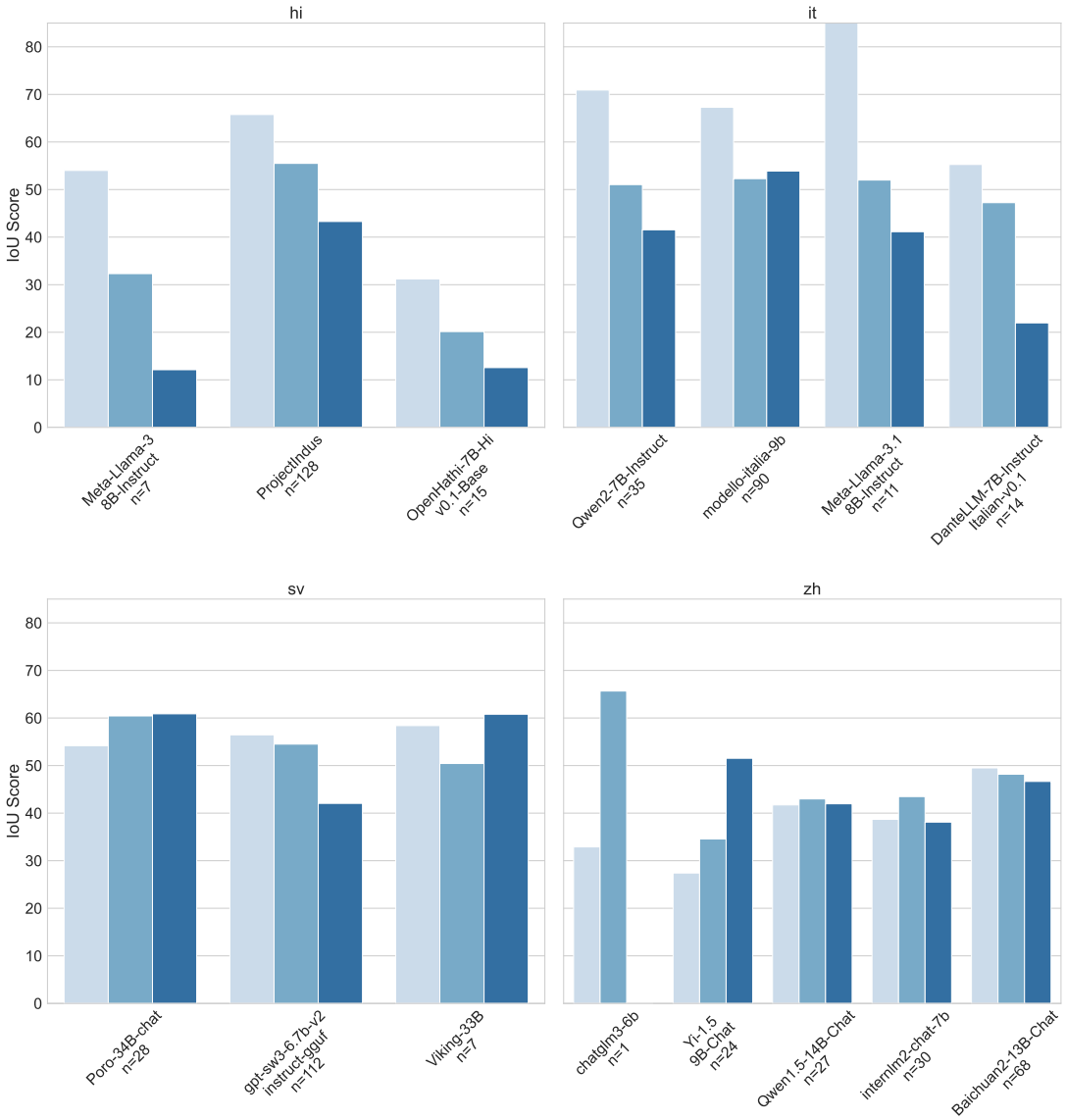


Figure 2: The performance of the zero-shot baseline, GPT-consistency and self-consistency for each language that was seen during calibration and model used to generate the answers that are to be searched for hallucinations. n indicates the number of samples/answers this statistic is based on.

A comparison of hallucination detection performance across different approaches, languages, and model architectures (Figure 2) suggests that the effectiveness of a given method is not solely determined by the target language but is also influenced by the model that generated the evaluated text.

Notably, GPT-based approaches tend to outperform self-consistency when the underlying model is relatively small ($\leq$8B parameters). However, for larger models ($>$8B parameters), self-consistency consistently surpasses the zero-shot baseline. This observation may explain SC's relatively strong performance for zh, as roughly 4/5 of the Chinese evaluation data was generated by models exceeding 8B parameters. The same trend can also be observed in sv, it, fi and fr.

Figure 3 shows similar and consistent behavior for the smaller underlying models of the unseen languages during threshold calibration. Only for fa, models with a parameter size $>$9B were used, but since none of those were seen during calibration for the other languages, SC' performance also degrades for the 35B model.

This model-specific analysis is to be interpreted with caution due to the small and imbalanced sample sizes. To make more certain, claims the experiments would have to be repeated on a larger dataset.

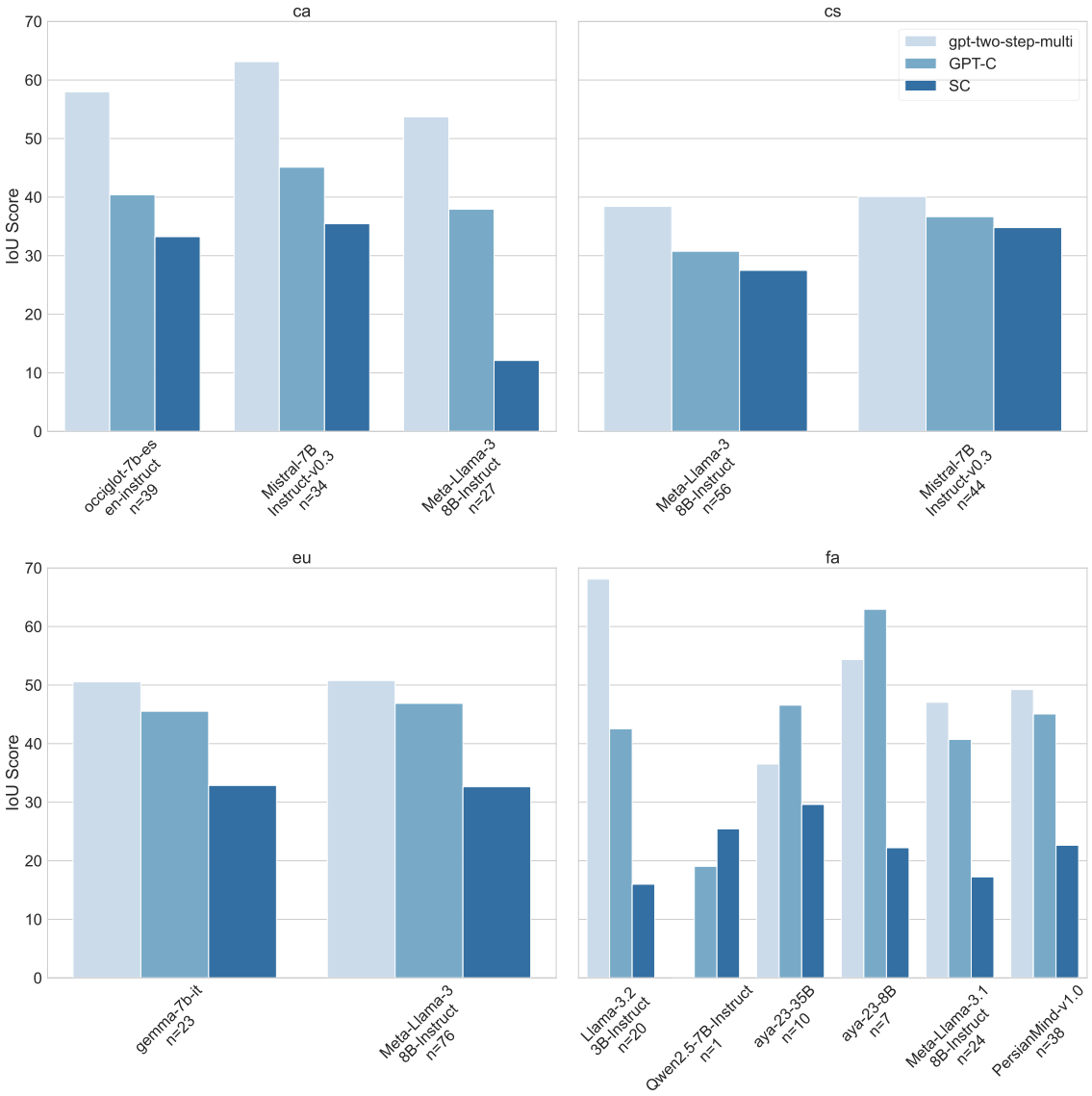


Figure 3: The performance of the zero-shot baseline, GPT-consistency and self-consistency for each language that was *not* seen during calibration and model used to generate the answers that are to be searched for hallucinations. n indicates the number of samples/answers this statistic is based on.