HalluRAG-RUG at SemEval-2025 Task 3: Using Retrieval-Augmented Generation for Hallucination Detection in Model Outputs

Silvana Abdi
s.abdi.4@student.rug.nl
University of Groningen

Mahrokh Hassani
m.hassani@student.rug.nl
University of Groningen

Rosalien Kinds
r.a.kinds@student.rug.nl
University of Groningen

Timo Strijbis
t.strijbis@student.rug.nl
University of Groningen

Roman Terpstra
r.p.terpstra@student.rug.nl
University of Groningen

Abstract

Large Language Models (LLMs) suffer from a critical limitation: hallucinations, which refer to models generating fluent but factually incorrect text. This paper presents our approach to hallucination detection in English model outputs as part of the SemEval-2025 Task 3 (*Mu-SHROOM*). Our method, HalluRAG-RUG, integrates Retrieval-Augmented Generation (RAG) using Llama-3 and prediction models using token probabilities and semantic similarity. We retrieved relevant factual information using a named entity recognition (NER)-based Wikipedia search and applied abstractive summarization to refine the knowledge base. The hallucination detection pipeline then used this retrieved knowledge to identify inconsistent spans in model-generated text. This result was combined with the results of two systems, which identified hallucinations based on token probabilities and low-similarity sentences. Our system placed 33rd out of 41, performing slightly below the 'mark all' baseline but surpassing the 'mark none' and 'neural' baselines with an IoU of 0.3093 and a correlation of 0.0833.

1 Introduction

The rise of Large Language Models (LLMs) has brought attention to an important limitation they have, a phenomenon often referred to as LLM 'hallucinations'. This phenomenon occurs when an AI-generated text contains or describes facts that are not supported by the provided reference. These facts do not necessarily need to be false to be labeled a hallucination. Instead, they are cases where the answer text is more specific than it should be, given the information available in the provided context. To further clarify what a hallucination is, we provide the following example introduced by Dopierre et al. (2021):

- **Source Text**: *I am not sure where my phone is.*
- **Model-Generated Paraphrase**: *How can I find the location of any Android mobile?*

As seen in this example, the generated text is fluent but inaccurate concerning the source text. This is noted by the generation of information that is not found originally in the source text, specifically referring to 'Android mobile'.

The generation of false information can hinder a model's usefulness in many applications. Moreover, it can also be the cause for ethical concerns: when a text is syntactically sound, people quickly assume that it is also semantically sound. A user being presented with false information can cause considerable harm in many different domains.

The detection of hallucinations is an important task in improving model trustworthiness, so it is vital to develop and improve methods of hallucination detection. In this context, SemEval-2025 Task 3: *Mu-SHROOM, the Multilingual Shared-task on Hallucinations and Related Observable Overgeneration Mistakes* was organized (Vázquez et al., 2025)[1]. This year, the task inquires about the exact text spans in which hallucinations occur, as opposed to last year's binary classification task (Mickus et al., 2024). The organizers provided validation data in ten different languages, from which we only considered English.

Our approach to this particular task implements a combined model that implements Retrieval-Augmented Generation (RAG) in combination with factual information from Wikipedia and Llama-3, as well as supplementary prediction models based on token probabilities and low-similarity sentences. Our method achieved place 33 out of 41 for the English track in SemEval-2025 Task 3, performing slightly below the 'mark all' baseline.

[1] Link to the official Shared Task website: https://helsinki-nlp.github.io/shroom/

Proceedings of the The 19th International Workshop on Semantic Evaluation (SemEval-2025), pages 846–851
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

2 Related work

First, we will discuss some related work that was done during the previous iteration of this shared task, SemEval-2024 Task 6 (Mickus et al., 2024). One of the most common techniques across the papers written on this task is the use of transformer-based models and semantic similarity measures. Markchom et al. (2024) employed SentenceTransformers to generate embeddings for hypothesis and target texts, comparing them via cosine similarity. The aim was to detect hallucinations based on low similarity scores.

Other groups pivoted towards prompt-based methods. Borra et al. (2024) focused on zero-shot and few-shot learning. In zero-shot learning, the model relied on its pre-trained knowledge, using carefully crafted prompts. Few-shot learning incorporated a small set of labeled data, improving the model's ability to detect more subtle hallucinations. A large number of groups that participated used similar models: models like Vectara, Mistral/Mixtral, DeBERTa, and GPT (3.5 or 4) were used very commonly. The organizers note that especially the GPT-based models work well: four out of six top-scoring teams incorporated it in their approach (e.g., Mehta et al. (2024); Obiso et al. (2024)).

Outside of the context of the Shared Task, much work has been done regarding the use of RAG in hallucination reduction. The main intuition behind this approach is that the inclusion of factual information (the 'Retrieval' part) will reduce the generation of factually incorrect content (Gao et al., 2024). The use of RAG for hallucination reduction has been proven to be effective for multiple use cases, like conversation (Shuster et al., 2021) or structured outputs like workflow generation (Ayala and Bechard, 2024). Considering RAG's usefulness in hallucination reduction, it will be interesting to see whether the addition of relevant retrieved data also extrapolates to improved hallucination detection. This approach is not well-represented in the literature yet, so the merits of the method are yet to be seen.

3 Data

The Shared Task data was provided in 14 languages total, but for our approach, only the English data was considered.

The organizers of the shared task released both a validation set as well as a test set. The English validation set comprised 50 data points, while the test set, released at the start of the evaluation phase (initially without labels), comprised 154 data points. Each of the data points consists of the following elements:

- **ID**: The identification of the data point.
- **Lang**: The language used.
- **Model Input**: The prompt given to the model.
- **Model Output Text**: The model-generated output, which might contain hallucinations.
- **Model ID**: The identification for the model.
- **Model Output Tokens**: The tokenized model output text.
- **Model Output Logits**: The raw, unnormalized model output text.
- **Soft Labels**: The start and end indices of a hallucination along with a probability score.
- **Hard Labels**: The start and end indices of a hallucination, determined using majority voting among the annotators.

For the English datasets, the data points were annotated by up to 13 annotators. Each annotator was provided with the model output text and relevant context. Then, they were instructed to highlight each span of model output text that was inconsistent with the given context. Annotators were instructed to be as conservative as possible when marking hallucinations.

Dataset	% soft labels	% hard labels
Validation	77.6%	28.9%
Test	78.4%	34.9%

Table 1: The % of model output text that was marked as a hallucination.

Table 1 shows the percentages of model output text that was marked as a hallucination by the annotators for both labels. This shows that the soft labels show a relatively less conservative level of annotation than the hard labels, which only span roughly a third of the output text instead of three-quarters for the soft labels.

The authors additionally released an unlabeled training data set, comprising 809 data points for English. However, we did not use this training set for the development of our model as it did not fit within our chosen approach.

4 Method

As mentioned previously, our approach leverages a form of RAG, an LLM, and two prediction models. For efficiency, we split our pipeline into two segments: 1. Knowledge retrieval, and 2. Hallucination detection. This pipeline is illustrated in Figure 1.

4.1 Knowledge Retrieval

To retrieve relevant contextual information, we first extracted all named entities present in the model input, or prompt, provided by the dataset. For this, we used spaCy's NER-tagger (Honnibal and Montani, 2017). After collecting these named entities, we filter out the unwanted labels (e.g., monetary entities). The remaining named entities are then used to fetch any Wikipedia page using the Wikipedia API that possibly contains relevant information, which was saved with its respective data point. As a result, the number of retrieved pages per data point varied depending on the length of the prompt as well as the overall popularity of the categories found in the prompt. This varied from 1 or 2 pages to dozens of pages.

After retrieval, we computed the similarity between each sentence on each retrieved page and calculated the average similarity score of each page. We used the MPNet model [2] to calculate these similarity scores, where a high score indicates that the given sentence has a high chance of containing relevant information. By setting a similarity threshold, we narrowed down the amount of contextual information by only utilizing the pages (max = 3) with the highest similarity score. These pages were preprocessed to remove notes, links, and references and were saved to be summarized.

4.2 Summarization

Another crucial part of the knowledge retrieval pipeline is the summarization model. As there were still instances where the retrieved contextual information was too elaborate, even after setting a threshold, we implemented a summarization model. This model transformed the contextual information into a more concise and informative version, creating summarizations between 20 and 1291 words. We used `DistilBART-CNN-12-6`[3] to carry out this summarization task, which is a transformer-based model fine-tuned for abstractive summarization. This model was chosen because it is relatively fast and computationally light, minimizing the total computational load of our pipeline.

As the model has a maximum token limit of 1024 tokens, information of a longer length needed to be split into segments of text that were small enough to fit within the model's token limit while still preserving meaningful context. After tokenization, each chunk was provided to the model, which was then transformed more concisely while maintaining key information. Finally, repeated phrases were filtered out to ensure that redundant or overlapping content was minimized.

4.3 Hallucination Detection

Prompting For prompting, we used the Llama-3 (8B) Instruct model along with its tokenizer (AI@Meta, 2024). We experimented with several prompting techniques, including zero-shot, few-shot, and Chain-of-Thought (CoT). As our model had difficulty taking on examples or instructions to reason step-wise, the best-performing method was the zero-shot technique. Additionally, we tested different ways of instructing the model to extract hallucinations: either as character spans or as lists of words. We found that requesting words directly was more effective. The final prompt was as follows:

```
Text: {output_text}

Factual Information: {wiki_summary}

Compare the text with the factual
information. What spans in the text
are not consistent with the factual
information provided?

Provide a list of words or spans in
the exact format:

["word1", "word2", ...]

Do not return anything other than the
list of spans.

If there are no hallucinations,
return [].
```

The hallucination spans were extracted from the model's response using regular expressions.

Token Probability Analysis To detect low-confidence tokens, we computed token probabilities from the Llama-3 model as our next step. Specifically, we calculated the log probabilities for each token and converted them into probabilities

[2] https://huggingface.co/sentence-transformers/multi-qa-mpnet-base-cos-v1

[3] https://huggingface.co/sshleifer/distilbart-cnn-12-6

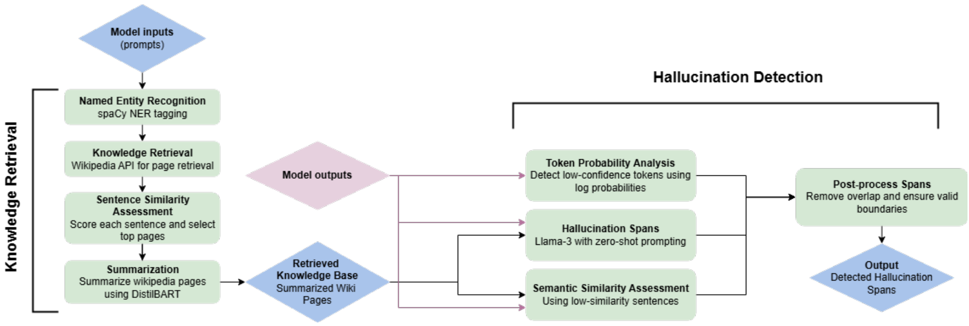
Model inputs
(prompts)
Knowledge Retrieval
Named Entity Recognition
spaCy NER tagging
Knowledge Retrieval
Wikipedia API for page retrieval
Sentence Similarity Assessment
Score each sentence and select top pages
Summarization
Summarize wikipedia pages using DistilBART
Model outputs
Retrieved Knowledge Base
Summarized Wiki Pages
Hallucination Detection
Token Probability Analysis
Detect low-confidence tokens using log probabilities
Hallucination Spans
Llama-3 with zero-shot prompting
Semantic Similarity Assessment
Using low-similarity sentences
Post-process Spans
Remove overlap and ensure valid boundaries
Output
Detected Hallucination Spans

Figure 1: Overview of our two-stage hallucination detection pipeline.

using the softmax function. We analyzed all tokens in the generated text but paid particular attention to content words (e.g., named entities, numbers, and nouns) since these were more likely to contain factual claims. We identified low-probability tokens by setting a threshold of 0.01. Tokens with probabilities below this threshold were considered uncertain and were saved as potential hallucinated character spans. The underlying intuition was that the model assigns lower probabilities to tokens when it is uncertain about their correctness, which often correlated with hallucinated content.

Semantic Similarity-Based Detection In addition to probability analysis, we performed semantic similarity assessment using a transformer-based sentence embedding model, SentenceTransformers' multi-qa-mpnet-base-cos-v1 (Reimers and Gurevych, 2019). All sentences in the model output were compared against the retrieved factual knowledge using this model to retrieve a cosine similarity score. A cosine similarity threshold of 0.5 was used, where a maximum similarity score below this threshold was flagged as a potential hallucination.

Assembling Spans Since multiple methods generated hallucination spans, we merged overlapping spans and removed spans exceeding text boundaries. These text boundaries refer to the length of the model output text, as in some cases, identified spans went beyond this length.

4.4 Evaluation Metrics

To evaluate the performance of our method for hallucination detection, we used the official shared task metrics: Intersection over Union (IoU) and Spearman Correlation. The IoU score measures the overlap between detected and ground-truth hallucination spans:

$$\text{IoU} = \frac{|\text{Predicted Spans} \cap \text{Ground Truth Spans}|}{|\text{Predicted Spans} \cup \text{Ground Truth Spans}|}$$

IoU is 1.0 if neither the reference nor the prediction contains hallucinations. Otherwise, it calculates the ratio of overlapping character indices between predicted and gold-standard hallucination spans.

The Spearman Correlation was used to evaluate the ranking similarity between predicted and reference soft labels. If either of them contains no variation, the score is binary. Otherwise, it computes the Spearman rank correlation between the two probability distributions over characters.

Any results are also compared to the baselines provided by the Shared Task authors, which consisted of a baseline that marked all characters as hallucinations, a baseline that marked no characters as a hallucinations, and a simple neural model based on XLM-RoBERTa[4].

5 Results

In Table 2, the results of our method on the test data are displayed, compared to the baseline scores and the scores obtained by the best-performing team. Additionally, in Table 3, we present our model's performance on the validation set. Our system scores below the 'mark all' baseline on the IoU metric, indicating that we fail to capture all hallucination spans. However, the correlation results suggest that our system tends to over-identify spans overall. In particular, we observed that the next-token-based approach frequently flagged multiple short spans, but our chosen LLama3 model does this as well for some sentences. This inflated our false positive count and thus lowered our correlation score. We think this is partly due to not

[4] https://huggingface.co/FacebookAI/xlm-roberta-base

implementing additional logic to merge or filter overlapping segments. Despite these shortcomings, our approach still surpasses the 'mark none' and 'neural' baselines on IoU and scored 33rd overall.

Model	IoU	Correlation
Best Team (iai_MSU)	0.6509	0.6294
Baseline (mark all)	0.3489	0.0000
HalluRAG-RUG	0.3093	0.0833
Baseline (mark none)	0.0325	0.0000
Baseline (neural)	0.0310	0.1190

Table 2: Our test set results, compared to the provided baselines and the best-performing team.

Precision	Recall	F1-score	IoU
0.30	0.96	0.40	0.30

Table 3: Performance of our model on the validation set.

To improve upon some shortcomings, we experimented with removing the summarization module from our pipeline, as well as removing the next token prediction. All of these experiments lowered the IoU considerably. Adjusting the prompt seemed to have the biggest impact on the IoU. We also experimented with retrieving words instead of character spans, and then reconstructing indices afterward, however this consistently produced worse scores overall.

To better understand our model's behavior, we analyze three representative cases that contain errors.

Input: *"What did Petra van Staveren win a gold medal for?"*
Model Output: *"Petra van **Stoveren** won a **silver** medal at the **2008 Summer Olympics in Beijing, China**."*
Gold Spans: *[(25, 31), (45, 49), (69, 83)]*
Detected Spans: *[(12, 13), (15, 24), (25, 34), (35, 44), (45, 80)]*
Analysis: The model correctly flagged key errors (misspelled name, incorrect medal, fabricated location) but over-predicted by marking additional parts of the sentence as hallucinated. This suggests a need for better span-merging techniques.

Input: *"How many genera are in the order Erysiphales?"*
Model Output: *"The **Elysiphale** order contains **5** genera."*
Gold Spans: *[(30, 31)]*
Detected Spans: *[(25, 35)]*
Analysis: The model mistakenly detected a larger hallucination span than needed. It correctly identified the incorrect taxonomic name but over-marked the number as part of a broader span. A more refined approach could improve precision by more accurately isolating factual errors.

Input: *"When did Chance the Rapper debut?"*
Model Output: *"Chance the Rapper debuted in **2011**."*
Gold Spans: *[(29, 33)]*
Detected Spans: *[(0, 34)]*
Analysis: Instead of isolating the incorrect year, our system flagged the entire sentence as a hallucination. This exemplifies a common issue where token probability-based hallucination detection overextends spans.

These cases highlight key failure points in our approach: entity misrepresentation, overgeneration, and excessive span marking.

6 Conclusion and Future Work

In this work, we presented a retrieval-augmented pipeline for detecting hallucinated spans in LLM output, focusing on English data from the Mu-SHROOM task. Our system combined token probability, factual checks, and summarized Wikipedia context to highlight hallucinated spans. While our approach outperformed two out of three baselines, it often detected an overabundance of spans, reducing precision and diluting overall performance. In addition, the practical constraints of our chosen model regarding input length and model parameters restricted performance. Despite these challenges, our results suggest that integrating retrieval methods and careful prompt engineering can help with validating LLM output.

Future Work

Future work could include refining the method for merging overlapping hallucination spans, potentially creating a higher threshold for span inclusion. Furthermore, exploring LoRA-style downscaling or newer open-source models like DeepSeek might help improve the performance of a RAG-based approach.

Acknowledgments

We would like to thank Malvina Nissim and Huiyuan Lai for their guidance and advice during this project, as well as the Mu-SHROOM organizers for their helpfulness in clarifying any doubts and questions.

References

AI@Meta. 2024. Meta-llama/Meta-Llama-3-8B-Instruct · Hugging Face. https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct.

Orlando Ayala and Patrice Bechard. 2024. Reducing hallucination in structured outputs via retrieval-augmented generation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*, pages 228–238, Mexico City, Mexico. Association for Computational Linguistics.

Federico Borra, Claudio Savelli, Giacomo Rosso, Alkis Koudounas, and Flavio Giobergia. 2024. Malto at semeval-2024 task 6: Leveraging synthetic data for llm hallucination detection. *arXiv preprint arXiv:2403.00964*.

Thomas Dopierre, Christophe Gravier, and Wilfried Logerais. 2021. PROTAUGMENT: Unsupervised diverse short-texts paraphrasing for intent detection meta-learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2454–2466, Online. Association for Computational Linguistics.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey. *Preprint*, arXiv:2312.10997.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Thanet Markchom, Subin Jung, and Huizhi Liang. 2024. Nu-ru at semeval-2024 task 6: Hallucination and related observable overgeneration mistake detection using hypothesis-target similarity and selfcheckgpt. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 253–260.

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, Mexico City, Mexico. Association for Computational Linguistics.

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)*, pages 1979–1993, Mexico City, Mexico. Association for Computational Linguistics.

Timothy Obiso, Jingxuan Tu, and James Pustejovsky. 2024. HaRMoNEE at SemEval-2024 task 6: Tuning-based approaches to hallucination recognition. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1322–1331, Mexico City, Mexico. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

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

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.