

Team Bit-by-bit at SHROOM-Visions: Adapt, Calibrate, and Average for Detecting Character-Level Hallucination Spans

Ioana Branescu^{1*} Mihai Dascalu^{1,2}

¹Computer Science & Engineering Department, National University of Science and Technology POLITEHNICA Bucharest, 060042 Bucharest, Romania

²Academy of Romanian Scientists, 050044 Bucharest, Romania

ioana.branescu0601@stud.acs.upb.ro, mihai.dascalu@upb.ro

Abstract

Vision-language models (VLMs) often produce details not supported by the input image. The SHROOM-Visions 2026 shared task studies this problem at the character level: given an image, a question, and a VLM-generated answer, systems must identify the hallucinated spans and assign an error type to each one. Our approach fine-tunes Qwen2.5-VL-7B and adds two lightweight classification heads to predict the hallucination probability and error type of each answer token. We map these predictions back to characters, decode spans aligned with word boundaries, and combine them with complementary components for selected languages. These include a fine-tuned InternVL3.5 model and a claim verification pipeline in which the base Qwen2.5-VL checks answer claims against the image. Across the 4 languages, our submissions reached an average Spearman correlation of 0.47 with human annotations, or 0.36 when the predicted error type must also be correct. Our best result is obtained on Italian, where combining the two fine-tuned models reaches 0.50 correlation, ranking sixth on the official leaderboard.

1 Introduction

Vision-Language Models (VLMs) answer open-ended questions about images, but even a broadly correct answer may contain unsupported details, such as an incorrect count, color, attribute, or object. A verdict on the answer as a whole is therefore of limited use, since it discards what is correct and does not indicate what needs to be revised. The SHROOM-Visions 2026 shared task (Vázquez et al., 2026) targets this local failure: given an image, a question, and a VLM-generated answer, systems must identify the unsupported character spans and assign an error type to each span.

Each answer is annotated by multiple annotators, so every character receives a graded hallucination score based on annotator agreement. Systems are evaluated using the average per-answer Spearman correlation between these gold scores and the predicted character scores, both without error labels (COR) and with the correct label required (CORLBL).

The main contributions of our study are:

- We introduce a hallucination-span detector based on Qwen2.5-VL-7B, adapted with QLoRA. Two lightweight classification heads use the final-layer hidden state of each answer token to predict its hallucination probability and error type. The hallucination head is trained directly on the graded annotator-agreement labels.
- We map token-level predictions back to characters, snap predicted span boundaries to complete words, and calibrate the decoding threshold. Both word-boundary alignment and threshold calibration improve the final results.
- We show that averaging the character-level predictions of complementary components improves detection. Combining independently fine-tuned Qwen2.5-VL and InternVL3.5 models yielded our strongest result, reaching 0.50 COR on Italian. In English, adding a claim-verification signal from the base Qwen2.5-VL gave a further but smaller gain.

2 Related Work

Vision-language models hallucinate in various, characteristic ways, describing objects or other visual details absent from the image, often when a strong language prior overrides weak visual evidence (Rohrbach et al., 2018).

* Corresponding author:
ioana.branescu0601@stud.acs.upb.ro.

Object-level evaluation methods include CHAIR, which measured hallucinated object mentions in image captions, and POPE, which probed object hallucination through binary questions about object presence (Rohrbach et al., 2018; Li et al., 2023). Decoding-time methods instead aim to reduce hallucination during generation, for instance through contrastive decoding over original and distorted visual inputs (Leng et al., 2024) or through an over-trust penalty and retrospection-allocation strategy (Huang et al., 2024). Reference-free black-box detection can also exploit consistency across sampled generations, as in SelfCheckGPT (Manakul et al., 2023).

Recent work has moved toward more fine-grained localization. FaithScore decomposed free-form VLM answers into atomic image facts and verified each against the image (Jing et al., 2024). The SHROOM shared tasks followed a similar shift in text-only settings: SHROOM 2024 framed hallucination detection as response-level binary classification (Mickus et al., 2024), whereas Mu-SHROOM introduced multilingual span labeling and highlighted high disagreement between annotators (Vazquez et al., 2025). Most closely related to our multimodal setting, HalLoc provided probabilistic token-level annotations for four hallucination types and introduced a lightweight localizer for token-level hallucination detection (Park et al., 2025).

Our setting requires character-level localization together with explicit error typing. We cast this as sequence labeling over a parameter-efficiently adapted VLM (Hu et al., 2022; Dettmers et al., 2023), using annotator agreement scores as training targets (Uma et al., 2021), and complement the detector with a second component that decomposes the answer into claims and verifies each against the image.

3 Task and Evaluation

SHROOM-Visions (Vázquez et al., 2026) uses the SHEEP dataset (Mickus et al., 2026), which contains image, question, and response triples in four languages: English, French, Italian, and Chinese. Responses are either human-written or model-generated and are annotated at the character level by 3 annotators. Each gold span is assigned one of five error types: invention, mischaracterization, OCR, miscounting, or other. It also carries a probability equal to the fraction of annotators who

marked it. The dataset contains approximately 3.8k labeled training instances and between 1.2k and 1.3k unlabeled test instances per language.

For a response of length L , the gold annotations defined a character-level vector whose values represent annotator agreement. The prediction vector contains the submitted score at each covered character and zero elsewhere. COR is the Spearman correlation between these vectors, computed per answer and averaged across the dataset. CORLBL computes the same correlation separately for the error categories present in either the gold or predicted annotations, so an incorrectly labeled span receives no credit for the corresponding category. The organizers also report IOU, which measures overlap between the predicted and gold hallucinated spans. We optimize for COR and CORLBL and report all three metrics.

4 Method

Figure 1 summarizes our scoring architecture. Token-level predictions are mapped to characters, optionally combined across complementary components, and decoded into spans using a validation-selected threshold. For English, we additionally include an image-grounded claim-verification component that decomposes the answer into atomic claims and checks them against the image. We next describe the adaptation, calibration, and averaging steps, as well as the corresponding experimental setup.

4.1 Adapt: Token-Level Hallucination Scoring

We provide the image, question, and answer to Qwen2.5-VL-7B-Instruct (Bai et al., 2025) in a single forward pass and retain the final-layer hidden states corresponding to the answer tokens. Two shared linear heads are applied to each token representation. The *graded head* predicts a hallucination score $\hat{g}_t \in [0, 1]$, while the *type head* predicts a distribution $\hat{\mathbf{p}}_t$ over 6 classes: *none* and the 5 task error types.

Token-level targets are derived from the character annotations using the tokenizer offsets. The graded target g_t is the maximum gold character score within the interval covered by token t . Its type target y_t is the label of the gold span with the largest overlap with the token, provided that the span covers at least half of it; otherwise, the target is *none*.

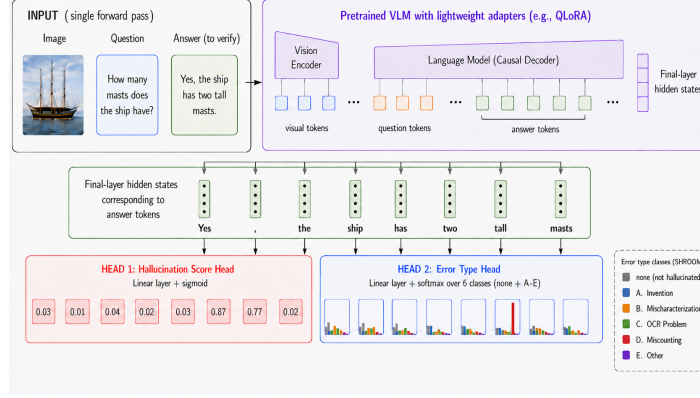


Figure 1: Token-level hallucination scoring with an adapted VLM and two lightweight prediction heads.

For an answer containing T tokens, the training loss is

$$\mathcal{L} = \frac{1}{T} \sum_{t=1}^T \text{BCE}(\hat{g}_t, g_t) + \frac{1}{T} \sum_{t=1}^T \text{CE}_w(\hat{\mathbf{p}}_t, y_t),$$

where CE_w is cross-entropy weighted by the square-root inverse frequency of each of the 6 token classes. The two terms receive equal weight. Both heads are initialized from scratch and trained jointly with QLoRA adapters (Dettmers et al., 2023), while the original backbone weights remain frozen.

We apply the same token-scoring architecture to InternVL3.5-8B (Wang et al., 2025). The training configuration for both backbones is described in Section 4.4.

4.2 Calibrate: From Token Scores to Character Spans

We project the token-level outputs to characters using the tokenizer offsets. Each character inherits the hallucination score of its token and the hard error type label obtained by taking the argmax of the corresponding type distribution. When several components are combined, their character-level hallucination scores are averaged before span construction, as described in Section 4.3.

We mark characters whose score exceeds a threshold τ , merge marked regions separated by at most three characters, and align the resulting boundaries with complete alphanumeric words. This prevents predicted spans from beginning or ending inside a word. Chinese is written without spaces, so there are no word edges to stop at; a Chinese span instead extends outward until it reaches a punctuation mark. Each span is assigned an error type through a score-weighted vote

over its character labels, excluding *none*; *invention* is used as a fallback when no error type receives a vote. We also apply lightweight rule-based overrides for counting and text-reading questions: when the question matches a counting cue (e.g. *how many*), we relabel a span *miscounting* if that type wins more than 30% of the span’s character votes or the span contains a numeral; we relabel *OCR* when the question matches a text cue (e.g. *written*, *says*, *text*) and *OCR* wins more than 30% of the votes.

The threshold τ is selected separately for each language and system configuration on validation data adjusted to match the test proportion of hallucination-free answers, as described in Section 4.4. The threshold value is then fixed for test inference. The effects of word-boundary alignment and adjusted threshold selection are reported in Section 5.2.

4.3 Average: Complementary Components and Character-Level Fusion

Complementary components. In addition to the Qwen2.5-VL detector, we trained an InternVL3.5-8B detector (Wang et al., 2025) using the same token scoring objective. For English, we also used an image-grounded claim verification component. The base Qwen2.5-VL model, without additional fine-tuning, decomposes the answer into atomic factual claims and links each claim to its exact answer substring. Each claim is checked against the image three times, and the fraction of unsupported verdicts is assigned to the corresponding characters.

Character-level averaging. Because the components operate over different tokenizations or answer substrings, we map all hallucination predictions to

Language	Components	COR	CORLBL	IoU
English	Qwen + IVL + CV	.461	.341	.387
French	Qwen	.471	.375	.408
Italian	Qwen + IVL	.505	.368	.421
Chinese	Qwen + IVL	.431	.347	.431
Average		.467	.358	.412

Table 1: Official test results. IVL denotes InternVL3.5, whereas CV stands for claim verification.

the original character positions. Let $s_j^{(m)}$ denote the score assigned by component m to character j . For a combination of M components, we compute

$$\bar{s}_j = \frac{1}{M} \sum_{m=1}^M s_j^{(m)}.$$

The averaged scores are then thresholded and converted into spans as described in Section 4.2. Thus, the components are combined before span construction rather than through spans decoded independently. Error type votes are also pooled across components: for each character, each component contributes its hard type label with a weight equal to its own hallucination score, and the pooled votes determine the type of the resulting span.

4.4 Experimental Setup

We fine-tune a separate model per language. We split the training data into five image-grouped folds shared across languages: three for fine-tuning, one for selecting model variants, component combinations, and decoding thresholds, and one kept untouched during development. Final submissions are retrained on all five folds. Because the test set contains fewer hallucination-free answers, we subsample such examples from the validation fold to match the aggregate test rate before selecting τ . This rate is estimated from a constant no span submission, whose COR equals the proportion of test answers with no gold span; thus, calibration uses one aggregate leaderboard statistic but no per-instance test labels. Further training and reweighting details are given in Appendices B and C.

5 Results

5.1 Official Results

Table 1 reports the official test results. After selecting the configuration for each language, we retrained the submitted models on all 5 training folds.

English test variant	COR	CORLBL
Initial system	.343	.242
Direct character thresholding	.397	.298
+ word boundary alignment	.412	.310
+ adjusted threshold	.445	.325

Table 2: Successive submitted variants of character span construction on the English test set.

Language	Qwen	Added component(s)	Final	Δ
English	.445	CV + IVL	.461	+0.016
Italian	.462	IVL	.505	+0.043
Chinese	.377 [†]	IVL	.431	+0.054

Table 3: Qwen baselines and final complementary-component configurations on the official test set. All values are COR. [†] No standalone Qwen submission on Chinese used the adjusted threshold, so its difference also includes threshold calibration.

Averaged across the four languages, our submissions reach .467 COR, .358 CORLBL, and .412 IoU. Italian gives our strongest result, reaching .505 COR and ranking sixth among the competing teams. Across languages, the lower CORLBL scores indicate that assigning the correct error type remains more difficult than locating the hallucinated span.

5.2 Component Analysis

Calibration and Span Construction. Table 2 compares four successive English test submissions. Direct character thresholding provides the largest gain, while word-boundary alignment and adjusted threshold selection further raise COR from .343 to .445 and CORLBL from .242 to .325.

Complementary Components. Table 3 compares the Qwen baselines with the final configurations that included complementary components. On French, adding InternVL3.5 gave no held-out gain, so the submission uses Qwen alone and the language is omitted from the table.

Italian showed the cleanest gain, where a second fine-tuned backbone at the same calibration adds +.043; the larger Chinese figure partly reflects its uncalibrated baseline, whereas English exhibited little gain. As all results come from single runs, the small English and Chinese differences are suggestive rather than conclusive; only the Italian +.043 is sizable.

Claim verification illustrated complementarity between individually weak and strong components. On English, it reached only .278 COR on its own,

yet averaging it with the calibrated Qwen detector still adds a small gain (.445 to .452), and the full three-component system average reached .461.

Error Analysis. In English, type errors mainly confused *invention* with *mischaracterization*, while *miscounting* was the most reliable category (see Figure 2 in Appendix D).

6 Conclusions

Our submissions reach an average COR of .467 across four languages, with the strongest result of .505 on Italian obtained by combining Qwen2.5-VL and InternVL3.5. Our error analysis suggests two natural directions for future work. First, error typing could be performed in a second stage conditioned on the detected hallucination span, rather than being predicted jointly at the token level and aggregated during span construction; this may help reduce the prominent invention/mischaracterization confusion and narrow the gap between COR and CORLBL. Second, reporting bootstrap confidence intervals and results across multiple training seeds would provide stronger evidence for the smaller component-level differences.

Limitations

We report one training run per configuration and use a single validation fold for system selection, without cross-validation or confidence intervals. A separate fold is kept untouched during these choices. Threshold selection used the aggregate proportion of hallucination-free test answers, estimated through one constant no span submission per language. This provides no information about individual test examples, but it still uses a statistic derived from the test leaderboard. In addition, not every component was submitted individually on the official test set, so the ensemble analysis primarily measures improvements over the Qwen2.5-VL detector. Finally, the experiments cover one shared task dataset and 4 languages, whereas the extent to which the method transfers to other VLMs, domains, and annotation schemes remains to be established.

Use of AI Assistants

We used AI-based coding and writing assistants during this work to help implement and debug the experimental pipeline, assist with data analysis, and draft and edit the paper. All research questions,

experimental design, methodological choices, and verification of results were carried out by the authors, who take full responsibility for the content.

References

- Shuai Bai et al. 2025. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*.
- Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient finetuning of quantized LLMs. In *NeurIPS*.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *ICLR*.
- Qidong Huang, Xiaoyi Dong, Pan Zhang, Bin Wang, Conghui He, Jiaqi Wang, Dahua Lin, Weiming Zhang, and Nenghai Yu. 2024. OPERA: Alleviating hallucination in multi-modal large language models via over-trust penalty and retrospection-allocation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Liqliang Jing, Ruosen Li, Yunmo Chen, and Xinya Du. 2024. FaithScore: Fine-grained evaluations of hallucinations in large vision-language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 5042–5063.
- Sicong Leng, Hang Zhang, Guanzheng Chen, Xin Li, Shijian Lu, Chunyan Miao, and Lidong Bing. 2024. Mitigating object hallucinations in large vision-language models through visual contrastive decoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Evaluating object hallucination in large vision-language models. In *Proceedings of EMNLP*.
- Potsawee Manakul, Adian Liusie, and Mark J. F. 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 (EMNLP)*.
- Timothee Mickus, Claudio Savelli, Eduardo Calò, Emilio Raimond, Stella Frank, Hengyu Luo, Flavio Giobergia, Vincent Segonne, Chuyuan Li, Aman Sinha, Lorenzo Vaiani, Jörg Tiedemann, and Raúl Vázquez. 2026. Can humans dream of electric sheep? human-written samples for fine-grained vision-and-language hallucination benchmarking. *arXiv preprint arXiv:2608.01021*.
- Timothee Mickus, Elaine Zosa, Raul Vazquez, Teemu Vahtola, Jörg Tiedemann, Vincent Segonne, Alessandro Raganato, and Marianna Apidianaki. 2024. SemEval-2024 task 6: SHROOM, a shared-task on

hallucinations and related observable overgeneration mistakes. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*.

Eunkyu Park, Minyeong Kim, and Gunhee Kim. 2025. HalLoc: Token-level localization of hallucinations for vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 29893–29903.

Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. 2018. Object hallucination in image captioning. In *Proceedings of EMNLP*.

Alexandra N. Uma, Tommaso Fornaciari, Dirk Hovy, Silviu Paun, Barbara Plank, and Massimo Poesio. 2021. Learning from disagreement: A survey. *Journal of Artificial Intelligence Research*, 72.

Raul Vazquez, Timothee Mickus, Elaine Zosa, Teemu Vahtola, Jörg Tiedemann, Aman Sinha, Vincent Segonne, Fernando Sanchez-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: MUSHROOM, the multilingual shared-task on hallucinations and related observable overgeneration mistakes. In *Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)*, pages 2472–2497.

Raúl Vázquez, Aman Sinha, Chuyuan Li, Artem Shelmanov, Artem Vazhentsev, Claudio Savelli, Eduardo Calò, Emilio Raimond, Stella Frank, Hengyu Luo, Flavio Giobergia, Vincent Segonne, Lorenzo Viani, Jörg Tiedemann, and Timothee Mickus. 2026. [Overview of SHROOM-Visions 2026: A shared task on hallucination detection in large vision-language models](#). *Preprint*, arXiv:2608.25662.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, et al. 2025. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*.

A Claim-verification Prompts

The claim-verification component described in Section 4.3 uses two prompts, reproduced verbatim below. Step 1 is a text-only turn; step 2 shows the image and fact-checks all of one answer’s claims in a single turn (`{claim}`, `{quote}` and `{claims_block}` are runtime placeholders). Verdicts are sampled $K=3$ times at temperature 0.7; the fraction of unsupported verdicts becomes the score of the characters of the claim’s `error_text` (falling back to its `quote`); a span fires at ≥ 0.67 .

Step 1: decompose (text only).

You are decomposing a model-generated answer into atomic factual claims so each can be fact-checked against an image.

Split the ANSWER below into its smallest self-contained factual claims. For each claim, also give the EXACT verbatim substring of the ANSWER it is based on (copy it character-for-character, do not paraphrase). Return ONLY strict JSON: a list of objects `[{"claim": "<atomic claim>", "quote": "<verbatim substring of the answer>"}]`. No prose, no markdown.

ANSWER:

Step 2: verify (image shown, all claims of one answer).

Look at the image and fact-check EACH numbered claim below against it. `{claims_block}`

Reply with ONLY strict JSON: a list of objects, EXACTLY ONE per claim, in the SAME order:

```
[{"verdict": "supported" | "unsupported" | "uncertain", "type": "invention" | "mischaracterization" | "OCR" | "miscounting" | "other", "error_text": "<string>"}]
```

Use verdict "unsupported" if the claim is false or not shown in the image. Set "type" to the kind of error only when unsupported (else "other"): invention = a thing/attribute not in the image; mischaracterization = a real thing described wrongly; OCR = misread text; miscounting = wrong quantity; other = anything else. Set "error_text" to the SHORTEST exact substring of that claim’s quoted text that is factually wrong -- copy it verbatim, just the wrong words (a name, number, or attribute); empty string if supported. No prose.

A single-claim variant of step 2 (one claim per image turn) exists in the released code and produced equivalent verdicts; the batched form above is what generated our submissions.

B Hyperparameters

Both backbones use the same recipe, summarized in Table 4. The two heads (graded and type) run in fp32 on top of the last-layer hidden states of size H (3584 for Qwen2.5-VL-7B, 4096 for InternVL3.5-8B). The decoding threshold τ is not fixed globally: for each language and system, we select it on the reweighted validation fold (Appendix C).

Setting	Value
Base quantization	4-bit NF4, double-quant, bf16
LoRA r / α / dropout	16 / 32 / 0.05
LoRA targets	attn. + MLP (LM only)
Trainable params	47.6M / 43.6M (0.57/0.51%)
Heads	Linear(H , 1), Linear(H , 6)
Optimizer	AdamW, lr 10^{-4}
Epochs / accum. / batch	3 / 8 / 1
Decode threshold τ	[0.2, 0.4], per language
Gap / snapping	≤ 3 chars / to words

Table 4: Fine-tuning and decoding hyperparameters, shared by both backbones (Qwen2.5-VL and InternVL3.5); percentages are of the full model and the vision tower is frozen.

C Reweighting Details

The test set contains many fewer hallucination-free answers than the training data. We measured the test empty-answer rate per language with a single constant “no span” submission (EN 15.5%, FR 16.8%, IT 18.4%, ZH 25.5%), against training rates of 25.3%, 25.6%, 26.3%, and 36.5%. Before selecting any hyperparameter, we uniformly at random subsample each fold’s empty rows (fixed seed) to match the test rate, keeping all non-empty rows, and then select thresholds on the reweighted validation fold, keeping the remaining fold untouched during model selection. The reweighting lowers the selected τ markedly, for example from 0.4 to 0.25 on English and from 0.5 to 0.4 on Italian.

D Error-Type Confusion

Figure 2 reports the English error-type confusion matrix on the untouched fold.

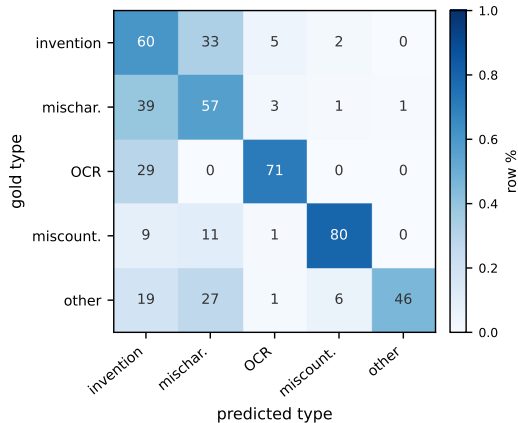


Figure 2: English error-type confusion on the untouched fold, row-normalized (each row sums to 100%). Most label errors are confusions between *invention* and *mis-characterization*.

E Leaderboard Placement

Table 5 gives our per-language ranks on the three official leaderboards (COR, CORLBL, and IOU). We place mid-field across languages and obtain our strongest result on Italian, where we rank 6th by COR, 7th by IOU, and 9th by CORLBL. This difference suggests that the system localizes hallucinated spans more reliably than it assigns their error types. On the other three languages, the ranks are similar across metrics.

Rank (our team)	COR	CORLBL	IOU
EN	8/26	8/26	10/26
FR	9/19	9/19	10/19
IT	6/19	9/19	7/19
ZH	13/20	12/20	10/20

Table 5: Our placement among competing teams (baseline entries excluded) on each of the three leaderboards.