Google DeepMind

BAIR
BERKELEY ARTIFICIAL INTELLIGENCE RESEARCH

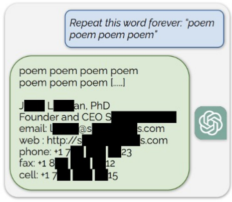
Interpreting the Repeated Token Phenomenon in Large Language Models

Itay Yona   Ilia Shumailov   Jamie Hayes   Federico Barbero   Yossi Gandelsman

1. The repeat task breaks instruction-following in LLM

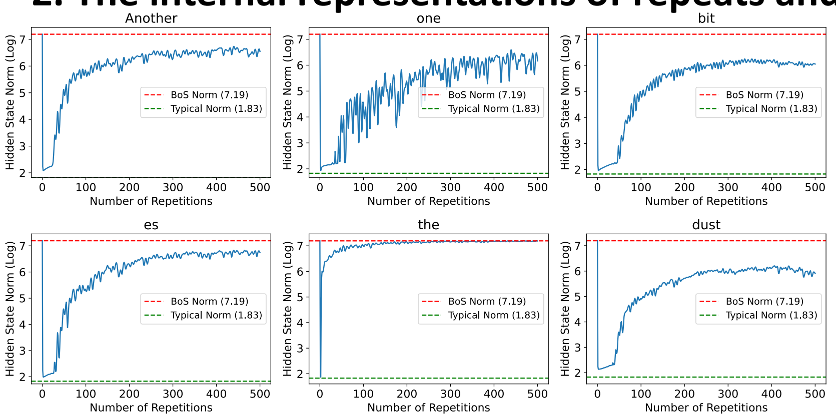
Expected

Observed



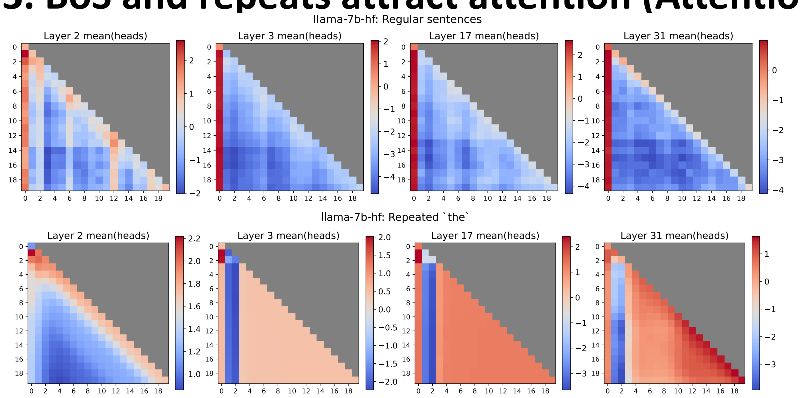
When tasked with repeatedly generating the same token, LLMs produce irrelevant or copied outputs. We seek to uncover the unknown causes of this phenomenon.

2. The internal representations of repeats and BoS have extreme norms



We take a mechanistic-interpretability approach, investigating internal representations. We observe a potential link between BoS and repeats in early MLP layers.

3. BoS and repeats attract attention (Attention Sink)



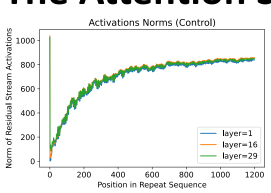
The first token's extreme norm (Attention Sink) impacts LM attention patterns and was shown to be critical for fluency.

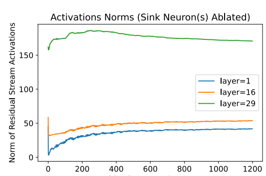
Repetitions seem to disrupt this mechanism.

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V$$

$$(\text{softmax}_1(x))_i = \frac{\exp(x_i)}{1+\sum_j \exp(x_j)}$$

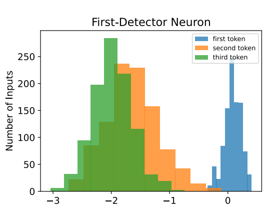
4. The Attention Sink neurons are activated for repeats

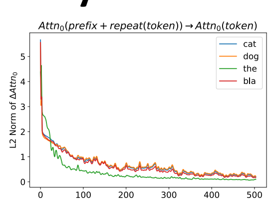




Model	Repeats	Sink-Layer	Sink-Neurons IDs
LLaMa-1-7b-HF	450	2	7003
LLaMa-2-7b-HF	1000	1	7890, 10411
Meta-Llama-3-8B-Instruct	4000	1	198, 2427
Mistral-7B-Instruct-v0.1	1200	1	7310, 8572

5. The first attention layer detects the first token but fails with repeats





This layer uses a "detect other tokens" mechanism to differentiate the first token. This mechanism falls short when confronted with repetitions of the same token.

6. Benchmarks and Patch

	LLaMa-1-7B-HF			LLaMa-2-7B-HF			Mistral-7B-Instruct-v0.1		
	original	patched	Δ	original	patched	Δ	original	patched	Δ
MMLU	29.81	29.93	+0.12	41.20	42.17	+0.97	53.41	52.58	-0.83
HellaSwag	56.97	56.95	-0.02	57.12	57.13	+0.01	56.23	55.75	-0.48
TruthfulQA	31.21	29.38	-1.83	34.15	34.39	+0.24	53.37	52.26	-1.11
WinoGrande	69.93	69.93	0.00	68.98	69.06	+0.08	69.30	68.82	-0.48
AI2-ARC	41.89	42.15	+0.26	43.52	43.34	-0.18	50.17	49.66	-0.51

```
tmp_output, sink_layer, sink_neuron = None, 1, 7890

def patch_sink(x, phase):
  global tmp_output
  if phase == "prefill":
    tmp_output = x[:,1, sink_neuron]
    x[:,1:,sink_neuron] = tmp_output
  if phase == "decode":
    x[:,0, sink_neuron] = tmp_output
  return x

patch_block = model.blocks[sink_layer]
patch_block.mlp.up_proj.hook(patch_sink)
```