Hallucinated Span Detection with Multi-View Attention Features

Anonymous ACL submission

Abstract

This study addresses the problem of hallucinated span detection in the outputs of large language models. It has received less attention than output-level hallucination detection despite its practical importance. Prior work has shown that attentions often exhibit irregular patterns when hallucinations occur. Motivated by these findings, we extract features from the attention matrix that provide complementary views capturing (a) whether certain tokens are influential or ignored, (b) whether attention is biased toward specific subsets, and (c) whether a token is generated referring to a narrow or broad context, in the generation. These features are input to a Transformer-based classifier to conduct sequential labelling to identify hallucinated spans. Experimental results indicate that the proposed method outperforms strong baselines on hallucinated span detection with longer input contexts, such as data-to-text and summarisation tasks.

1 Introduction

Large Language Models (LLMs) have significantly advanced natural language processing and demonstrated high performance across tasks (Minaee et al., 2024). However, hallucinations persisting in texts generated by LLMs have been identified as a serious issue, which undermines LLM safety (Ji et al., 2024b).

To tackle this challenge, hallucination detection has been actively studied (Huang et al., 2025). Model-level (e.g., (Min et al., 2023)) or response-level (e.g., (Manakul et al., 2023)) hallucination detection has been proposed. However, identification of the hallucinated span is less explored despite its practical importance. Hallucinated span detection enables understanding and revising the problematic portion of the output. It also provides clues to mitigate hallucinations in LLM development.

To address this, we tackle hallucinated span detection. While there have been various types of hallucinations (Wang et al., 2024), this study targets hallucinations on contextualised generations that add baseless and contradictive information against the given input context. Motivated by the findings that irregular attention patterns are observed when hallucination occurs (Chuang et al., 2024; Zaranis et al., 2024), we extract features to characterise the distributions of attention weights. Specifically, the proposed method extracts an attention matrix from an LLM by inputting a set of prompt, context, and LLM output of concern. It then assembles features for each token from the attention matrix: average and diversity of incoming attention as well as diversity of outgoing attention, which complementarily capture the attention patterns of language models. The former two features indicate whether attention is distributed in a balanced manner for tokens in the output text. The last feature reveals if an output token was generated by broadly attending to other tokens. These features are then fed to a Transformer encoder with a conditional random field layer on top to conduct sequential labelling to determine whether a token is hallucinated or not.

Experimental results on hallucinated span detection confirmed that the proposed method outperforms strong baselines on data-to-text and summarisation tasks, improving token-level F1 score for 4.9 and 2.9 points, respectively. An in-depth analysis reveals that the proposed method is capable of handling longer input contexts. Our code is available at `https://anonymous_for_review`.

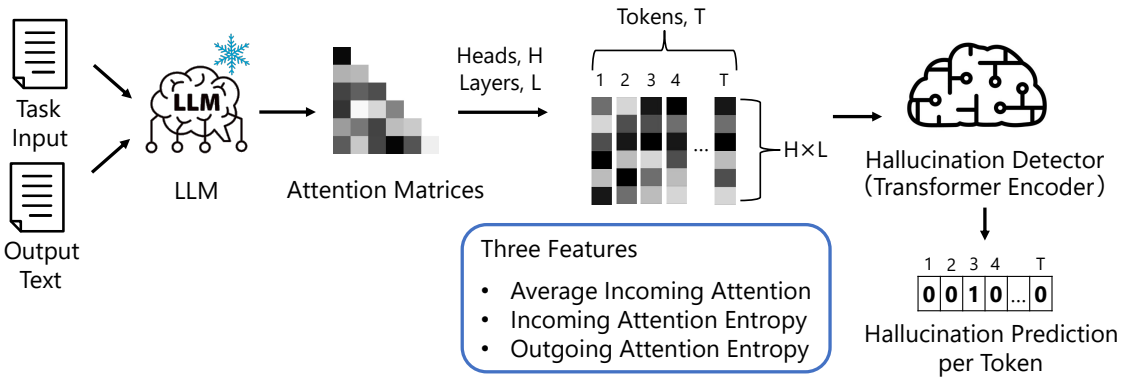

Figure 1: Overview of the proposed method

2 Related Work

This section discusses hallucination detection that utilises various internal states of LLMs.

Attention-Based Hallucination Detection Lookback Lens (Chuang et al., 2024) is the most relevant method to our study, which identifies hallucinations using only attention matrices. It computes the "Lookback" ratio of attention to assess whether generated tokens attend well to the input context. In contrast, our features primarily focus on the attention of output texts. ALTI+ (Ferrando et al., 2022; Zaranis et al., 2024) tracks token interactions across layers. ALTI+ has been applied to hallucination detection in machine translation, highlighting cases where the model fails to properly utilise source text information. A drawback of ALTI+ is its computational cost. It computes a token-to-token contribution matrix for each layer and for each attention head. Therefore, memory consumption linearly increases depending on the length of context and output as well as LLM sizes. Indeed, Zaranis et al. (2024) excluded sequences longer than 400 tokens due to GPU memory constraints.

Other Internal States for Hallucination Detection Hallucination detection has also explored various internal states of LLMs other than attention. Xiao and Wang (2021) and Zhang et al. (2023) identify hallucinations as tokens generated with anomalously low confidence based on the probability distribution in the final layer. Azaria and Mitchell (2023) and Ji et al. (2024a) use layer-wise Transformer block outputs to estimate hallucination risk. These studies assume that hallucination detection will be conducted on the same LLM generating output and can access such Transformer block outputs. In contrast, we empirically showed that the proposed method can also be applied to closed LLMs. Further, attention-based methods are distinctive from these studies in that they aim to model inter-token interactions.

3 Proposed Method

The proposed method is illustrated in Figure 1. It conducts sequential labelling, i.e., predicts binary labels that indicate whether a token in text, which has been generated by a certain LLM, is hallucinated or not. Specifically, the proposed method takes a set of prompt, input context, and output generated by an LLM of concern as input to another LLM and obtains the attention matrix of the output text span. It then extracts features from the attention matrix (Sections 3.1 and 3.2). These features are fed to a Transformer encoder model with the prediction head of a conditional random field (CRF) to conduct sequential labelling to identify hallucinated spans (Section 3.3). As the attention matrix provides crucial information for our method, we compare the raw attention and a variation based on the analysis of attention mechanism (Kobayashi et al., 2020) (Section 3.4). We remark that only the hallucination detection model needs training, i.e., the LLM for attention matrix extraction is kept frozen, which makes our method computationally efficient.

Our method applies to both scenarios where the LLM that generated outputs and the LLM for hallucinated span detection are the same or different. In practice, the latter setting is expected to be more common in an era where

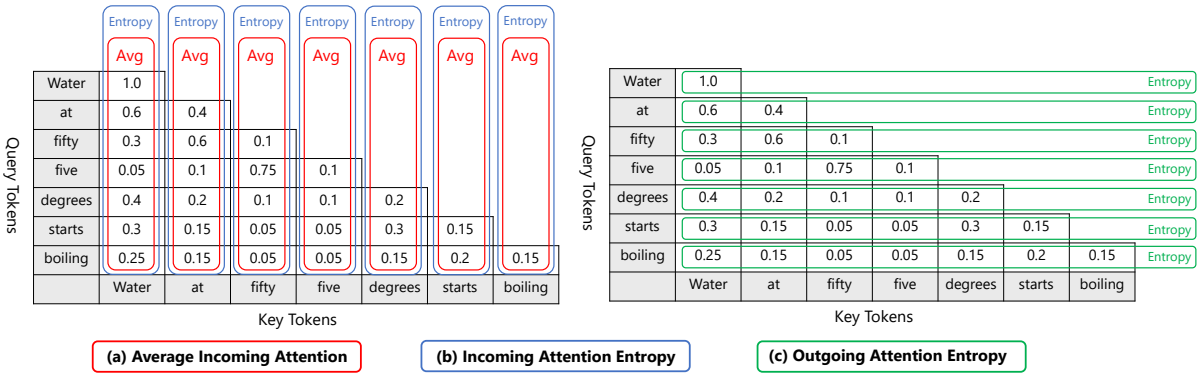

Figure 2: Feature extraction from attention matrix (these attention values are for illustrative purposes.)

LLMs are widely used for writing tasks. In addition, we cannot access the internal state of proprietary LLMs. Our experiments assume the scenario where the LLM for generation and the LLM for detection are different.

3.1 Feature Design

Previous studies revealed that irregular patterns of attention are incurred when hallucination occurs (Chuang et al., 2024; Zaranis et al., 2024). Based on these findings, we design features to complementarily capture irregular attentions. Specifically, we extract features providing complementary views of the attention matrix as shown in Figure 2: (a) average attention a token receives (**Average Incoming Attention**), (b) diversity of attention a token receives (**Incoming Attention Entropy**), and (c) diversity of tokens that a token attends to (**Outgoing Attention Entropy**).

Average Incoming Attention We compute the average attention weights that a token receives when generating others. This feature indicates whether certain tokens are influential or ignored in generation. Specifically, it computes the average attention weight in the key direction on the attention matrix as illustrated on the left side of Figure 2.

Incoming Attention Entropy This feature captures the diversity of attention weights, i.e., whether attention is biased toward specific subsets or is more uniformly distributed. It computes the entropy of attention weights in the key direction on the attention matrix as illustrated on the left side of Figure 2.

Outgoing Attention Entropy The final feature models the diversity of tokens that a token attends to when being generated. This indicates whether the model references a narrow or broad range of context for generating the token. Specifically, this feature computes the entropy of attention weights in the query direction on the attention matrix as illustrated on the right side of Figure 2.

Given the complex and diverse nature of attention dynamics, we do not regard individual features as independently effective. Rather, we assume these features *complementary* capture irregular attention patterns due to hallucination by providing views from different angles.

3.2 Feature Extraction

We extract these features for each token from the attention matrix. As notation, the output by an LLM to detect hallucinated span consists of T tokens. The LLM for attention matrix extraction consists of L layers of a Transformer decoder with H heads of multi-head attention.

Average Incoming Attention This feature computes the average attention weights that a token receives when generating other tokens. The attention matrix $\boldsymbol{A}$ is lower triangular due to masked self-attention, meaning each query token i attends only to key tokens j with $1 \leq j \leq i$. Thus, earlier tokens receive attention more often, and tokens close to the end receive attention less often. To compensate for the imbalanced frequency, we adjust the attention weights $\alpha_{i,j}$ as:

$$\alpha'_{ij} = \alpha_{ij} \cdot i. \tag{1}$$

Using the adjusted attention matrix $\boldsymbol{A}'$, the average attention that a key token j receives

is computed as:

$$\mu_j^{(\ell,h)} = \frac{1}{T-j+1} \sum_{i=j}^{T} \alpha'^{(\ell,h)}_{ij}, \quad (2)$$

where $1 \leq \ell \leq L$ is the layer index and $1 \leq h \leq H$ is the head index. The final feature vector is obtained by concatenating the average attention weights across all layers and heads:

$$\boldsymbol{v}(j) = [\mu_j^{(1,1)}, \mu_j^{(1,2)}, \ldots, \mu_j^{(L,H)}] \in \mathbb{R}^{LH} \quad (3)$$

Incoming Attention Entropy To model the diversity of attention a token receives, we use the entropy of the weights. As discussed in the previous paragraph, the attention matrix is lower triangular. To compensate for different numbers of times to receive attention, we normalise an entropy value by dividing by the maximum entropy:

$$\beta_j^{(\ell,h)} = \frac{-\sum_{i=j}^{T} \kappa_{ij}^{(\ell,h)} \log \kappa_{ij}^{(\ell,h)}}{\log(T-j+1)}, \quad (4)$$

$$\kappa_{ij}^{(\ell,h)} = \frac{\alpha'^{(\ell,h)}_{ij}}{\sum_{k=1}^{i} \alpha'^{(\ell,h)}_{ik}}. \quad (5)$$

The final feature vector is a concatenation of the entropy values across layers and heads:

$$\boldsymbol{e}(j) = [\beta_j^{(1,1)}, \beta_j^{(1,2)}, \ldots, \beta_j^{(L,H)}] \in \mathbb{R}^{LH} \quad (6)$$

Outgoing Attention Entropy This feature models the diversity of tokens that a token attends to when being generated. Similar to the "Incoming Attention Entropy" feature, we compute the entropy of attention weights of query tokens[1] by dividing by the maximum entropy:

$$\gamma_i^{(\ell,h)} = \frac{-\sum_{j=1}^{i} \alpha_{ij}^{(\ell,h)} \log \alpha_{ij}^{(\ell,h)}}{\log(i)}. \quad (7)$$

The final feature vector is a concatenation of the entropy values across layers and heads:

$$\hat{\boldsymbol{e}}(i) = [\gamma_i^{(1,1)}, \gamma_i^{(1,2)}, \ldots, \gamma_i^{(L,H)}] \in \mathbb{R}^{LH} \quad (8)$$

Final Feature Vector The three features $\boldsymbol{v}(j)$ (Average Incoming Attention), $\boldsymbol{e}(j)$ (Incoming Attention Entropy), and $\hat{\boldsymbol{e}}(i)$ (Outgoing Attention Entropy) are concatenated as a final feature vector for hallucination detection. Each feature has LH elements; thus, the final feature vector consists of $3LH$ elements.

[1]Remind that attention weights are normalised in the query direction.

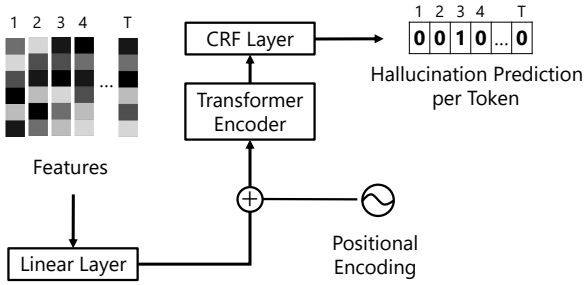

Figure 3: Hallucination Detector

3.3 Hallucination Detector

Our hallucination detector consists of a linear layer, a Transformer encoder layer, and a CRF layer on top, as illustrated in Figure 3. To handle *spans*, we employ the CRF layer to model dependencies between adjacent tokens, improving the consistency of hallucinated spans compared to independent token-wise classification.[2] The CRF has been successfully integrated with Transformer-based models for structured NLP tasks (Yan et al., 2019; Wang et al., 2021).

Feature vectors are first standardised to have *zero* mean and 1 standard deviation per feature type. After standardisation, the feature vector first goes through a linear layer for transformation, which is primarily employed to adapt to various LLMs that can have different numbers of layers and attention heads. Then the transformed vector is input to the transformer layer with positional encoding to incorporate token order information. Finally, the CRF layer predicts a binary label indicating whether a token is hallucinated (label 1) or not (label 0). During inference, the Viterbi algorithm determines the most likely label sequences.

3.4 Attention Weights

Attention weights have been used to analyse context dependency (Clark et al., 2019; Kovaleva et al., 2019; Htut et al., 2019) of Transformer models. Recently, Kobayashi et al. (2020) revealed that the norm of the transformed input vector plays a significant role in the attention mechanism. They reformulated the computation in the Transformer as:

$$\boldsymbol{y}_i = \sum_{j=1}^{T} \alpha_{i,j} f(\boldsymbol{x}_j) \quad (9)$$

[2]We empirically confirmed that a linear layer is inferior to CRF in our study.

Dataset	QA	Data2Text	Summarisation
train	4,584 (1,421) (31.0%)	4,848 (3,360) (69.3%)	4,308 (1,347) (31.3%)
valid	450 (143) (31.8%)	450 (315) (70.0%)	450 (135) (30.0%)
test	900 (160) (17.8%)	900 (579) (64.3%)	900 (204) (22.7%)
Total	5,934 (1,724) (29.1%)	6,198 (4,254) (68.6%)	5,658 (1,686) (29.8%)

Table 1: Number of samples in the RAGTruth dataset (Numbers in parentheses indicate the raw number of and percentage of sentences containing at least one hallucination span.)

where $\alpha_{i,j}$ is the raw attention weight and $f(\boldsymbol{x}_j)$ is the transformed vector of input $\boldsymbol{x}_j$. The transformation function is defined as:

$$f(\boldsymbol{x}) = \left(\boldsymbol{x}\boldsymbol{W}^V + \boldsymbol{b}^V\right)\boldsymbol{W}^O, \quad (10)$$

where $\boldsymbol{W}^V \in \mathbb{R}^{d_{\text{in}} \times d_v}$ and $\boldsymbol{b}^V \in \mathbb{R}^{d_v}$ are the parameters for value transformations and $\boldsymbol{W}^O \in \mathbb{R}^{d_v \times d_{\text{out}}}$ is the output matrix multiplication. Kobayashi et al. (2020) found that frequently occurring tokens often receive high attention weights but have small vector norms, reducing their actual contribution to the output. This suggests that attention mechanisms adjust token influence, prioritising informative tokens over frequent but less meaningful ones.

This study compares the effectiveness of raw and the transformed attention weights of Kobayashi et al. (2020). Specifically, we employ the adjusted attention matrix $\boldsymbol{A}_{\text{norm}}$ defined as:

$$\boldsymbol{A}_{\text{norm}} = \boldsymbol{A} \cdot \text{diag}(\|f(\boldsymbol{x})\|), \quad (11)$$

where $\boldsymbol{A}$ is the raw attention weight matrix, and $\text{diag}(\|f(\boldsymbol{x})\|)$ represents a diagonal matrix containing the transformed vector norms.

4 Evaluation

We evaluate the effectiveness of the proposed method for hallucinated span detection.

4.1 Dataset

As the dataset providing hallucination *span* annotation, we employ RAGTruth (Niu et al., 2024)[3], a benchmark dataset that annotates responses generated by LLMs (GPT-3.5-turbo-0613, GPT-4-0613, Llama-2-7B-chat, Llama-2-13B-chat, Llama-2-70B-chat, and Mistral-7B-Instruct). It covers three scenarios of using LLMs in practice, i.e., question answering (QA), data-to-text generation (Data2Text), and news summarisation (Summarisation). RAGTruth provides 18,000 annotated responses, where hallucinated spans in each response are tagged at the character level. The number of samples is shown in Table 1. As there is no official validation split in RAGTruth, we randomly sampled 450 instances (75 IDs) from the training set for validation.

4.2 Evaluation Metric

Although RAGTruth labels hallucinations at the character level, we convert these labels into the token level for intuitive interpretation of evaluation results. We employed the same tokeniser of LLM to extract attention matrices.

We compute the token-level precision (Prec) and recall (Rec). Given a set of gold-standard hallucination tokens $\mathcal{Y} = \{y_0, y_1, \cdots, y_N\}$ and predicted hallucination tokens $\hat{\mathcal{Y}} = \{\hat{y}_0, \hat{y}_1, \cdots, \hat{y}_M\}$,

$$\text{precision} = \frac{|\hat{\mathcal{Y}} \cap \mathcal{Y}|}{|\hat{\mathcal{Y}}|}, \text{recall} = \frac{|\hat{\mathcal{Y}} \cap \mathcal{Y}|}{|\mathcal{Y}|}. \quad (12)$$

Matching of the gold-standard and predicted tokens is computed in the context of output texts. The primary evaluation metric is the F1 score of token-level hallucination predictions, which is the harmonic mean of precision and recall. Following the RAGTruth evaluation scheme, we used the micro-average of precision, recall, and F1.

4.3 Implementation

The proposed method consists of the linear layer, the Transformer encoder layer, and the CRF layer. The settings of the Transformer layer, i.e., the numbers of layers and attention heads, the dimensions, and the dropout rate, were tuned together with other hyperparameters of learning rate and weight decay using the Data2Text task, as it provides the largest samples. We apply the same hyperparameters

[3] https://github.com/ParticleMedia/RAGTruth

Methods	LLM	QA			Data2Text			Summarisation		
		Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
Ours_{raw}	Llama	47.7	**68.7**	56.3	**55.6**	55.0	**55.3**	51.1	36.7	42.7
$\text{Ours}_{\text{norm}}$		57.4	54.0	55.6	53.4	**57.1**	55.2	51.0	**39.5**	**44.5**
Fine-tuning		**62.8**	56.9	**59.7**	55.4	46.2	50.4	**52.0**	34.6	41.6
Lookback Lens		53.5	7.6	13.2	0.0	0.0	0.0	0.0	0.0	0.0

Table 2: Hallucinated span detection results on Llama-3-8B-Instruct. The proposed method is denoted as "Ours" with variations of raw attention ("raw") or the transformed attention ("norm"). It outperformed the baselines on tasks with longer input contexts, i.e., Data2Text and Summarisation.

for other tasks. The specific hyperparameter search range is in Appendix A. We employed early stopping on training: training was terminated if the F1 score on the validation set did not improve for 10 consecutive epochs.

As the LLM to obtain attention matrices, we employed the recent smaller yet strong models of Llama-3-8B-Instruct (Touvron et al., 2023; Llama Team, 2024) and Qwen2.5-7B-Instruct (Team, 2025) (see Appendix B.3 for details). We adapted the template by Niu et al. (2024) for promoting. Notice that these LLMs are different from the ones used to create the RAGTruth dataset, which simulates the scenario where we cannot access the LLMs generated outputs for hallucinated span detection.

4.4 Baselines

We compared the proposed method to two baselines employing the same LLMs as our method.

Fine-tuned LLMs Although straightforward, fine-tuned LLMs serve as a strong baseline (Niu et al., 2024). We fine-tuned the LLMs using the prompt of Niu et al. (2024) with instructions to predict hallucinated spans. More details are provided in Appendix B.4.

Lookback Lens We employed Lookback Lens (Chuang et al., 2024), which also utilises the attention matrix for hallucination detection. It computes the "Lookback" ratio; the ratio of attention weights on the input context versus newly generated tokens. The Lookback feature is input to a logistic regression model to predict the probability of a token being hallucinated.[4] We regarded tokens for which the predicted probabilities are equal to or larger than 0.5 as hallucination, following the traits of the logistic regression classifier. We used the author's implementation[5] for the Lookback Lens model training.

[4]Lookback Lens can also conduct span-level prediction by segmenting texts using a sliding window. For direct comparison to our method, we used the token-level variant (i.e., window size is one).

	QA		Data2Text		Summ.	
	In	Out	In	Out	In	Out
Mean	400	140	788	199	723	136
Max	646	437	1,499	406	2,063	412
Min	244	9	517	69	225	16

Table 3: Numbers of tokens of context ('In') and output ('Out') (measured using Llama-3-8B-Instruct tokeniser).

4.5 Experimental Results

The experimental results on Llama-3-8B-Instruct are shown in Table 2. The proposed method is denoted as "Ours" with variations of using raw attention weights (denoted as "raw") and the transformed attention weights (denoted as "norm").

The proposed method outperformed both the fine-tuning and Lookback Lens for hallucinated span detection in Data2Text and summarisation, achieving the highest token-level F1 scores. On QA, the proposed method tends to have higher recall yet lower precision, i.e., it tends to overly detect hallucinations. A possible factor is shorter lengths of input context. Table 3 shows the numbers of tokens in context and output texts. QA has significantly shorter contexts on average compared to Data2Text and summarisation, while the output lengths are similar. This result may imply that the proposed method better handles tasks where consistency with long context is important, like

[5]https://github.com/voidism/Lookback-Lens

Source text: [...] From the giant sequoias of Yosemite to the geysers of Yellowstone, the United States' national parks were made for you and me. And for Saturday and Sunday, they're also free. Though most of the National Park Service's 407 sites are free year-round, the 128 parks that charge a fee - like Yellowstone and Yosemite - will be free those two days. It's all part of National Park Week, happening April 18 through April 26, and it's hosted by the National Park Service and the National Park Foundation. [...]
Output summary: National Park Service offers free admission to 128 parks, including Yellowstone and Yosemite, on April 18-19 and 25-26, as part of National Park Week.
Ground Truth: on April 18-19 and 25-26 **Ours$_{\text{raw}}$:** April 18-19 and 25-26 **Fine-tuning:** - (Detection failed)

Table 4: Hallucination detection example (Summarisation)

Methods	QA				Data2Text				Summarisation			
	0–2	2–4	4–6	6–8	0–2	2–4	4–6	6–8	0–2	2–4	4–6	6–8
Ours$_{\text{raw}}$	27.7	–	48.6	59.4	**33.0**	–	**52.6**	**63.3**	0.0	**42.3**	28.5	54.4
Ours$_{\text{norm}}$	25.1	–	41.1	61.0	**33.0**	–	51.2	61.9	0.0	41.9	30.5	**59.0**
Fine-tuning	**38.4**	–	**52.7**	**62.3**	23.8	–	45.8	57.9	0.0	41.0	**31.4**	56.4

Table 5: Token-level F1 scores of hallucinated span detection per different hallucination ratios (Llama-3-8B-Instruct). "–" indicates there was no sample falling in the corresponding bin.

summarisation. We conduct further analysis in Sections 4.6 and 4.7.

For attention weights, the effectiveness of the raw and transformed attention weights depends on tasks. The raw attention weights performed higher in QA, while the transformed weights outperformed the raw attention in summarisation, and they are comparable on Data2Text.

Lookback Lens consistently exhibited the lowest F1 scores.[6] Our inspection confirmed that Lookback Lens overfitted the majority class, i.e., no hallucination. Hallucinated spans are much more infrequent compared to the no-hallucination tokens. This implies that making a binary decision based on the predicted hallucination probability is non-trivial. Furthermore, Lookback Lens seems to have struggled to handle longer input contexts, i.e., Data2Text and summarisation tasks, in contrast to the proposed method. This may be because the Lookback Lens strongly depends on attention weights for the input context. We evaluated the combination of features of Lookback Lens and ours to see if they are complementary. As a result, no improvement was observed; possibly because our "Outgoing Attention Entropy" feature also takes the input context into account.

[6]This looks largely different from the original paper. We remark that in addition to the experimental dataset difference, the original paper reported AUROC.

Table 4 presents an example of hallucination detection on summarisation. In the output text, the red-coloured span indicates the hallucination. While the Fine-tuning failed to detect the hallucination, the proposed method successfully identified the span very close to the ground truth (only missing a preposition). Further examples are in Appendix C.

4.6 Effects of Hallucination Ratio

Intuitively, the ratio of hallucinated tokens in a text affects the performance. When the frequency of hallucinations is small, detection should become more challenging. Table 5 shows the token-level F1 scores on different percentages of hallucinated tokens. These results confirm that the intuition holds true. Across methods and tasks, higher F1 scores were achieved when hallucinated tokens were more frequent.

Another interesting observation is that the effect of task type is dominant than the hallucinated token ratio. Table 5 shows that the superior method is consistent across different frequencies of hallucinated tokens within the same task.

4.7 Effects of Hallucination Type

We further analysed the hallucination detection capability of the proposed method for different hallucination types. RAGTruth categorises hal-

Methods	LLM	QA			Data2Text			Summarisation		
		Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
Ours_{raw}	Qwen	38.5	**73.7**	50.6	53.5	**57.1**	55.2	49.6	**35.7**	**41.5**
$\text{Ours}_{\text{norm}}$		39.0	64.7	48.7	55.5	55.3	**55.4**	49.3	33.6	39.9
Fine-tuning		**60.1**	57.1	**58.6**	**58.9**	51.4	54.9	**62.0**	30.0	40.4
Lookback Lens		46.6	5.6	9.9	50.0	0.0	0.0	0.0	0.0	0.0

Table 6: Hallucinated span detection results on Qwen2.5-7B-Instruct

QA (Total Tokens: 124,817)					
Methods	SInfo	EInfo	SConf	EConf	All
Ours_{raw}	**74.1**	**74.4**	–	4.0	**68.7**
$\text{Ours}_{\text{norm}}$	50.6	60.0	–	3.8	54.0
Fine-tuning	48.7	63.8	–	**7.8**	56.9
Hal. Tokens	1,020	4,742	–	501	6,263
Data2Text (Total Tokens: 178,343)					
Methods	SInfo	EInfo	SConf	EConf	All
Ours_{raw}	29.4	50.5	**7.3**	64.7	55.5
$\text{Ours}_{\text{norm}}$	**37.8**	**52.7**	**7.3**	**64.8**	**57.1**
Fine-tuning	35.8	51.6	0.0	43.7	46.2
Hal. Tokens	595	3,118	41	3,580	7,334
Summarisation (Total Tokens: 121,248)					
Methods	SInfo	EInfo	SConf	EConf	All
Ours_{raw}	**65.2**	46.5	**8.5**	16.4	36.7
$\text{Ours}_{\text{norm}}$	49.7	**51.3**	**8.5**	18.5	**39.5**
Fine-tuning	44.9	43.7	8.1	**18.6**	34.6
Hal. Tokens	187	2,067	71	1,160	3,485

Table 7: Recall of hallucinated span detection per hallucination type (Llama-3-8B-Instruct)

lucinations into four types: Subtle Introduction of Baseless Information (**SInfo**) and Evident Introduction of Baseless Information (**EInfo**) indicate whether the output text subtly adds information or explicitly introduces falsehoods. Subtle Conflict (**SConf**) and Evident Conflict (**EConf**) indicate whether the output alters meaning or directly contradicts the input text. For more details, see Niu et al. (2024).

Table 7 shows detection recalls for different hallucination types.[7] For Data2Text, the recall of Evident Conflict is significantly higher than SInfo and EInfo. This result indicates that the proposed method better captures conflicting information against input context than baseless information introduced by LLMs. The trend is the opposite on QA and summarisation, where the proposed method achieved much higher recall on SInfo and EInfo than on SConf and EConf, which implies that baseless information was easier to capture for the proposed method. These results indicate that detection difficulties of different hallucination types can vary depending on tasks.

[7]Precision (and thus F1) is difficult to compute because it is non-trivial to decide to which category does detected hallucination belong.

4.8 Performance on Qwen

Table 6 shows the results on Qwen2.5-7B-Instruct. While the results are consistent with Table 2, Qwen was consistently inferior to Llama regarding the proposed method, which should be attributed to different implementations of their attention mechanisms. Qwen has fewer numbers of layers and attention heads, and thus its feature dimension is smaller than Llama. In addition, the parameters in multi-head attention are more aggressively shared in Qwen. These differences may affect the attention features extracted from Qwen. More details of the differences between Llama and Qwen are discussed in Appendix B.3.

5 Conclusion

We proposed the hallucinated span detection method using features that assemble attention weights from different views. Our experiments confirmed that these features are useful in combination for detecting hallucinated spans, outperforming a previous method that also uses attention weights.

This study focused on hallucination detection, but our method may also apply to broader abnormal behaviour detection of LLMs. As future work, we plan to explore its potential for detecting backdoored LLMs (He et al., 2023), which behave normally on regular inputs but produce malicious outputs when triggered. Since our approach analyses attention distributions, it may detect anomalous attention patterns caused by the triggers.

Limitations

While we confirmed the effectiveness of the proposed method on two models: Llama-3-8B-Instruct and Qwen2.5-7B-Instruct, there are lots more LLMs. The effectiveness of our method when applied to attention mechanisms from other models remains unverified. In addition, our experiments are limited to the English language. We will explore the applicability of our method to other languages by employing multilingual LLMs.

Our method requires training data that annotates hallucinated spans, which is costly to create. A potential future direction is an exploration of an unsupervised learning approach. The success of the current method implies that our features successfully capture irregular attention patterns on hallucination. We plan to train our method only on non-hallucinated human-written text. We then identify hallucinations as instances in which attention patterns deviate from the learned normal patterns.

References

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. 2023. GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4895–4901.

Amos Azaria and Tom Mitchell. 2023. The Internal State of an LLM Knows When It's Lying. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 967–976.

Yung-Sung Chuang, Linlu Qiu, Cheng-Yu Hsieh, Ranjay Krishna, Yoon Kim, and James R. Glass. 2024. Lookback Lens: Detecting and Mitigating Contextual Hallucinations in Large Language Models Using Only Attention Maps. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1419–1436.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What Does BERT Look at? An Analysis of BERT's Attention. In *Proceedings of the Workshop on Analysing and Interpreting Neural Networks for NLP (BlackboxNLP)*, pages 276–286.

Javier Ferrando, Gerard I. Gállego, Belen Alastruey, Carlos Escolano, and Marta R. Costa-jussà. 2022. Towards Opening the Black Box of Neural Machine Translation: Source and Target Interpretations of the Transformer. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8756–8769.

Xuanli He, Qiongkai Xu, Jun Wang, Benjamin Rubinstein, and Trevor Cohn. 2023. Mitigating Backdoor Poisoning Attacks through the Lens of Spurious Correlation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 953–967.

Phu Mon Htut, Jason Phang, Shikha Bordia, and Samuel R. Bowman. 2019. Do Attention Heads in BERT Track Syntactic Dependencies? *arXiv:1911.12246*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 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, Delong Chen, Etsuko Ishii, Samuel Cahyawijaya, Yejin Bang, Bryan Wilie, and Pascale Fung. 2024a. LLM Internal States Reveal Hallucination Risk Faced With a Query. In *Proceedings of the Workshop on Analysing and Interpreting Neural Networks for NLP (BlackboxNLP)*, pages 88–104.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2024b. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2020. Attention is Not Only a Weight: Analyzing Transformers with Vector Norms. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7057–7075.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the Dark Secrets of BERT. In *Proceedings of Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4365–4374.

AI @ Meta Llama Team. 2024. The Llama 3 Herd of Models. *arXiv:2407.21783*.

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

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi.

2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 12076–12100.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2024. Large Language Models: A Survey. *arXiv:2402.06196*.

Cheng Niu, Yuanhao Wu, Juno Zhu, Siliang Xu, KaShun Shum, Randy Zhong, Juntong Song, and Tong Zhang. 2024. RAGTruth: A Hallucination Corpus for Developing Trustworthy Retrieval-Augmented Language Models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 10862–10878.

Qwen Team. 2025. Qwen2.5 technical report. *arXin:2412.15115*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. *arXiv:2302.13971*.

Chenyi Wang, Tianshu Liu, and Tiejun Zhao. 2021. HITMI&T at SemEval-2021 Task 5: Integrating Transformer and CRF for Toxic Spans Detection. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval)*, pages 870–874.

Yuxia Wang, Haonan Li, Xudong Han, Preslav Nakov, and Timothy Baldwin. 2024. Do-Not-Answer: Evaluating Safeguards in LLMs. In *Findings of the Association for Computational Linguistics: EACL*, pages 896–911.

Yijun Xiao and William Yang Wang. 2021. On Hallucination and Predictive Uncertainty in Conditional Language Generation. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 2734–2744.

Hang Yan, Bocao Deng, Xiaonan Li, and Xipeng Qiu. 2019. TENER: Adapting Transformer Encoder for Named Entity Recognition. *arXiv:1911.04474*.

Emmanouil Zaranis, Nuno M Guerreiro, and Andre Martins. 2024. Analyzing Context Contributions in LLM-based Machine Translation. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 14899–14924.

Tianhang Zhang, Lin Qiu, Qipeng Guo, Cheng Deng, Yue Zhang, Zheng Zhang, Chenghu Zhou, Xinbing Wang, and Luoyi Fu. 2023. Enhancing Uncertainty-Based Hallucination Detection with Stronger Focus. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 915–932.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. 2024. LlamaFactory: Unified Efficient Fine-Tuning of 100+ Language Models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 400–410.

A Details of Transformer Encoder Training

In this study, we used the Optuna library[8] to perform hyperparameter optimisation shown in Table 8. The training was terminated if the F1 score on the validation dataset did not improve for 10 consecutive epochs. The setting of the model with the highest F1 score was selected for formal evaluation.

Hyperparameter	**Search Range**
Learning rate	1e-5 $\sim$ 1e-3
Number of layers	$[2, 4, 6, 8, 10, 12, 14, 16]$
Number of heads	$[4, 8, 16, 32]$
Dropout rate	0.1 $\sim$ 0.5
Weight decay	1e-6 $\sim$ 1e-2
Model dimension	$[256, 512, 1024]$
Parameter	**Setting**
Optimizer	AdamW
Batch size	64 (Summ. : 32)
Number of trials	200 (Summ. : 100)
Maximum epochs	150

Table 8: Settings of Transformer encoder

B Details of Experiment Settings

B.1 Computational Environment

All the experiments were conducted on NVIDIA RTX A6000 (48GB memory) GPUs. For training the Transformer encoder of the proposed method, we used 2 GPUs. For fine-tuning the LLM, we used 4 GPUs in parallel.

B.2 Prompts and Preprocessing of RAGTruth

The prompts used in our experiments are shown in Table 10 and Table 11.

[8] https://optuna.org/

Parameter	Value
Fine-tuning method	full fine-tuning
Learning rate	5e-6
Batch size	1
Number of epochs	3
Optimizer	AdamW
Warmup steps	10

Table 9: Fine-tuning Parameters

The hallucination labels in RAGTruth are provided at the character span level. For example, a hallucination might be annotated with "start": 219, "end": 229. Character span labels were converted into token-level labels.

B.3 LLM Details

Llama-3-8B-Instruct has 32 layers and 32 attention heads, while Qwen2.5-7B-Instruct has 28 layers and 28 heads. Both models replace standard Multi-Head Attention (MHA) with Grouped-Query Attention (GQA) (Ainslie et al., 2023), but Llama-3 uses more layers and heads than Qwen2.5.

MHA assigns each query to a single key-value pair, whereas GQA allows multiple queries to share a key-value pair, reducing the number of trainable parameters. Llama-3-8B-Instruct processes 32 queries while reducing the number of keys and values to 8, so each key-value pair corresponds to 4 queries. In contrast, Qwen2.5-7B-Instruct processes 28 queries and reduces the number of keys and values to 4, making each key-value pair correspond to 7 queries.

We conjecture these differences were reflected in the different performances of Llama and Qwen in our method.

B.4 Fine-Tuning

Fine-tuning was conducted using LLaMA-Factory (Zheng et al., 2024)[9], a library specialized for fine-tuning LLMs. The fine-tuning parameters are shown in Table 9. The fine-tuned model predicts the hallucinated span by predicting character indexes. If a hallucination label changes within a single token in predictions, the entire token is considered as being hallucinated.

[9] https://github.com/hiyouga/LLaMA-Factory

C Hallucination Detection Examples

Table 12 presents hallucination detection results in the QA task. The Fine-tuning baseline incorrectly judged the non-hallucinated span as hallucinated and largely overlooked the truly hallucinated span. In contrast, the proposed method mostly correctly identified the hallucinated span.

Table 13 presents hallucination detection results in the summarisation task where the proposed method failed. In the first example, the proposed method overlooked the hallucinated span. In the second example, the proposed method mistook the non-hallucinated span as hallucinated.

QA Prompt

Original text (including tokens):

```
<|begin_of_text|><|start_header_id|>system<|end_header_id|>
You are an excellent system, generating output according to the instructions.
<|eot_id|><|start_header_id|>user<|end_header_id|>
Briefly answer the following question:
{question}
Bear in mind that your response should be strictly based on the following three
passages:
{passages}
In case the passages do not contain the necessary information to answer the question,
please reply with:
"Unable to answer based on given passages."
output:
<|eot_id|><|start_header_id|>assistant<|end_header_id|>
{answer} <|eot_id|>
```

Data2Text Prompt

Original text (including tokens):

```
<|begin_of_text|><|start_header_id|>system<|end_header_id|>
You are an excellent system, generating output according to the instructions.
<|eot_id|><|start_header_id|>user<|end_header_id|>
Instruction:
Write an objective overview about the following local business based only on the
provided structured data in the JSON format.
You should include details and cover the information mentioned in the customers'
review.
The overview should be 100 - 200 words.  Don't make up information.
Structured data:
{json_data}
Overview:
<|eot_id|><|start_header_id|>assistant<|end_header_id|>
{Converted text} <|eot_id|>
```

Summarisation Prompt

Original text (including tokens):

```
<|begin_of_text|><|start_header_id|>system<|end_header_id|>
You are an excellent system, generating output according to the instructions.
<|eot_id|><|start_header_id|>user<|end_header_id|>
Summarize the following news within {word count of the summary} words:
{text to summarize}
output:
<|eot_id|><|start_header_id|>assistant<|end_header_id|>
{summary} <|eot_id|>
```

Table 10: Prompts for RAGTruth (Using Llama-3-8B-Instruct)

QA Prompt

Original text (including tokens):

```
<|im_start|>system
You are an excellent system, generating output according to the
instructions.<|im_end|>
<|im_start|>user
Briefly answer the following question:
{question}
Bear in mind that your response should be strictly based on the following three
passages:
{passages}
In case the passages do not contain the necessary information to answer the question,
please reply with:
"Unable to answer based on given passages."
output:<|im_end|>
<|im_start|>assistant
{answer}<|im_end|>
```

Data2Text Prompt

Original text (including tokens):

```
<|im_start|>system
You are an excellent system, generating output according to the
instructions.<|im_end|>
<|im_start|>user
Instruction:
Write an objective overview about the following local business based only on the
provided structured data in the JSON format.
You should include details and cover the information mentioned in the customers'
review.
The overview should be 100 - 200 words.  Don't make up information.
Structured data:
{json_data}
Overview:<|im_end|>
<|im_start|>assistant
{Converted text}<|im_end|>
```

Summarisation Prompt

Original text (including tokens):

```
<|im_start|>system
You are an excellent system, generating output according to the
instructions.<|im_end|>
<|im_start|>user
Summarize the following news within {word count of the summary} words:
{text to summarize}
output:<|im_end|>
<|im_start|>assistant
{summary}<|im_end|>
```

Table 11: Prompts for RAGTruth (Using Qwen2.5-7B-Instruct)

Key Part of the Question

`Briefly answer the following question: differences between elements, compounds, and mixtures. Bear in mind that your response should be strictly based on the following three passages: ...` (Following this, three passages appear, but there is no information on the bonding of elements or compounds. The content primarily focuses on mixtures.)

Ground Truth Label

`... 3. Separation: Elements can be separated easily by physical means, while separating compounds requires more complex methods like chemical reactions. It is easy to physically separate the components of a mixture. 4. Bonding: Elements are held together by covalent bonds, while compounds can have various types of bonds (covalent, ionic, metallic, etc.). Mixtures do not have any bonds between their components.`

Ours$_{\text{raw}}$

`... 3. Separation: Elements can be separated easily by physical means, while separating compounds requires more complex methods like chemical reactions. It is easy to physically separate the components of a mixture. 4. Bonding: Elements are held together by covalent bonds, while compounds can have various types of bonds (covalent, ionic, metallic, etc.). Mixtures do not have any bonds between their components.`

Fine-tuning

`... 3. Separation: Elements can be separated easily by physical means, while separating compounds requires more complex methods like chemical reactions. It is easy to physically separate the components of a mixture. 4. Bonding: Elements are held together by covalent bonds, while compounds can have various types of bonds (covalent, ionic, metallic, etc.). Mixtures do not have any bonds between their components.` (Incorrect detection in point 3 + incomplete detection in point 4. The discussion on separation in point 3 is fully described in the original text.)

Table 12: Hallucination detection example (QA)

Example 1
Key Part of the Target Sentence for Summarisation `... Doug Ducey signed legislation to allow Arizonans to get any lab test without a doctor's order. Freedom of information - always sounds like a good thing. ...` (The target sentence for summarisation contains no mention of Doug Ducey being the governor of Texas. In fact, he was a former governor of Arizona, making this incorrect.)
Ground Truth Label `The article discusses the increasing trend of individuals getting tested for various medical conditions without a prescription. Texas Governor Doug Ducey has signed legislation allowing Arizonans to get any lab test they desire without consulting a doctor first. ...`
Ours$_{\text{raw}}$ `The article discusses the increasing trend of individuals getting tested for various medical conditions without a prescription. Texas Governor Doug Ducey has signed legislation allowing Arizonans to get any lab test they desire without consulting a doctor first. ...` (Detection failed)
Fine-tuning `The article discusses the increasing trend of individuals getting tested for various medical conditions without a prescription. Texas Governor Doug Ducey has signed legislation allowing Arizonans to get any lab test they desire without consulting a doctor first. ...`
Example 2
Key Part of the Target Sentence for summarisation `... Still, the average monthly benefit for retired workers rising by $59 to $1,907 will undoubtedly help retirees with lower and middle incomes to better cope with inflation. ...` ($1907-$59=$1848 increase)
Ground Truth Label `... Retired workers can expect an average monthly benefit of $1,907, up from $1,848. ...`
Ours$_{\text{raw}}$ `... Retired workers can expect an average monthly benefit of $1,907, up from $1,848. ...` (False detection)
Fine-tuning `... Retired workers can expect an average monthly benefit of $1,907, up from $1,848. ...`

Table 13: Hallucination detection example (Summarisation)