AggTruth: Contextual Hallucination Detection using Aggregated Attention Scores in LLMs *

Piotr Matys[0009−0004−9282−2892], Jan Eliasz[0009−0007−0851−1816]**, Konrad Kiełczyński[0009−0009−3223−2336], Mikołaj Langner[0009−0007−9531−5329], Teddy Ferdinan[0000−0003−3701−3502)], Jan Kocoń[0000−0002−7665−6896)], and Przemysław Kazienko[0000−0001−5868−356X]

Department of Artificial Intelligence, Wroclaw Tech, Wrocław, Poland
{jan.eliasz, jan.kocon, kazienko}@pwr.edu.pl

Abstract. In real-world applications, Large Language Models (LLMs) often hallucinate, even in Retrieval-Augmented Generation (RAG) settings, which poses a significant challenge to their deployment. In this paper, we introduce AggTruth, a method for online detection of contextual hallucinations by analyzing the distribution of internal attention scores in the provided context (passage). Specifically, we propose four different variants of the method, each varying in the aggregation technique used to calculate attention scores. Across all LLMs examined, AggTruth demonstrated stable performance in both same-task and cross-task setups, outperforming the current SOTA in multiple scenarios. Furthermore, we conducted an in-depth analysis of feature selection techniques and examined how the number of selected attention heads impacts detection performance, demonstrating that careful selection of heads is essential to achieve optimal results.

Keywords: LLM · NLP · Hallucination detection · Retrieval-Augmented Generation (RAG) · Attention map · Contextual hallucination

1 Introduction

Large language models (LLMs) have gained popularity over the past few years due to their remarkable capabilities on a wide range of natural language processing tasks [10, 16, 19]. However, they often suffer from one major drawback, a tendency to hallucinate. Although definitions of hallucination may vary across sources, the one we adopt defines it as the phenomenon of producing responses that are non-sensical or unfaithful to the provided source content [8]. This phenomenon is a critical barrier to deploying LLMs in real-world applications where accuracy and reliability are crucial.

* Financed by: (1) the National Science Centre, Poland (2021/41/B/ST6/04471); (2) CLARIN ERIC (2024–2026), funded by the Polish Minister of Science (agreement no. 2024/WK/01); (3) CLARIN-PL, the European Regional Development Fund, FENG programme (FENG.02.04-IP.040004/24); (4) statutory funds of the Department of Artificial Intelligence, Wroclaw Tech; (5) the Polish Ministry of Education and Science ("International Projects Co-Funded" programme); (6) the European Union, Horizon Europe (grant no. 101086321, OMINO); (7) the EU project "DARIAH-PL", under investment A2.4.1 of the National Recovery and Resilience Plan. The views expressed are those of the authors and do not necessarily reflect those of the EU or the European Research Executive Agency.

** Corresponding author: jan.eliasz@pwr.edu.pl

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_18

In our work, we specifically focus on detecting contextual hallucinations in the Retrieval-Augmented Generation (RAG) setup [13]. These hallucinations can be classified as intrinsic hallucinations, which means that the model output directly conflicts with the provided context. Although RAG effectively enhances the factuality of LLMs by enriching their inputs with additional contextual information, it has its own limitations. The model may still produce hallucinations when faced with a noisy or inconsistent context or because of the inappropriate use of the accurate context [7]. Effective detection of hallucinations is also a crucial step in the self-learning LLM framework [4], as it provides information on model inaccuracies. In our study, we analyzed different hallucination detection techniques and proposed some strategies. Our key contributions are as follows:

- Proposing four attention map aggregation techniques allowing classifier training for windowed, real-time detection of the contextual hallucinations
- Conducting complex performance comparison of feature selection methods across many models and datasets
- Analyzing the influence of the number of heads on hallucination detection
- Pointing out possible improvements in the current SOTA based on the attention map method for contextual hallucination detection

2 Related work

Research on hallucination detection in LLMs has evolved along several paths, with some focusing more on external response comparison methods while others examine the model's internal states during generation.

One of the simplest external methods involves assessing the factuality of a model in a zero-resource fashion by comparing inconsistencies between multiple responses [14]. A similar but more statistically grounded idea, based on semantic entropy, has also been presented [3]. Both approaches are limited by the requirement to generate multiple answers, compromising online efficiency.

Fine-tuning has also been proposed as a strategy to mitigate hallucinations [5]. However, while this approach is effective for specific tasks, it may lead to catastrophic forgetting [9] of tasks that are not included in the fine-tuning process.

The internal state analysis approach offers an alternative perspective. For example, such analyses can include hidden states [1]. However, this article only considers a true-false dataset. Building on this foundation, [2] showed that the attention map can also be an alternative. This approach has demonstrated better performance in transferring between different tasks while having a smaller number of input features.

3 Contextual hallucination detection method based on attention map

As [2] has shown, attention map analysis provides valuable insight for LLMs hallucination detection during context-based retrieval-augmented generation (RAG)

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_18

tasks. We hypothesize that examining the complete attention matrix may not be necessary. Instead, focusing specifically on the attention patterns related to the provided context (passage) could be sufficient. Such an approach should provide features that enable proper hallucination detection. We propose creating attention-based features by analyzing how newly generated tokens pay attention to the provided passage within a defined window as illustrated in Figure 1.

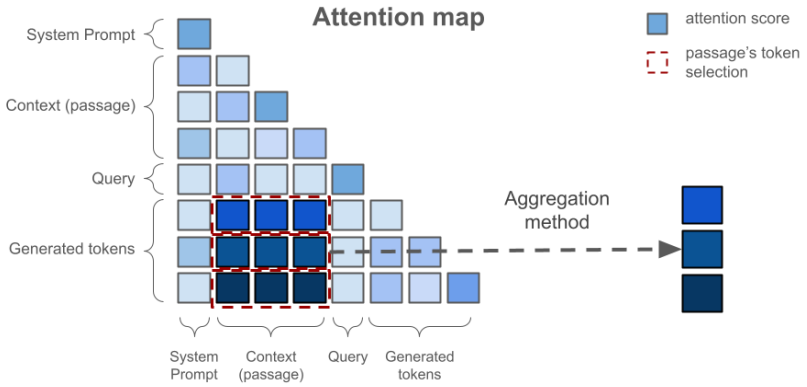

Fig. 1. Selection of tokens from the LLM input and aggregation of attention scores. The darker subregion indicates the attention scores of generated tokens on the provided passage. The aggregation of each such region provides one feature for the hallucination detection. Here, we have three regions resulting in three features.

3.1 Attention maps aggregation techniques

For every response generated by a transformer, a four-dimensional matrix of shape $L \times H \times N \times C$ (Layers $\times$ Heads $\times$ Generated tokens $\times$ Context tokens) is created. To train a hallucination detection model, attention scores over the provided passage for each generated token have to be aggregated to reduce the map dimensionality and volume. For each token generated, its attention distribution across the passage is aggregated.

Let $a_{l,h,t,i}$ be an attention score corresponding to layer $l \in \{1, \ldots, L\}$, head $h \in \{1, \ldots, H\}$, generated token $t \in \{1, \ldots, N\}$, context token $i \in \{1, \ldots, C\}$, and $\mathbf{a}_{l,h,t}$ be a vector of such scores for the whole context. The proposed aggregation techniques are as follows:

AggTruth Sum The simplest approach to aggregate attention scores is to sum up the attention scores that were distributed over passage tokens during the generation of a given token (Equation 1).

$$\text{Sum}_{l,h,t} = \sum_{i=1}^{C} a_{l,h,t,i} \tag{1}$$

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_18

It is based on the assumption that LLMs produce more factual responses when they pay more attention to the provided context tokens. Due to the use of only tokens from the provided context, the sum of the attention scores does not equal 1, which is a desired characteristic, as it means that the examples differ from each other.

AggTruth CosSim In this approach, each head in a layer is described by the attention scores it assigns to the passage tokens while generating a new token. Then, the cosine similarity between the heads within the layer is calculated and the values are averaged for each head (Equation 2).

$$\text{CosSim}_{l,h,t} = \frac{1}{H-1} \sum_{\substack{h'=1 \\ h' \neq h}}^{H} \frac{\mathbf{a}_{l,h,t} \cdot \mathbf{a}_{l,h',t}}{\|\mathbf{a}_{l,h,t}\| \|\mathbf{a}_{l,h',t}\|} \tag{2}$$

This produces a single value for each head, representing its similarity to other heads in the layer when generating a given token. The intuition behind this approach is that when unreliable content is produced, all the heads become similar to each other, indicating that the model lacks a clear focal point.

From the other perspective, apart from asking how much, we want to ask in which way the LLM's attention spreads across context tokens. To answer that, in our study, we treat attention scores as pseudo-probabilities. That allows us to examine the quantities related to the probability distributions. To fulfill the summing-up-to-one criteria, we do not simply normalize the attention scores. It is undesirable as it loses information about the original magnitude and distribution of attention scores. Instead, for the remaining aggregation techniques, we append an additional value to the end of the attention score vector calculated as the difference between 1 and the calculated sum of attention scores.

AggTruth Entropy The concept of uncertainty leads to another aspect of the possible cause of hallucinations produced by LLMs. The approach described by Equation 3 measures whether a model during generation is focused only on some particular information or on the context as a whole.

$$\text{Entropy}_{l,h,t} = -\sum_{i=1}^{C} a_{l,h,t,i} \log_2 a_{l,h,t,i} \tag{3}$$

AggTruth JS-Div We consider hallucinations as outliers, representing a lack of knowledge required to perform the task. The main motivation behind this work was the assumption that hallucinated and non-hallucinated examples could be differentiated based on the stability of the distributions across layers, allowing

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_18

for the identification of outliers. Since attention scores are treated as probability distributions, the average distribution of the heads in each layer can be calculated. The Jensen-Shannon distance is then used to measure the dissimilarity between each head and the average distribution within each layer separately (Equation 4).

$$\text{JS-Div}_{l,h,t} = \sqrt{\frac{1}{2}\sum_{i=1}^{C}\left(a_{l,h,t,i}\ln\frac{a_{l,h,t,i}}{m_{l,h,t,i}} + a_{l,\text{ref},t,i}\ln\frac{a_{l,\text{ref},t,i}}{m_{l,h,t,i}}\right)} \tag{4}$$

where:

$$a_{l,\text{ref},t,i} = \frac{1}{H}\sum_{h=1}^{H} a_{l,h,t,i}, \qquad m_{l,h,t,i} = \frac{a_{l,h,t,i} + a_{l,\text{ref},t,i}}{2}$$

3.2 Feature (heads) selectors

Central tendency measure ratio ($\textbf{Center}_\textbf{r}$) Any central tendency measure that can be calculated for hallucinated and not hallucinated examples can be used. The idea is to select the $\mathbf{r}$ fraction of heads with the highest (lowest) ratio between the two classes. In this study, we used the median, one of the most popular statistics for comparing groups. We keep the top $\mathbf{r/2}$ fraction of heads with the highest ratio and $\mathbf{r/2}$ fraction of heads with the lowest ratio. This technique is proposed as a naive baseline for selecting heads.

Above random feature performance ($\textbf{Random}_\textbf{n,k}$) A feature with random values is appended to the dataset. Only heads with a higher absolute importance value are accepted. As we use logistic regression, we choose heads based on the values of their coefficients. The procedure is repeated $\mathbf{n}$ times and only the heads accepted at least $\mathbf{k}$ times are selected. This is a simplified version of the Boruta feature selection algorithm [11], which omits feature selection based on the binomial distribution and employs logistic regression instead of random forest. Using the original algorithm took much longer as more iterations were necessary, but the results were comparable.

Above random feature performance - positive coefficients ($\textbf{Random}^{+}_\textbf{n,k}$) The idea is the same as above, but only positive coefficients are considered.

Lasso based selection (Lasso) L1 regularization used by the Lasso technique shrinks some coefficients to zero, resulting in a sparse model. We leverage this property to select only the features with non-zero coefficients.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_18

Spearman correlation with the target (Spearman$_\mathbf{r}$) During the exploratory data analysis, a high degree of multicollinearity was observed among the heads. As a result, selecting them based on the correlation among themselves or the variance inflation factor is ineffective. Thus, the selection process focuses on the correlation with the target variable and selects the **r** fraction of heads that exhibit the highest statistically significant Spearman correlation factor, with a strict threshold set at $p = 0.001$. Additionally, a setup is introduced for $r =$ auto, which means that only features with a correlation greater than half of the highest observed value are considered.

3.3 Aligning attention scores and hidden states to proper token

The attention map returned along with a newly generated token contains information about the impact of previous tokens on its generation. However, the token under consideration does not influence the attention scores. That is, regardless of which token is selected in the generation step (which may depend on whether greedy decoding or another sampling method is used), the attention map remains the same. A token starts to influence the attention map only after it has been generated. We hypothesize that to create features that describe token **t**, the attention map extracted from the process of generating token **t**+**1** should be used. If this is not taken into account, each token is described by the features of its predecessor.

3.4 Passage percentage feature

Generating new tokens influences the number of tokens to which LLMs must pay attention. For the same input passage, the overall percentage of attention given to the passage decreases as new tokens are generated because the attention becomes more dispersed across a greater number of tokens. It may have an undesirable impact on certain attention map aggregation techniques, such as AggTruth Sum, as the aggregation results naturally decrease over time.

To address this issue, we propose incorporating an additional feature into the dataset that represents the percentage ratio of the passage length to the total input length when each token is generated.

4 Experimental Setup

To produce a reliable dataset for further evaluation, we have composed a complete pipeline (Figure 2) that for each prompt leads to features that are eventually used to train hallucination detection classifier. The pipeline retrieves LLM answers with corresponding attention maps and then submits these answers to GPT-4o for evaluation. The process continues by selecting the attention scores put on the passage tokens and aligning each generated token with a binary label (1 for hallucinated, 0 for non-hallucinated). Attention scores are then aggregated using the chosen AggTruth aggregation technique and all generated tokens are

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_18

divided into segments through windowing, i.e., they are processed into overlapping chunks of size 8, suggested by best baseline method [2], sliding token by token. If a chunk contains any hallucinated token, then it is aligned with label 1 and 0 otherwise. In addition, for every chunk, the features are averaged over the tokens, resulting in **a single feature representing a single head**. After processing all the examples, the results are concatenated into a unified dataset. This data then passes through a selector, producing the final feature set that enables classifiers to identify potential hallucinations within LLM responses.

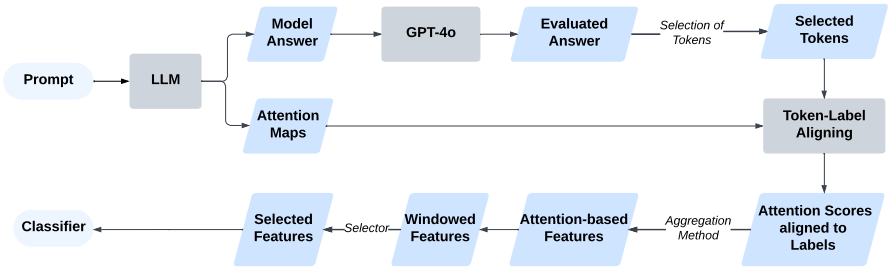

Fig. 2. The whole end-to-end pipeline. It begins with a *Prompt* passed to an *LLM* and eventually results in *Selected Features* based on which the final *Classifier* detects potential hallucinations in the obtained answer.

4.1 Data

We first established a comprehensive dataset of language model responses with token-level annotations to train classifiers for hallucination detection. We restricted our analysis to 4,096 LLM's context length, considering that some of the models we evaluate employ the Sliding Window Attention Mechanism [6], which prevents the extraction of attention scores over the whole passage once the sliding window size is exceeded.

We focused our evaluation on two fundamental natural language processing tasks: Question Answering (QA) and Summarization. For the QA assessment, we used 1,535 examples from Natural Questions (NQ) [12] and 896 from HotPotQA [18]. The summarization evaluation incorporated 1000 examples from CNN/Daily Mail (CNN/DM) [17] and 1000 from XSum [15].

The responses were generated using greedy decoding across the examined LLMs for all dataset examples. This process naturally produced outputs that contained both factual and hallucinated content. To establish ground truth labels at the token level, we employed GPT-4o as our judging model. We show the results of this evaluation for all models in Table 1. This choice was supported by recent research [2] showing that GPT-4o's judgments align with human assessments in 97% of the cases. We also conducted a study where a human expert annotated 75 samples, achieving a Cohen's Kappa of 0.7 between the expert and GPT-4o.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_18

LLM	Layers	Heads	Hidden Size	NQ	HotPotQA	CNN/DM	XSum
meta-llama/Llama-2-7b-chat-hf	32	32	4096	54.9	62.0	74.4	63.2
meta-llama/Llama-3.1-8B-Instruct	32	32	4096	75.8	56.1	86.8	77.0
unsloth/gemma-2-9b-it-bnb-4bit	42	16	3584	84.4	85.3	92.6	75.5
microsoft/Phi-3.5-mini-instruct	32	32	3072	61.0	74.2	75.7	66.6

Table 1. Configuration with key hyperparameters and truthfulness evaluation results of examined LLMs.

4.2 Evaluation Configurations

Our framework evaluates methods in same-task and cross-task settings. The dataset is divided into training and test sets, with the training set undergoing 5-fold cross-validation to assess model's stability and validation performance. The test set is further subdivided into subsets for the same-task and cross-task to evaluate the model's generalization. As our classifier, a Logistic Regression model with class-weight balancing is employed. This model was chosen because of its speed , interpretability and was the default model in the baselines. Time-series-specific models such as LSTM were rejected in the early testing phase due to their poor performance on the hallucination detection task. Features before being passed to the classifier are normalized using the min-max strategy.

The evaluation structure follows two primary configurations, as shown in Table 2. This dual-configuration approach enables us to assess the model's ability to generalize across different question-answering and summarization tasks.

Source	Target	Source		Target	
		Train/Val	Test	Test(1)	Test(2)
QA	SUM.	NQ	HotPotQA	XSum	CNN/DM
SUM.	QA	CNN/DM	XSum	HotPotQA	NQ

Table 2. Description of datasets used at each stage of evaluation for each task setup.

4.3 Baselines

Lookback-Lens This method utilizes features constructed as the ratio of attention placed on the prompt versus newly generated tokens [2]. In contrast to our approach, the attention placed on the entire provided prompt is analyzed (including the system prompt and query), which naturally may lack robustness for changing input. This method generalizes well and achieves competitive results in a cross-task setting, being the current SOTA for windowed, online hallucination detection.

Hidden states-based classifier Based on the literature [1] hidden states can serve as effective features for hallucination detection. Their findings indicate that the optimal results are achieved by utilizing hidden states from the upper layers of the model architecture. Following that observation, we obtain the features of the layers $L, L-4, L-8$, where L represents the total number of

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_18

layers in the language model. The hidden states from these layers are processed by averaging the values across tokens within a defined window, which are then used directly as input features for the classification models.

5 Results

Method	Source	Target	Source			Target		Gap [%]
			Train	Val	Test	Test(1)	Test(2)	
Text based NLI								
SOTA NLI*	QA	SUM.	—	—	—	—	0.530	—
	SUM.	QA	—	—	—	—	0.649	—
Hidden States based								
24th Layer	QA	SUM.	0.954	0.945	0.717	0.625	0.629	8.752
	SUM.	QA	0.988	0.982	0.700	0.707	0.628	5.356
28th Layer	QA	SUM.	0.955	0.946	0.727	0.603	0.623	9.564
	SUM.	QA	0.988	0.981	0.691	0.704	0.611	6.730
32th Layer	QA	SUM.	0.950	0.939	0.739	0.605	0.621	9.038
	SUM.	QA	0.986	0.978	0.678	0.660	0.573	11.117
Attention based								
Lookback Lens (paper)**	QA	SUM.	—	—	—	—	0.661	—
	SUM.	QA	—	—	—	—	0.660	—
Lookback Lens (classifiers)***	QA	SUM.	—	—	0.554	0.666	0.635	13.688
	SUM.	QA	—	—	**0.722**	0.506	0.506	19.299
Lookback Lens (retrained)****	QA	SUM.	0.839	0.833	**0.752**	0.643	0.681	3.952
	SUM.	QA	0.898	0.882	0.700	0.523	0.521	18.820
AggTruth Sum	QA	SUM.	0.802	0.799	0.723	**0.670**	**0.710**	**2.612**
	SUM.	QA	0.894	0.885	0.706	**0.724**	**0.660**	**2.714**

Table 3. Comparison of Llama-2 hallucination detection efficiency (AUROC) between AggTruth Sum, SOTA NLI based method, hidden states based classifier and Lookback-Lens.

*SOTA NLI: Results cited from [2].
**Lookback Lens (paper): Results cited from [2].
***Lookback Lens (classifiers): Results obtained by trained classifiers shared by the authors of [2].
****Lookback Lens (retrained): Results obtained by classifiers retrained with the script shared by the authors of [2], but using the same prompt as in AggTruth experimental setup.

For each model and for each source–target pair a **Gap** value (Equation 5) is presented apart from AUROC.

$$\mathrm{Gap}_m = \frac{1}{|S|} \sum_{s \in S} \frac{\max_{m' \in M} \mathrm{AUC}^s_{m'} - \mathrm{AUC}^s_m}{\max_{m' \in M} \mathrm{AUC}^s_{m'}} \cdot 100\% \tag{5}$$

where $m \in M$ is a specific method for which Gap is being calculated among all of M methods that we take into account, S is a set of all three test sets and AUC^s_m is an AUC score obtained by method m on a test set s. The maximum observed AUC scores across all models and test sets are presented in Table 7. To examine the stability of the methods, the Gap is averaged over the test sets. It is necessary as simply averaging AUCROC values is not informative enough,

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_18

because different datasets contain varying numbers of positive samples and have different levels of difficulty, leading to different ranges of possible metric values.

Table 3 shows the efficiency of hallucination detection for Llama-2. Unlike hidden state-based approaches, attention-based methods are less prone to overfitting to the training dataset, resulting in a higher AUCROC on out-of-domain target datasets. However, for Lookback-Lens, a substantial decrease in performance can be observed for SUM. $\rightarrow$ QA setting for both the provided by authors and retrained classifiers compared to the results of [2]. AggTruth Sum achieves the highest AUCROC across all target tasks, while also getting stable, high AUCROC on the source test setting, resulting in the lowest Gap values. This observation suggests that AggTruth can be more robust to the choice of the training data.

Table 4 presents the results for other models. It lacks the Lookback-Lens method, as the code provided by its authors is not easily adaptable for models other than Llama-2. Considering the gap values, AggTruth JS-DIV performed best for Phi-3.5 and Llama-3, while AggTruth Sum led on Gemma-2. Overall, AggTruth achieves the highest AUCROC and the lowest gap values. It should be noted that, even though the hidden state-based approach performed best in QA $\rightarrow$ SUM. task for Gemma-2, on SUM. $\rightarrow$ QA it performs worse than a random classifier, which is proof of the instability of this approach.

The best results for each AggTruth aggregation method grouped by task are presented in appendix Table 6.

6 Predictive Power of Different Heads

In [2], the authors showed that increasing the number of heads passed to the classifier increases the performance, and taking all of them produces the best results. However, they focused on a very small subset of heads (10, 50, and 100) for Llama-2 which is approximately 1%, 5%, and 10% of all its heads. Additionally, they checked it only on responses divided into predefined spans (in contrast to windows) that are not available during the decoding. One of the most important questions we wanted to answer was whether we would observe a similar percentage of heads needed to achieve the best performance for each of the LLMs and what the optimal number of them is. In Table 5 we show the results for the detection tasks, comparing the use of all heads versus a percentage of them across different LLMs. Here, feature selection can differ between tasks, as it is a training dataset-specific setting and as different training data can contain various amounts of predictive information. We state that not taking all heads can not only reduce the training time and complexity of the model but also improve the performance. The best performance was observed when taking at least around half of the heads for all LLMs, but it is possible to use even $10 - 20\%$ of them for some tasks without a significant decrease. In some experiments, that low number of heads was sufficient to obtain higher AUCROC than when using all of them. As we did not perform hypertuning and only ran

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_18

LLM	Method	Source	Target	Source			Target		Gap [%]
				Train	Val	Test	Test(1)	Test(2)	
llama-3.1-8B-Instruct	Hidden States (24th Layer)	QA	SUM.	0.932	0.925	0.877	0.606	0.623	8.957
		SUM.	QA	0.996	0.991	0.680	**0.860**	0.768	5.049
	Hidden States (28th Layer)	QA	SUM.	0.932	0.925	0.885	0.607	0.619	8.745
		SUM.	QA	0.997	0.993	0.665	0.851	0.761	6.439
	Hidden States (32th Layer)	QA	SUM.	0.927	0.921	**0.886**	0.582	0.589	11.314
		SUM.	QA	0.996	0.990	0.633	0.844	0.751	8.573
	AggTruth JS-Div	QA	SUM.	0.854	0.851	0.845	**0.689**	**0.722**	**1.554**
		SUM.	QA	0.893	0.875	**0.715**	0.766	**0.853**	**3.786**
phi-3.5-mini-instruct	Hidden States (24th Layer)	QA	SUM.	0.939	0.927	0.702	0.611	0.634	6.401
		SUM.	QA	0.967	0.953	0.670	0.523	0.648	11.161
	Hidden States (28th Layer)	QA	SUM.	0.939	0.928	0.707	0.601	0.632	6.753
		SUM.	QA	0.968	0.953	0.662	0.559	**0.649**	9.867
	Hidden States (32th Layer)	QA	SUM.	0.933	0.920	**0.720**	0.593	0.620	7.215
		SUM.	QA	0.967	0.952	0.648	0.555	0.614	12.503
	AggTruth JS-Div	QA	SUM.	0.797	0.790	0.712	**0.674**	**0.647**	**2.172**
		SUM.	QA	0.849	0.835	**0.703**	**0.702**	0.636	**2.029**
gemma-2-9b-it-bnb-4bit	Hidden States (34th Layer)	QA	SUM.	0.961	0.952	0.812	0.579	**0.644**	**4.383**
		SUM.	QA	0.999	0.996	0.582	0.604	0.610	14.497
	Hidden States (38th Layer)	QA	SUM.	0.961	0.952	0.828	0.569	0.630	4.946
		SUM.	QA	0.999	0.996	0.586	0.443	0.537	24.891
	Hidden States (42th Layer)	QA	SUM.	0.958	0.948	**0.824**	0.563	0.610	6.426
		SUM.	QA	0.998	0.995	0.582	0.316	0.482	33.292
	AggTruth Sum	QA	SUM.	0.835	0.827	0.742	**0.625**	0.609	6.606
		SUM.	QA	0.745	0.736	**0.617**	**0.749**	**0.625**	**5.901**

Table 4. Results of hallucination detection for other LLMs that compare our approach with hidden states (baseline).

specific configurations, we suppose that for specific datasets even better results can be achieved.

During the study, we were interested in examining the predictive power of the proposed features to differentiate samples and investigate why selecting features with positive coefficients can be so effective. For each examined dataset and LLM, we calculated the mean value of each feature separately for a group of the non-hallucinated and the hallucinated.

For each head, we conducted Welch's t–test (with p–value= 0.01) to check whether a head evinces higher mean feature values for non-hallucinated examples than for hallucinated ones. Figure 3 shows the percentages of heads that meet this condition. We observed that for most used datasets and LLMs both mean Sum and mean Jensen-Shannon distance for the non-hallucinated examples were statistically higher than for the hallucinated ones. For some datasets, almost all means for non-hallucinated examples were higher. Due to a relatively small number of datasets and observations, we leave it for further investigation. Nevertheless, we would like to point out some characteristics that can potentially differentiate these hallucinated samples in an effective way.

When selecting an aggregation method for LLM, careful consideration is needed since different models may perform optimally with different methods.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_18

LLM	AggTruth	Source	Target	Selector	Heads [%]	Test	Test(1)	Test(2)	Gap [%]
llama-2-7b-chat-hf	Sum	QA	SUM.	—	100.0	0.727	0.582	0.697	6.836
				$\text{Random}^{+}_{3,3}$	48.4	0.723	0.670	0.710	**2.048**
				$\text{Center}_{0.2}$	**19.9**	0.713	0.638	0.739	2.798
		SUM.	QA	—	100.0	0.705	0.722	0.660	2.159
				$\text{Spearman}_{1.0}$	96.2	0.706	0.724	0.660	**1.975**
				$\text{Random}^{+}_{3,3}$	**45.6**	0.699	0.692	0.639	4.776
llama-3.1-8B-Instruct	JS-Div	QA	SUM.	—	100.0	0.844	0.687	0.720	1.740
				$\text{Random}_{3,2}$	96.8	0.845	0.689	0.722	**1.554**
				Lasso	**19.2**	0.856	0.685	0.708	1.973
		SUM.	QA	—	100.0	0.714	0.765	0.854	3.847
				$\text{Random}_{3,1}$	86.7	0.715	0.766	0.853	**3.786**
				$\text{Random}^{+}_{3,3}$	**43.2**	0.702	0.770	0.847	4.512
phi-3.5-mini-instruct	JS-Div	QA	SUM.	—	100.0	0.715	0.651	0.633	3.905
				$\text{Spearman}_{1.0}$	76.6	0.712	0.674	0.647	**2.172**
				$\text{Center}_{0.2}$	**19.9**	0.659	0.660	0.663	4.406
		SUM.	QA	—	100.0	0.703	0.702	0.636	**2.029**
				$\text{Random}_{3,1}$	95.2	0.702	0.703	0.635	2.038
				$\text{Spearman}_{\text{auto}}$	**31.3**	0.696	0.713	0.633	2.045
gemma-2-9b-it-bnb-4bit	Sum	QA	SUM.	—	100.0	0.729	0.637	0.590	7.512
				$\text{Center}_{0.5}$	49.9	0.742	0.625	0.609	**6.606**
				$\text{Spearman}_{0.2}$	**19.9**	0.759	0.583	0.634	6.819
		SUM.	QA	—	100.0	0.662	0.697	0.572	8.642
				$\text{Spearman}_{0.5}$	49.9	0.659	0.748	0.593	**5.563**
				$\text{Spearman}_{0.1}$	**10.0**	0.617	0.749	0.625	5.901

Table 5. Influence of the percentage of selected heads on results with different selectors. Comparisons are made for each LLM, aggregation method, and task between no feature selection (marked with hyphens), best-obtained selection method w.r.t. Gap value, and method resulted in the lowest number of heads.

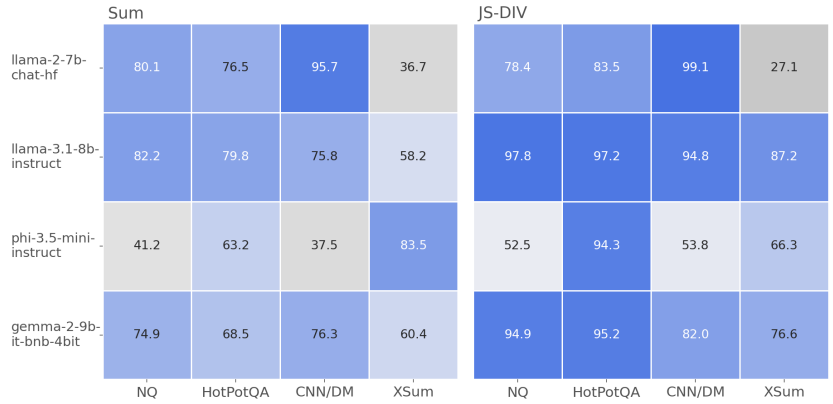

Fig. 3. Percentage of heads for which the mean features (i.e. AggTruth Sum/JS-Div) are statistically higher for non-hallucinated examples than for the hallucinated ones. The comparison was conducted using Welch's t–test with p–value = 0.01

Based on our research findings, we recommend the AggTruth Sum as the preferred default method. It is the most stable and robust approach that resulted in the lowest overall gap across all LLMs examined and tasks, outperforming baselines on the majority of datasets, which is shown in Figure 4. More importantly, calculating the sum of the attention scores is the fastest method across

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_18

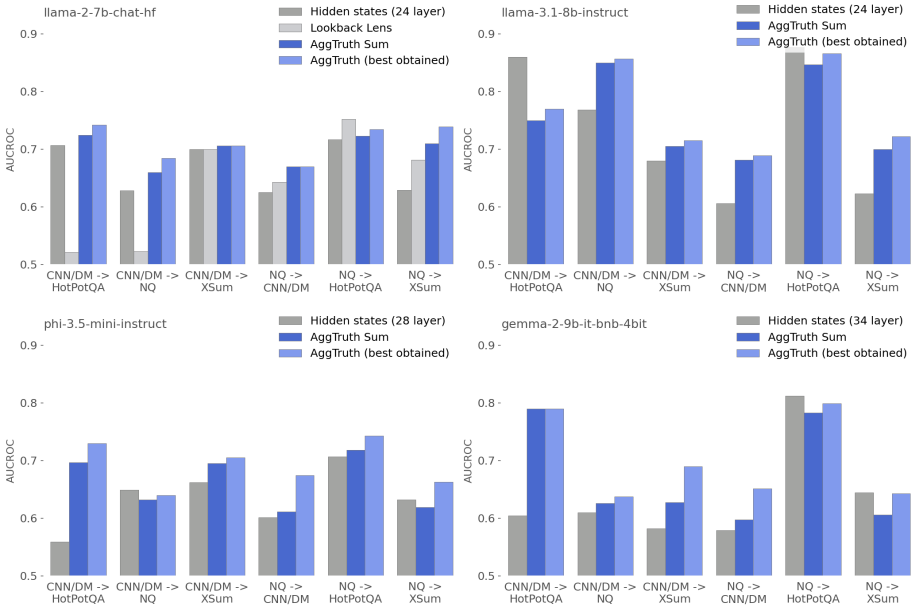

Fig. 4. AUCROC hallucination detection results grouped by LLMs and tasks for best obtained hidden states-based method w.r.t. Gap value, Lookback Lens (only for Llama-2), AggTruth Sum and best obtained AggTruth method for a specific dataset and LLM.

all the proposed aggregations. This performance benefit is particularly crucial in classifier-guided decoding scenarios, where the most truthful generation path is selected [2]. It is also the most intuitive and explainable approach which should be preferred when performance is satisfactory. The feature selection method is more defined by the training set, but our experiments show that the Spearman selector can be used as a first guess. Alternatively, the above random feature selector (potentially taking only positive coefficients) can be applied.

7 Conclusions

We proposed a novel approach to effective online contextual hallucination detection, called AggTruth, with four possible aggregation methods based on attention scores of context tokens. Our method outperforms other baselines, achieving high and stable results across all examined datasets and various LLMs. Moreover, the presented feature selection methods not only reduce the number of heads needed for classification but also improve overall detection performance.

We believe that analyzing specific parts of attention scores vectors can open up new opportunities to detect hallucinations not only in RAG settings but also for other tasks. By design, the AggTruth detector is indicated to be used during the LLM decoding step. AggTruth also has its natural limitations, such as the access to context tokens' attention scores for generated tokens, which are

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_18

not available when using flash-attention. Additionally, some attention patterns, such as sparse or window attention, can limit the use of AggTruth. However, such optimizations, in addition to examining a window of size one and different aggregation methods, are intended for future work along with the development of a novel dedicated classifier-guided decoding method.

References

1. Azaria, A., et al.: The internal state of an LLM knows when it's lying. In: EMNLP 2023. pp. 967–976. ACL, Singapore (Dec 2023)
2. Chuang, Y.S., et al.: Lookback lens: Detecting and mitigating contextual hallucinations in large language models using only attention maps (2024)
3. Farquhar, S., et al.: Detecting hallucinations in large language models using semantic entropy. Nature **630**(8017), 625–630 (Jun 2024)
4. Ferdinan, T., et al.: Into the unknown: Self-learning large language models. In: SENTIRE at ICDM'2024. pp. 423–432. IEEE (2024)
5. Gekhman, Z., et al.: Does fine-tuning llms on new knowledge encourage hallucinations? (2024), https://arxiv.org/abs/2405.05904
6. Gemma, T., et al.: Gemma 2: Improving open language models at a practical size (2024), https://arxiv.org/abs/2408.00118
7. Huang, L., et al.: A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. ACM Trans. on Inform. Sys. (2024)
8. Ji, Z., et al.: Survey of hallucination in natural language generation. ACM Computing Surveys **55**(12), 1–38 (Mar 2023)
9. Ke, Z., et al.: Continual training of language models for few-shot learning (2022), https://arxiv.org/abs/2210.05549
10. Kocoń, J., et al.: Chatgpt: Jack of all trades, master of none. Information Fusion **99**, 101861 (2023)
11. Kursa, M.B., Rudnicki, W.R.: Feature selection with the boruta package. Journal of statistical software **36**, 1–13 (2010)
12. Kwiatkowski, T., et al.: Natural questions: A benchmark for question answering research. Transactions of the ACL **7**, 452–466 (2019)
13. Lewis, P., et al.: Retrieval-augmented generation for knowledge-intensive nlp tasks (2021), https://arxiv.org/abs/2005.11401
14. Manakul, P., et al.: Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models (2023), https://arxiv.org/abs/2303.08896
15. Narayan, S., et al.: Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization (2018)
16. Peng, B., Alcaide, E., et al.: Rwkv: Reinventing rnns for the transformer era. In: Findings of the Association for Computational Linguistics: EMNLP 2023 (2023)
17. See, A., et al.: Get to the point: Summarization with pointer-generator networks. In: ACL 2017. pp. 1073–1083. ACL (2017)
18. Yang, Z., et al.: Hotpotqa: A dataset for diverse, explainable multi-hop question answering (2018), https://arxiv.org/abs/1809.09600
19. Zhao, W.X., et al.: A survey of large language models (2024), https://arxiv.org/abs/2303.18223

A Appendix

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_18

LLM	AggTruth	Selector	Source	Target	Source			Target		Gap [%]
					Train	Val	Test	Test (1)	Test (2)	
llama-2-7b-chat-hf	CosSim	Lasso	QA	SUM.	0.858	0.853	**0.731**	0.587	0.713	5.639
		$\text{Spearman}_{1.0}$	SUM.	QA	0.884	0.875	0.659	0.716	**0.684**	3.380
	Entropy	$\text{Random}^{+}_{3,3}$	QA	SUM.	0.807	0.805	0.726	0.651	**0.721**	2.393
		$\text{Spearman}_{1.0}$	SUM.	QA	0.894	0.885	0.699	**0.742**	0.674	**0.844**
	JS-Div	$\text{Center}_{0.2}$	QA	SUM.	0.767	0.765	0.727	0.639	**0.721**	2.911
		$\text{Spearman}_{\text{auto}}$	SUM.	QA	0.839	0.833	0.674	0.684	0.609	7.793
	Sum	$\text{Random}^{+}_{3,3}$	QA	SUM.	0.802	0.799	0.723	**0.670**	0.710	**2.048**
		$\text{Spearman}_{1.0}$	SUM.	QA	0.894	0.885	**0.706**	0.724	0.660	1.975
llama-3.1-8B-Instruct	CosSim	$\text{Center}_{0.1}$	QA	SUM.	0.799	0.797	0.849	0.641	0.687	5.336
		$\text{Random}_{3,3}$	SUM.	QA	0.948	0.930	0.667	0.755	0.846	6.747
	Entropy	$\text{Spearman}_{\text{auto}}$	QA	SUM.	0.862	0.858	**0.861**	0.638	0.632	7.551
		$\text{Spearman}_{1.0}$	SUM.	QA	0.896	0.878	0.691	0.750	0.851	5.622
	JS-Div	$\text{Random}_{3,2}$	QA	SUM.	0.854	0.851	0.845	**0.689**	**0.722**	**1.554**
		$\text{Random}_{3,3}$	SUM.	QA	0.893	0.875	**0.715**	**0.766**	**0.853**	**3.786**
	Sum	$\text{Spearman}_{1.0}$	QA	SUM.	0.870	0.866	0.847	0.681	0.700	2.883
		—	SUM.	QA	0.928	0.910	0.705	0.750	0.850	4.979
phi-3.5-mini-instruct	CosSim	$\text{Spearman}_{\text{auto}}$	QA	SUM.	0.666	0.663	0.671	0.575	0.578	12.421
		Lasso	SUM.	QA	0.812	0.799	0.647	**0.709**	0.623	5.069
	Entropy	$\text{Spearman}_{0.5}$	QA	SUM.	0.801	0.795	0.713	0.634	0.637	4.580
		$\text{Random}_{3,2}$	SUM.	QA	0.854	0.843	**0.705**	0.666	0.634	3.736
	JS-Div	$\text{Spearman}_{1.0}$	QA	SUM.	0.797	0.790	0.712	**0.674**	**0.647**	**2.172**
		—	SUM.	QA	0.849	0.835	0.703	0.702	**0.636**	**2.029**
	Sum	$\text{Spearman}_{0.2}$	QA	SUM.	0.754	0.750	**0.718**	0.611	0.619	6.427
		$\text{Center}_{0.5}$	SUM.	QA	0.800	0.790	0.695	0.697	0.632	2.843
gemma-2-9b-it-bnb-4bit	CosSim	—	QA	SUM.	0.841	0.830	**0.793**	0.608	0.591	6.299
		$\text{Center}_{0.2}$	SUM.	QA	0.793	0.778	0.604	0.758	**0.637**	5.527
	Entropy	$\text{Center}_{0.5}$	QA	SUM.	0.840	0.834	0.734	**0.651**	**0.622**	**4.884**
		$\text{Spearman}_{0.1}$	SUM.	QA	0.749	0.742	0.624	0.690	0.612	8.724
	JS-Div	$\text{Center}_{0.2}$	QA	SUM.	0.772	0.764	0.756	0.591	0.593	8.566
		$\text{Spearman}_{0.2}$	SUM.	QA	0.738	0.732	**0.626**	**0.777**	0.625	**4.276**
	Sum	$\text{Center}_{0.1}$	QA	SUM.	0.732	0.725	0.783	0.597	0.606	6.531
		$\text{Spearman}_{\text{auto}}$	SUM.	QA	0.745	0.736	0.617	0.749	0.625	5.901

Table 6. Best obtained results (w.r.t. Gap value) of AggTruth methods for hallucination detection, grouped by LLM, aggregation method, and task type.

LLM	Source	Target	Test	Test (1)	Test (2)
llama-2-7b-chat-hf	QA	SUM.	0.752	0.670	0.739
	SUM.	QA	0.722	0.742	0.684
llama-3.1-8B-Instruct	QA	SUM.	0.886	0.689	0.722
	SUM.	QA	0.715	0.860	0.857
phi-3.5-mini-instruct	QA	SUM.	0.743	0.674	0.663
	SUM.	QA	0.705	0.730	0.649
gemma-2-9b-it-bnb-4bit	QA	SUM.	0.828	0.651	0.644
	SUM.	QA	0.690	0.790	0.637

Table 7. Best observed results in the conducted experiments for each LLM, task type, and test set.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_18