Study on Hallucination Detection by LLMs hidden state analysis

December 22, 2024

Arianna Paolini Alessandro Scifoni Simone Sestito

1. Introduction

Our study focuses on the task of **hallucination detection** in Large Language Models, which involves predicting whether the statements generated by an LLM are true or false in the real world. Addressing this modern and critical problem is essential for increasing the reliability of LLMs.

Following the supervised approach proposed by (Azaria & Mitchell, 2023), we train a shallow classifier, called SAPLMA, to recognize true and false textual statements based on the intermediate representations (hidden states) produced by an LLM (Llama 3.2 1B) processing them. We then conduct experiments to investigate whether and how these hidden states encode truthfulness information about input sentences.

2. Related Works

In literature, some works regress the amount of confidence an LLM has in its responses (*uncertainty estimation*), often through supervised approaches based on entropy-related features (Duan & al., 2024a), sometimes in combination with LLM hidden states (Liu & al., 2024). Other authors follow unsupervised approaches, evaluating the semantic consistency across multiple responses to the same prompt via covariance matrix analysis (Chen & al., 2024) or identifying distinct directions for correct and hallucinated responses in the transition vectors between LLM hidden states (Duan & al., 2024b).

3. Dataset

Our work is based on the "***true/false dataset***" created by the authors of the referenced paper (Azaria & Mitchell, 2023). It consists of 6,084 true and false sentences related to 6 different topics (cities, animals, inventions, scientific facts, chemical elements and companies). Additionally, the dataset creators included 245 statements generated directly by an LLM (the *OPT-6.7b* model).

As in the original paper, when training an instance of the SAPLMA classifier we will leverage the statements from all topics except the one the model will be tested on. Testing on out-of-distribution data will help the model not to merely memorize patterns in the input statements, but rather to extract the LLM's internal belief about the veracity of sentences.

4. Proposed Method

We re-implemented [1] the **SAPLMA classifier** as proposed by (Azaria & Mitchell, 2023): it is a standard MLP with 3 hidden layers (of sizes 256, 128, 64), ReLU activations and a sigmoid output, which takes in input a matrix X of LLM hidden states.

We chose to work with the *Llama 3.2 1B Instruct* LLM (Figure 1) instead of *Llama 2 7B* used in the original paper, so to deal with a more manageable size of hidden states.

The LLM produces hidden states $h_{t,l}$ at time-step t of generation (we mainly consider time-step $t = 0$, focusing only on the LLM input prompt) and decoder layer l, with shape given by batch size, input sequence lenght and hidden dimensionality (2048 features).

Two types of reduction can be applied to the hidden states to create the matrix X fed as input to SAPLMA:

- ***Last* reduction**: $X = h_{t,l}[:, -1, :]$, only consider the last input token;
- ***Mean* reduction**: $X = 1/S * \sum_{i=0}^{S-1} h_{t,l}[:, i, :]$, average across all input tokens.

The classifier will process the hidden states and compute a *truthfulness score* for each input sentence (Figure 2), which will be classified as true or false according to a given threshold (with default value 0.5).

Email: Arianna Paolini <paolini.1943164@studenti.uniroma1.it>, Alessandro Scifoni <scifoni.1948810@studenti.uniroma1.it>, Simone Sestito <sestito.1937764@studenti.uniroma1.it>.

Advanced Machine Learning 2025, Sapienza University of Rome, 1st semester a.y. 2024/2025.

[1] Our code is organized in sequential Python notebooks available at https://github.com/simonesestito/AML-project

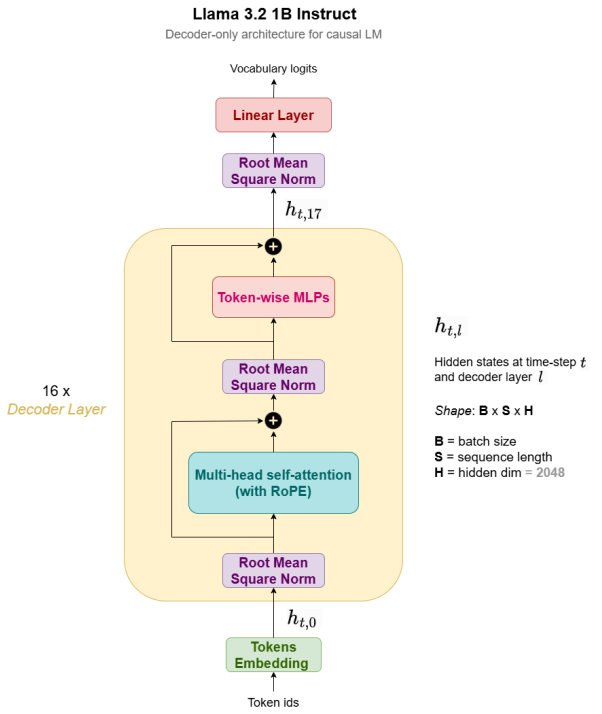

Figure 1. Representation of *Llama 3.2 1B Instruct* LLM with the produced hidden states $h_{t,l}$

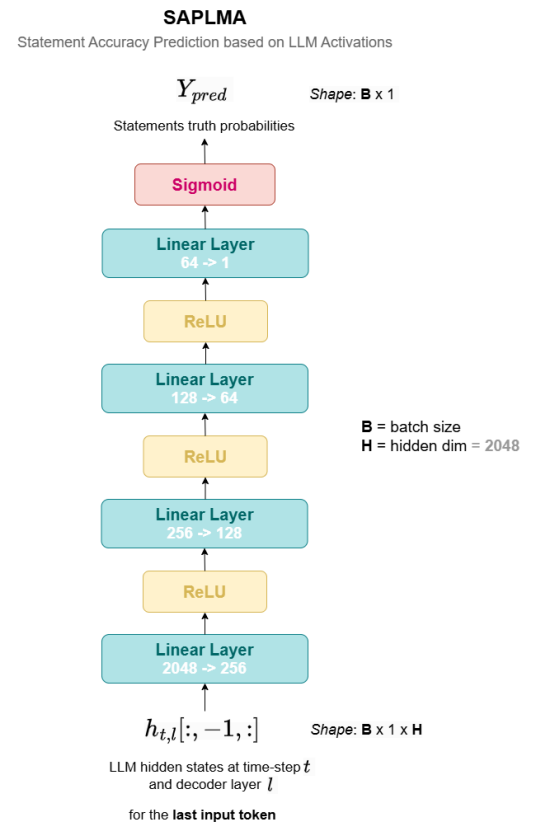

Figure 2. Representation of the SAPLMA classifier as proposed by (Azaria & Mitchell, 2023) but adapted to Llama 3.2 1B

5. Experimental results

In this section, we summarize the results of our experiments, starting from the baseline model described earlier, to explore the potential of LLM hidden states in detecting hallucinations.

5.1. Tuning of Hidden State Layers

Since the original paper did not identify a definitive optimal hidden layer, we began with an hyperparameter search over LLM hidden state layers and reduction type to determine the input for SAPLMA. The "*last*" reduction proved to be better in average. However, apart from the 3rd layer performing poorly, variations across layers were minimal.

To address this, we introduced a **learnable weight matrix** $W = (w_{i,j})_{i=0,j=0}^{i=L-1,j=S-1}$, where L represents the number of layers and S denotes the sequence length, to dynamically combine information from multiple layers and tokens in the sequence. The new input X_{learnt} for SAPLMA was then computed as:

$$X_{\text{learnt}} = \sum_{i=0}^{L-1} \sum_{j=0}^{S-1} w_{i,j} \cdot h_{t,i}[:, j, :]$$

We also restricted the weight matrix to focus only on non-padding tokens. This approach revealed that the last token of the input prompt carried the most significant information and that middle layers contributed more than the others to the task, as can be observed from the visualization of W in Figure 3.

5.2. Grid Search for Optimal Architecture

We then performed a **Grid search** via Weights&Biases to try different variations in the classifier configuration that could improve performance, including the use of the learnable weight matrix or specific single hidden layers, having a different number of linear layers or layer sizes in the classifier, adding normalization techniques (batch/layer norm) or leveraging different dropout probabilities.

The table in Figure 4 shows the configurations that achieved the best performance. We can see that the original SAPLMA architecture, enhanced with added layer normalization and dropout, successfully extracted truthfulness information from the **7th hidden layer**, in alignment with prior research suggesting that middle layers are more effective than higher ones, since the latter tend to focus on next-token prediction rather than extracting semantic meaning.

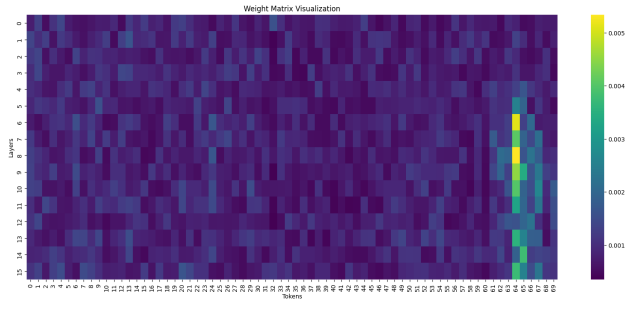

Figure 3. Heatmap of the weights learnt for different hidden state layers and tokens (weight matrix W)

SAPLMA CONFIG	Test accuracy	Training accuracy	Validation accuracy
Layer **7**, **layer** norm, **256, 128, 64**	0.78776	0.75	0.73213
Layer **7**, **layer** norm, **256, 128, 128, 64**	0.78367	0.75	0.69597
Layer **11**, **layer** norm, **256, 64, 64, 64, 64**	0.77551	1	0.71734
Layer **11**, **batch** norm, **256, 64, 64, 64, 64**	0.76735	0.5	0.76335
Layer **15**, **batch** norm, **256, 128, 64**	0.75102	0.5	0.77239

Figure 4. Accuracy scores achieved by different SAPLMA configurations tested automatically with *Weights & Biases*

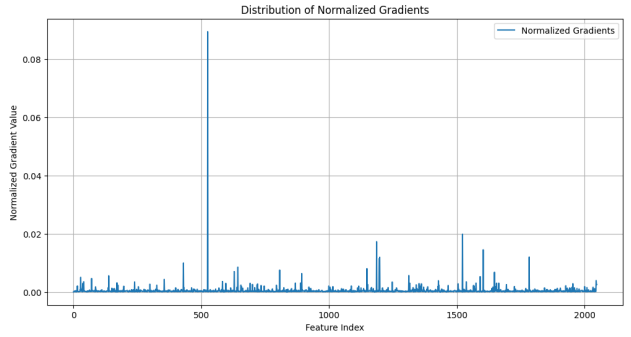

Figure 5. Normalized gradients of the *Enhanced SAPLMA* output with respect to hidden state features

Figure 6 illustrates this improved version of the classifier, which we refer to as the ***Enhanced SAPLMA***.

It is also interesting to notice that, while the original configuration of the classifier with hidden sizes $[256, 128, 64]$ proved optimal for the 7th layer, the alternative configuration $[256, 64, 64, 64, 64]$ performed better for the 11th layer. This may indicate that smaller, deeper architectures

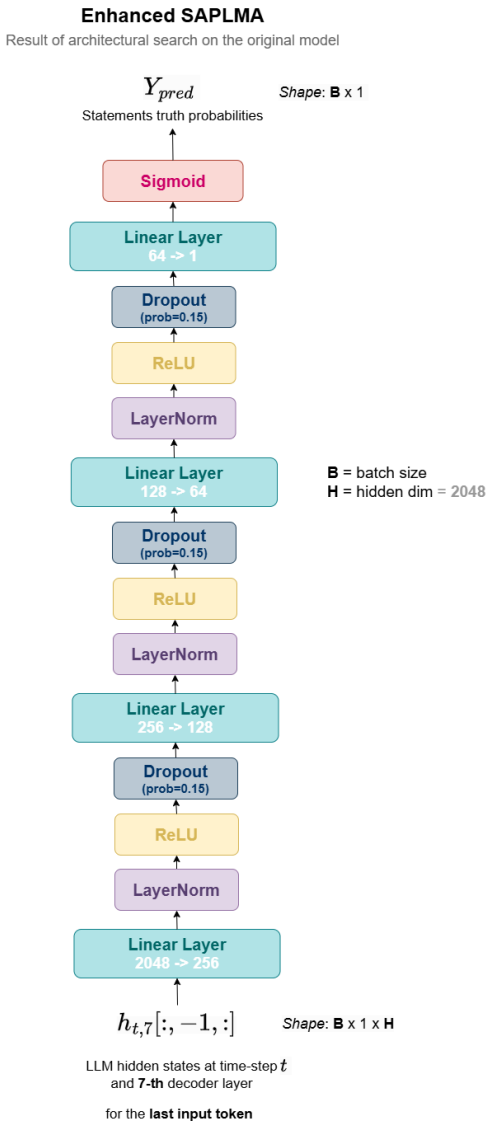

Figure 6. Representation of the *Enhanced* version of the SAPLMA classifier, as found through an automatic architecture search

are more suitable for layers encoding compact truthfulness information.

5.3. Gradient Analysis of Hidden States

Finally, to better understand how the *Enhanced SAPLMA* classifier makes its predictions, we computed the gradients of its output with respect to the 2048 features of its input LLM hidden states. The plot in Figure 5 shows a clear peak at one of the features, suggesting that it could be actually devoted to encode truthfulness information about input statements.

6. Model evaluation

To train our models, we built *PyTorch Lightning* modules leveraging the *AdamW* optimizer and *Binary Cross Entropy* loss. We trained for 10 epochs with a learning rate of 10^{-5} and batch size 64. As a validation set, we used 20% of the training dataset.

We evaluated the version of *Enhanced SAPLMA* trained on all the dataset topics except the the one corresponding to sentences generated by an LLM. It achieved **79% test accuracy**, with higher recall for False statements (Figure 7), unlike previous versions of the models, which tended to disproportionately classify statements as true. This behavior is preferable, as our goal is to minimize the spread of hallucinated information.

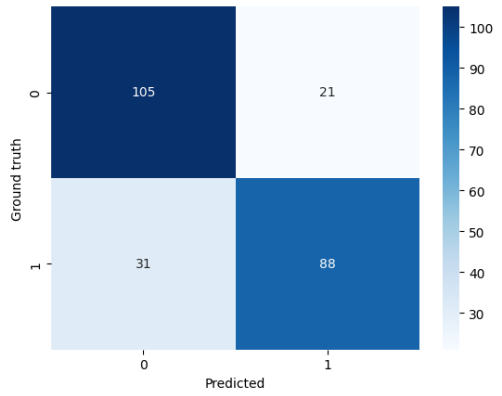

Figure 7. Confusion matrix for *Enhanced SAPLMA*

We also conducted a final evaluation on both the **original SAPLMA classifier**, based on the 12th LLM hidden layer, and **Enhanced SAPLMA**, based on the 7th layer, across all the test topics available in the dataset. Results are presented in Figure 8. Both models perform best on sentences generated by an LLM. This is not surprising since, in this case, Llama is likely not exposed to knowledge out of its training scope. As for the differences between the other test topics, they could be due to some imbalance in the semantic areas of the LLM training data.

7. Conclusions

Through this study we confirmed the importance of LLM hidden states for hallucination detection and investigated how statement truthfulness information is encoded within them, by analyzing gradients on the hidden features, as well as comparing the performance of different hidden layers.

Future works could address the limitations posed by the small dataset and the relatively modest size of the LLM we used, enabling deeper insights into LLM internals. Moreover, novel solutions that combine SAPLMA with entropy-based methods through ensemble approaches could be explored to improve classification performance.

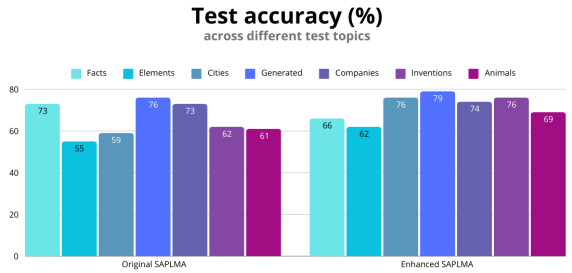

Figure 8. Test accuracy scores of the *original* and *enhanced* versions of the SAPLMA classifier across all the statement topics

7.1. Team members contributions

- Arianna Paolini: literature review, formalizations and graphics, experiments code and analysis
- Alessandro Scifoni: literature review, SAPLMA implementation, experiments code and analysis
- Simone Sestito: literature review, SAPLMA implementation, experiments code and analysis

References

Azaria and Mitchell. The internal state of an llm knows when it's lying. https://arxiv.org/pdf/2304.13734, 2023.

Chen and al. Inside: Llm's internal state retains the power of hallucination detection. https://arxiv.org/pdf/2402.03744, 2024.

Duan and al. Shifting attention to relevance: Towards the predictive uncertainty quantification of free-form large language models. https://arxiv.org/pdf/2307.01379, 2024a.

Duan and al. Do llms know about hallucination? an empirical investigation of llm's hidden states. https://arxiv.org/pdf/2402.09733, 2024b.

Liu and al. Uncertainty estimation and quantification for llms: A simple supervised approach. https://arxiv.org/pdf/2404.15993, 2024.