

MSA at SemEval-2025 Task 3: High Quality Weak Labeling and LLM Ensemble Verification for Multilingual Hallucination Detection

Baraa Hikal, Ahmed Nasreldin, Ali Hamdi

Faculty of Computer Science, MSA University, Egypt

{baraa.moaweya, ahmed.nasreldin, ahamdi}@msa.edu.eg

Abstract

This paper describes our submission for SemEval-2025 Task 3: Mu-SHROOM, the Multilingual Shared-task on Hallucinations and Related Observable Overgeneration Mistakes (Vázquez et al., 2025). The task involves detecting hallucinated spans in text generated by instruction-tuned Large Language Models (LLMs) across multiple languages. Our approach combines task-specific prompt engineering with an LLM ensemble verification mechanism, where a primary model extracts hallucination spans and three independent LLMs adjudicate their validity through probability-based voting. This framework simulates the human annotation workflow used in the shared task validation and test data. Additionally, fuzzy matching refines span alignment. Our system ranked 1st in Arabic and Basque, 2nd in German, Swedish, and Finnish, and 3rd in Czech, Farsi, and French.

1 Introduction

Large Language Models (LLMs) are highly effective in generating text; however, they sometimes produce hallucinations—misleading content that is not properly grounded in the input data (Huang et al., 2025). Identifying these spans is essential for improving the reliability of LLM-generated outputs in translation, summarization, and conversational AI (Alaharju, 2024). SemEval-2025 Task 3: Mu-SHROOM tackles this challenge by presenting a multilingual benchmark for detecting character-level hallucinations across multiple languages. The task involves detecting hallucinated spans in instruction-tuned LLM outputs, presenting challenges in language diversity, annotation consistency, and accurate span localization. (Sriramanan et al., 2025)

To tackle this challenge, our system utilizes a hybrid approach that integrates task-specific prompt

engineering for weak label generation with an LLM ensemble verification mechanism (Hikal et al., 2025). Our methodology follows a multi-step adjudication process in which a primary LLM identifies hallucination spans, and three independent LLMs subsequently verify their validity through a probability-based voting mechanism (Kang et al., 2024b). Additionally, we apply fuzzy matching techniques to improve the alignment of hallucination spans with ground truth annotations, thereby enhancing detection accuracy (Chaudhuri et al., 2003).

By participating in this task, we gained insights into language-specific hallucination challenges and the strengths and limitations of LLM-based verification. Certain LLMs demonstrated closer alignment with human annotations, while hallucination patterns varied significantly, particularly in morphologically rich languages where annotation ambiguity was higher (Abdelrahman, 2024). Our results indicate that ensemble verification and span refinement substantially improve hallucination detection, offering a robust approach for mitigating LLM hallucinations in multilingual settings.¹

2 Related Work

Hallucination detection in Large Language Models (LLMs) has been studied in machine translation, text summarization, and conversational AI (Ji et al., 2023). Earlier approaches primarily relied on sentence-level classification, whereas recent research has transitioned to span-level detection for greater precision (Joshi et al., 2020). Self-consistency verification and knowledge-grounded approaches have improved hallucination identification, but many depend on external data, limiting their applicability in multilingual settings. (Mehta et al., 2024)

Multilingual NLP models struggle with hallucinations, especially in low-resource languages

¹<https://github.com/baraahekal/mu-shroom>

where confidence scores are unreliable (Kang et al., 2024a). Morphologically rich languages introduce additional challenges due to intricate annotation inconsistencies (Tsarfaty et al., 2013). Prior work on translation-based verification has attempted to address this, but these approaches are ineffective in zero-shot scenarios (Nie, 2022).

Ensemble verification methods enhance detection accuracy by utilizing multiple models. Approaches such as multi-agent verification and cross-model adjudication have proven effective in assessing LLM outputs (Liu and Wang, 2024). Our system expands on these approaches by integrating weak label generation with an ensemble verification pipeline, while also utilizing fuzzy matching to improve span alignment. Unlike previous methods that rely on single-model hallucination detection, our approach leverages an ensemble of LLMs for adjudication, reducing model bias and improving hallucination span refinement via fuzzy matching.

3 System Overview

Our hallucination detection approach integrates task-specific prompt engineering, an LLM ensemble verification mechanism, and post-processing refinements. The system is composed of three key components: fine-tuned prompt construction, hallucination span verification through LLM ensembles, and post-processing with fuzzy matching. An overview of the full pipeline is illustrated in Figure 1.

3.1 Prompt Engineering for Weak Label Generation

We analyzed the validation dataset to extract annotator instructions and identify patterns, enabling the construction of a fine-grained prompt with few-shot examples. Iterative refinement improved extraction accuracy. Detailed prompt in Appendix A.

3.2 Selection of State-of-the-Art LLMs

Building on the insights from the Vectara LLM Report, we chose Gemini-2.0-Flash-Exp, Qwen-2.5-Max (Yang et al., 2024), GPT-4o (OpenAI, 2024), and DeepSeek-V3 (Liang and et al., 2024) as our primary models for hallucination detection. These models were selected for their strong factual accuracy and reliable generation capabilities, ensuring consistent performance across multiple languages. Figure 2 illustrates the model rankings from the report.

3.3 LLM Ensemble Verification Mechanism

Our hallucination detection pipeline utilizes a multi-stage ensemble verification process. With four selected LLMs—Gemini-2.0-Flash-Exp, Qwen-2.5-Max, GPT-4o, and DeepSeek-V3—we systematically rotate through different configurations, where one model identifies hallucinated spans while the other three act as adjudicators. This setup is inspired by the Mu-SHROOM annotation process, where multiple human annotators reviewed and adjudicated hallucination spans in the validation and test datasets. By simulating this human adjudication process with LLMs, we aim to improve label consistency and mitigate annotation biases.

Span Extractor Model (SEM) A primary LLM identifies hallucinated spans by analyzing question-answer pairs. Given a question Q and an answer A , the span extractor outputs candidate hallucination spans $S = \{s_1, s_2, \dots, s_k\}$:

$$S = \text{LLM}_{\text{extract}}(Q, A, \text{prompt})$$

Voting Adjudicator Models (VAMs) The three remaining LLMs act as adjudicators, independently assessing each span $s_i \in S$ and assigning a hallucination probability score:

$$p_{ij} = M_j(s_i, Q), \quad p_{ij} \in [0, 1]$$

where M_j represents an adjudicator LLM.

Iterative Model Rotation: This process is repeated for all possible combinations of the four models, ensuring that each model serves as the span extractor exactly once, while the other three act as adjudicators. Given four models, this results in a total of four unique verification runs.

Consensus-Based Labeling (CBL): The final hallucination probability for each span is determined by aggregating the probabilities across all verification runs:

$$p_i = \frac{1}{N} \sum_{j=1}^N p_{ij}$$

where $N = 3$ is the number of adjudicator models per run. The final hallucination label is assigned using a majority voting scheme across all runs. A span is classified as hallucinated if:

$$p_i \geq 0.7$$

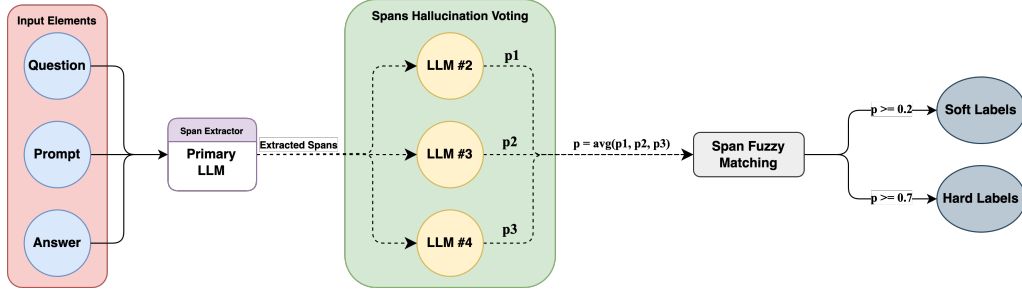


Figure 1: Overview of our hallucination detection pipeline.

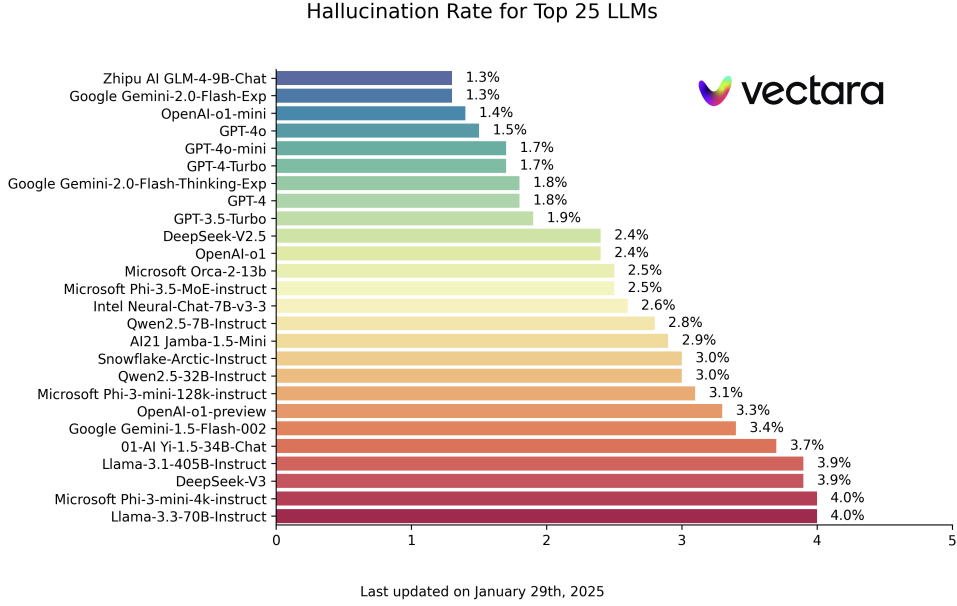


Figure 2: Performance rankings of LLMs according to the Vectara Hallucination Leaderboard (Vectara, 2024).

The threshold $\tau = 0.7$ was chosen based on empirical observations on the validation set. During tuning, we found that lower thresholds (e.g., 0.5) tended to produce too many false positives by labeling uncertain spans as hallucinations, while higher thresholds (e.g., 0.8) missed subtle hallucinations annotated by human reviewers. A threshold of 0.7 offered the best trade-off between precision and recall, and its behavior closely matched the annotation patterns observed in the Mu-SHROOM validation data (Vázquez et al., 2025).

This iterative model selection ensures robustness by reducing individual model biases and leveraging diverse perspectives from different LLMs.

3.4 Post-Processing with Fuzzy Matching

LLMs frequently introduce minor inconsistencies in span extraction, such as variations in capitalization, extra spaces, or incomplete word boundaries. To minimize these errors, we use fuzzy matching with a similarity threshold of 0.9 (partial ratio). The

similarity score between a predicted span s_i and a ground truth span g_j is given by:

$$\text{Similarity}(s_i, g_j) = 1 - \frac{\text{Lev}(s_i, g_j)}{\max(|s_i|, |g_j|)}$$

where $\text{Lev}(s_i, g_j)$ is the Levenshtein distance. If $\text{Similarity}(s_i, g_j) \geq 0.9$, the span is considered correctly aligned.

3.5 Algorithm Implementation

Our pipeline follows a multi-stage verification process where a primary LLM extracts candidate hallucination spans, and three adjudicator models verify them using probability-based voting. Fuzzy matching refines span alignment, improving precision. This ensemble approach mitigates model bias and enhances robustness.

Algorithm 1 in Appendix B outlines the full process.

| Lang | Question | Answer |
|---------|--|---|
| English | What did Petra van Staveren win a gold medal for? | Petra van Stoveren won a silver medal in the 2008 Summer Olympics in Beijing, China . |
| French | En quelle année Childebert IV est-il devenu roi des Francs? | Childebert IV est devenu roi des Francs en 743 , après la mort de son père, Charles Martel . Il a régné jusqu'à sa mort en 751 . |
| German | Wie viele Zuschauer haben Platz im Gillette Stadium in Foxborough? | Das Gillette Stadium ist mit einer Kapazität von rund 50.000 Zuschauern das größte Stadion in der NFL . |
| Arabic | أين تقع إسبيرانس؟ | إسبيرانس توجد في جنوب أفريقيا، بالتحديد في ولاية نايلدر في منطقة الكاليفورنيا الجنوبية. هي معروفة بمنطقة الجبال والطبيعة الخلابة. |
| Arabic | بأي يد كانت تلعب كيكو موريجامي؟ | كيكوك موريجامي كانت تقود بالقدم اليمنى. |

Figure 3: Dataset examples in different languages. The hallucinated span(s) are highlighted.

4 Experiments and Results

4.1 Dataset

Our system was evaluated on the Mu-SHROOM dataset from SemEval-2025 Task 3. We leveraged only the validation and test sets, using the validation set for prompt refinement and the test set for final evaluation. Unlike traditional supervised approaches, we did not use the training set for model learning. Instead, we employed prompt-based weak labeling and an ensemble verification mechanism (Smith et al., 2024). The test set contained unlabeled examples, and final system evaluation was conducted by the task organizers.

Figure 3 presents dataset examples in different languages, highlighting hallucinated spans.

4.2 Evaluation Metrics

We evaluated our system using the official Mu-SHROOM metrics:

- **Intersection-over-Union (IoU):** Measures the overlap between predicted and gold hallucinated spans (Rezatofighi et al., 2019).
- **Probability Correlation (Corr):** Evaluates the correlation between predicted hallucination probabilities and human annotations (Sheugh and Alizadeh, 2015).

The IoU score for a predicted span s_p and a ground truth span s_g is computed as:

$$\text{IoU} = \frac{|s_p \cap s_g|}{|s_p \cup s_g|}$$

where $|s_p \cap s_g|$ represents the overlapping characters, and $|s_p \cup s_g|$ is the total number of unique characters in both spans.

4.3 Results

As each of the four LLMs alternates as the span extractor while the others act as adjudicators, we report results for each combination. The tables [1,2,3,4] show performance across languages.

| Lang | IoU Score | Probability Corr |
|------|--------------|------------------|
| AR | 0.576 | 0.536 |
| EU | 0.604 | 0.611 |
| DE | 0.526 | 0.567 |
| SV | 0.607 | 0.401 |
| FI | 0.587 | 0.501 |
| CS | 0.396 | 0.410 |
| FA | 0.540 | 0.511 |
| FR | 0.571 | 0.507 |
| EN | 0.506 | 0.538 |
| IT | 0.484 | 0.545 |
| HI | 0.684 | 0.725 |

Table 1: Performance when Qwen-2.5-Max acts as the span extractor.

| Lang | IoU Score | Probability Corr |
|------|--------------|------------------|
| AR | 0.669 | 0.648 |
| EU | 0.612 | 0.620 |
| DE | 0.601 | 0.547 |
| SV | 0.636 | 0.422 |
| FI | 0.625 | 0.521 |
| CS | 0.507 | 0.552 |
| FA | 0.669 | 0.679 |
| FR | 0.619 | 0.555 |
| EN | 0.531 | 0.519 |
| IT | 0.712 | 0.737 |
| HI | 0.662 | 0.690 |

Table 2: Performance when Gemini-2.0-Flash-Exp acts as the span extractor.

| Lang | IoU Score | Probability Corr |
|------|--------------|------------------|
| AR | 0.637 | 0.593 |
| EU | 0.604 | 0.611 |
| DE | 0.527 | 0.531 |
| SV | 0.610 | 0.398 |
| FI | 0.619 | 0.527 |
| CS | 0.432 | 0.486 |
| FA | 0.639 | 0.700 |
| FR | 0.601 | 0.485 |
| EN | 0.525 | 0.502 |
| IT | 0.736 | 0.756 |
| HI | 0.621 | 0.664 |

Table 3: Performance when GPT-4o acts as the span extractor.

| Lang | IoU Score | Probability Corr |
|------|--------------|------------------|
| AR | 0.658 | 0.644 |
| EU | 0.607 | 0.585 |
| DE | 0.613 | 0.610 |
| SV | 0.624 | 0.417 |
| FI | 0.642 | 0.546 |
| CS | 0.465 | 0.507 |
| FA | 0.632 | 0.671 |
| FR | 0.572 | 0.539 |
| EN | 0.529 | 0.487 |
| IT | 0.703 | 0.716 |
| HI | 0.659 | 0.697 |

Table 4: Performance when DeepSeek-V3 acts as the span extractor.

| Lang | Span Extractor | IoU | Corr | Rank |
|------|----------------------|-------|-------|------|
| AR | Gemini-2.0-Flash-Exp | 0.669 | 0.648 | 1/32 |
| EU | Gemini-2.0-Flash-Exp | 0.612 | 0.620 | 1/26 |
| DE | DeepSeek-V3 | 0.613 | 0.610 | 2/31 |
| SV | Gemini-2.0-Flash-Exp | 0.636 | 0.422 | 2/30 |
| FI | DeepSeek-V3 | 0.642 | 0.546 | 2/30 |
| CS | Gemini-2.0-Flash-Exp | 0.507 | 0.552 | 3/26 |
| FA | Gemini-2.0-Flash-Exp | 0.669 | 0.679 | 3/26 |
| FR | Gemini-2.0-Flash-Exp | 0.619 | 0.555 | 3/33 |
| IT | GPT-4o | 0.736 | 0.756 | 4/31 |
| HI | Qwen-2.5-Max | 0.684 | 0.725 | 5/27 |
| EN | Gemini-2.0-Flash-Exp | 0.531 | 0.519 | 6/44 |

Table 5: Best performance per language, with span extractor and final rank.

Our system outperformed other methods in Arabic and Basque, where annotation consistency was higher. However, performance dropped in English, likely due to increased annotation variability—English had up to 12 different annotators per sample (Vázquez et al., 2025) leading to inconsistencies.

4.4 Discussion

Our system effectively detects hallucinated spans across multiple languages by using ensemble ver-

ification to reduce model bias and fuzzy matching to refine span alignment. However, challenges remain—especially in dealing with annotation inconsistencies and ambiguous hallucinations, which tend to be more common in morphologically complex languages.

A key finding is that different LLMs vary in their alignment with human annotations, indicating that task-specific fine-tuning or alternative verification strategies could further improve detection accuracy. Additionally, improving span refinement techniques beyond fuzzy matching may reduce boundary mismatches and improve character-level precision.

5 Conclusion

We presented our system for *SemEval-2025 Task 3: Mu-SHROOM*, focusing on hallucinated span detection in LLM-generated text across multiple languages. Our approach combines prompt-engineered weak label generation with an LLM ensemble verification mechanism, demonstrating strong performance in multilingual hallucination detection.

Our results confirm the effectiveness of ensemble-based adjudication, ranking among the top systems in several languages. However, challenges such as annotation variability and morphological complexity highlight areas for further refinement.

Future work could focus on integrating external knowledge for hallucination verification, fine-tuning LLMs to better align with human annotations, and refining span localization techniques to enhance character-level precision. These improvements could further advance hallucination detection in multilingual NLP systems.

References

- Mostafa Abdelrahman. 2024. Hallucination in low-resource languages: Amplified risks and mitigation strategies for multilingual llms. *Journal of Applied Big Data Analytics, Decision-Making, and Predictive Modelling Systems*, 8(12):17–24.
- Henri Alaharju. 2024. Ensuring performance and reliability in llm-based applications: A case study.
- Surajit Chaudhuri, Kris Ganjam, Venkatesh Ganti, and Rajeev Motwani. 2003. Robust and efficient fuzzy match for online data cleaning. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 313–324.

- Baraa Hikal, Ahmed Nasreldin, Ali Hamdi, and Ammar Mohammed. 2025. Few-shot optimized framework for hallucination detection in resource-limited nlp systems. *arXiv preprint arXiv:2501.16616*.
- Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.
- Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38.
- Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the association for computational linguistics*, 8:64–77.
- Haoqiang Kang, Terra Blevins, and Luke Zettlemoyer. 2024a. Comparing hallucination detection metrics for multilingual generation. *arXiv preprint arXiv:2402.10496*.
- Inwon Kang, William Van Woensel, and Oshani Seneviratne. 2024b. Using large language models for generating smart contracts for health insurance from textual policies. In *AI for Health Equity and Fairness: Leveraging AI to Address Social Determinants of Health*, pages 129–146. Springer.
- Wenfeng Liang and et al. 2024. Deepseek-v3 technical report. <https://arxiv.org/abs/2412.19437>. Accessed: 2025-04-26.
- Haoyang Liu and Haohan Wang. 2024. Genotex: A benchmark for evaluating llm-based exploration of gene expression data in alignment with bioinformaticians. *arXiv preprint arXiv:2406.15341*.
- Rahul Mehta, Andrew Hoblitzell, Jack O’keefe, Hyeju Jang, and Vasudeva Varma. 2024. Halu-nlp at semeval-2024 task 6: Metacheckgpt-a multi-task hallucination detection using llm uncertainty and meta-models. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 342–348.
- Ercong Nie. 2022. Zero-shot learning on low-resource languages by cross-lingual retrieval. *Masterarbeit im Studiengang Computerlinguistik an der Ludwig-Maximilians-Universität München Fakultät für Sprach-und Literaturwissenschaften*.
- OpenAI. 2024. Gpt-4o system card. <https://arxiv.org/html/2410.21276v1>. Accessed: 2025-04-26.
- Hamid Rezaatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. 2019. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 658–666.
- Leily Sheugh and Sasan H Alizadeh. 2015. A note on pearson correlation coefficient as a metric of similarity in recommender system. In *2015 AI & Robotics (IRANOPEN)*, pages 1–6. IEEE.
- Ryan Smith, Jason A Fries, Braden Hancock, and Stephen H Bach. 2024. Language models in the loop: Incorporating prompting into weak supervision. *ACM/JMS Journal of Data Science*, 1(2):1–30.
- Gaurang Sriramanan, Siddhant Bharti, Vinu Sankar Sadasivan, Shoumik Saha, Priyatham Kattakinda, and Soheil Feizi. 2025. Llm-check: Investigating detection of hallucinations in large language models. *Advances in Neural Information Processing Systems*, 37:34188–34216.
- Reut Tsarfaty, Djamé Seddah, Sandra Kübler, and Joakim Nivre. 2013. Parsing morphologically rich languages: Introduction to the special issue. *Computational linguistics*, 39(1):15–22.
- 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: MUSHROOM, the multilingual shared-task on hallucinations and related observable overgeneration mistakes**.
- Vectara. 2024. **Hallucination leaderboard**. GitHub repository commit.
- An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024. Qwen2.5 technical report. <https://arxiv.org/abs/2412.15115>. Accessed: 2025-04-26.

A Appendix: Instruction Prompt template for Extraction and Annotation

Question & Answer Pair

i) **Question:** model-input

ii) **Answer:** model-output-text

Task Description

You are a professional annotator and {entry[lang]} linguistic expert. Your job is to detect and extract hallucination spans from the provided answer compared to the question.

Exact Span Matching

Extract spans word-for-word and character-for-character exactly as they appear in the answer. Ensure perfect alignment, including punctuation, capitalization, and spacing. If a span is partially supported, only extract the unsupported portion. Preserve original numeral formats: Persian/Arabic numerals must remain in their native script.

Minimal Spans

Select the smallest possible spans that, when removed, completely eliminate the hallucination. Prioritize precision: Avoid extracting entire sentences if a shorter phrase accurately captures the hallucination. Ensure the extracted span exclusively contains hallucinated content without removing valid information.

Hallucination Definition

Any phrase, entity, number, or fact that is not supported by the question. Any exaggeration or overly specific detail absent in the question. Incorrect names, locations, numbers, dates, or causes. In yes/no questions, unsupported answers (e.g., "Yes", "No") and speculative details.

Soft and Hard Labels

Assign probabilities [0.0 - 1.0] for soft labels based on hallucination confidence. Include spans with ≥ 0.7 probability in hard labels.

B Appendix: Our Proposed Framework

Algorithm 1 Hallucination Detection Pipeline

Require: Question Q , Answer A , LLM ensemble $\{M_1, M_2, M_3\}$, threshold $\tau = 0.7$

Ensure: Set of hallucinated spans S^*

1: $S \leftarrow \text{LLM}_{\text{extract}}(A, Q, \text{prompt})$

2: **for** each $s_i \in S$ **do**

3: **Compute hallucination scores:**

4: $p_{ij} = M_j(s_i, Q), \quad \forall M_j$

5: **Compute final probability:**

6: $p_i = \frac{1}{N} \sum_{j=1}^N p_{ij}$

7: **if** $p_i \geq \tau$ **then**

8: **Add hallucinated span to refined set:**

9: $S' \leftarrow S' \cup \{s_i\}$

10: **end if**

11: **end for**

12: **Apply fuzzy matching for span refinement:**

13: $S^* \leftarrow \text{FuzzyMatch}(S', \text{Ground Truth}, 0.9)$

14: **Return** S^*
