How do Large Language Models Handle Multilingualism?

Yiran Zhao[1,2†] **Wenxuan Zhang**[2,3‡] **Guizhen Chen**[2,4§] **Kenji Kawaguchi**[1] **Lidong Bing**[2,3]
[1] National University of Singapore [2] DAMO Academy, Alibaba Group, Singapore
[3] Hupan Lab, 310023, Hangzhou, China [4] Nanyang Technological University, Singapore
zhaoyiran@u.nus.edu kenji@comp.nus.edu.sg
{saike.zwx, guizhen.chen, l.bing}@alibaba-inc.com

How do LLMs handle multilingualism?

- Existing LLMs exhibit certain multilingual abilities (at least for some languages)
- But a fundamental question: ***how do LLMs handle multilingualism?***
- We won't be able to (efficiently) enhance the multilingual ability without having answers (or even certain clues) to this question

How do LLMs handle multilingualism?

- Existing LLMs exhibit certain multilingual abilities (at least for some languages)
- But a fundamental question: *how do LLMs handle multilingualism?*
- We won't be able to (efficiently) enhance the multilingual ability without having answers (or even certain clues) to this question

- **What do we know for now**
  - (Traditional) multilingual research: mainly works on understanding the cross-lingual transfer ability
    - train on English labeled data, perform tasks in other languages
  - More recent explainability-style studies
    - *We show that feed-forward layers emulate neural memories, where the first parameter matrix in the layer corresponds to keys, and the second parameter matrix to values.* [1]
    - ... *indicate that LMs process the input by transmitting the information relevant to the query from mid-sequence early layers to the final token using the attention mechanism* [2]

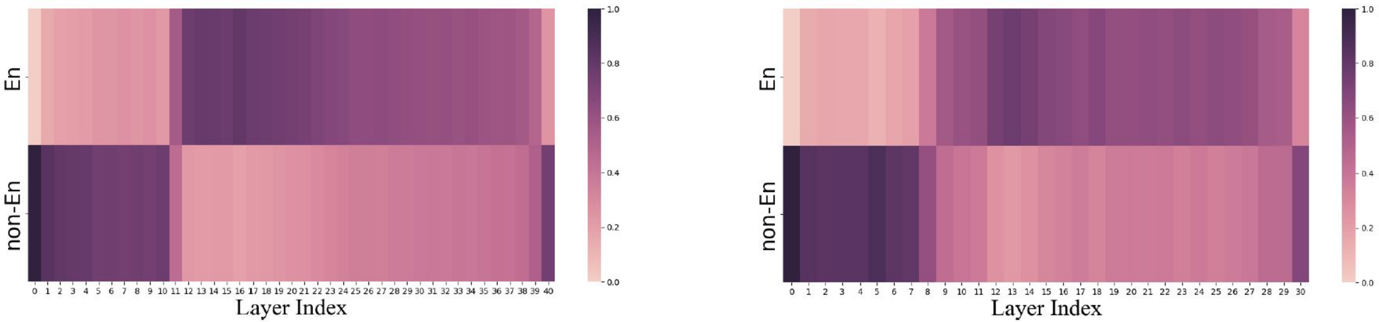
[1] Mor Geva, Roei Schuster, Jonathan Berant, Omer Levy. Transformer Feed-Forward Layers Are Key-Value Memories. EMNLP 2021
[2] Alessandro Stolfo, Yonatan Belinkov, Mrinmaya Sachan. A Mechanistic Interpretation of Arithmetic Reasoning in Language Models using Causal Mediation Analysis. EMNLP 2023

Investigating the embeddings first

- To gain an initial understanding, we analyze the decoded embeddings after each layer when processing inputs in various **non-English languages**.
- We then classify these embeddings as corresponding to either English or non-English tokens

Investigating the embeddings first

- To gain an initial understanding, we analyze the decoded embeddings after each layer when processing inputs in various **non-English languages**.
- We then classify these embeddings as corresponding to either English or non-English tokens



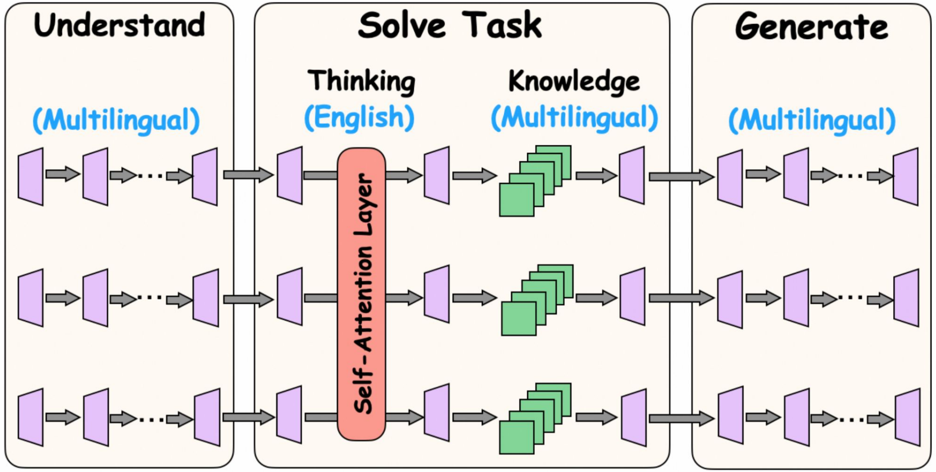
(a) Vicuna-13b-v1.5

(b) BLOOMZ-7b1

Figure 1: Ratio of English and non-English tokens among layers given non-English queries.

Non-English => English => Non-English

Put all together: A new framework



- ❑ In the first several layers, LLMs **understand** the user input and convert the diverse linguistic features into a unified representation.
- ❑ Transitioning to the **task-solving** phase, LLMs solve the tasks by thinking in English and incorporating multilingual knowledge, leveraging the self-attention and feed-forward structures respectively.
- ❑ Finally, models **generate** responses that align with the original language of the query.

Detect language-specific neuron

input. We denote the input of i-th layer in Transformer (Vaswani et al., 2017) as h_i, with the corresponding output represented as $h_{i+1} = T_i(h_i)$, where T_i represents the parameters of the i-th layer. For a specific neuron, denoted as $N_k^{(i)}$, within the i-th layer—whether located in the attention or feed-forward layer—the importance is quantified as the difference between output when $N_k^{(i)}$ is either activated or deactivated. Formally, it is defined as

$$\mathrm{Imp}(N_k^{(i)}|h_i) = \|T_i \backslash N_k^{(i)}(h_i) - T_i(h_i)\|_2, \quad (1)$$

where $T_i \backslash N_k(\cdot)$ denotes deactivating $N_k^{(i)}$ in T_i. Then, with a set of the corpus in the specific language, denoted as $\mathcal{C} = \{c_1, \cdots, c_l, \cdots, c_n\}$, we can calculate the importance of each neuron in each layer to each corpus. Furthermore, we can select neurons that are important to all corpus in $\mathcal{C}$, i.e.,

$$\mathrm{Imp}(N_k^{(i)}|c_l) \geq \epsilon, \ \forall c_l \in \mathcal{C}, \quad (2)$$

where ϵ is the pre-defined threshold. However, it is super time-consuming to traverse all neurons and all inputs sequentially. Therefore, we need to design a parallel algorithm for acceleration.

2.2 Parallel Neuron Detection

Feed-Forward Layer In Llama2 (Touvron et al., 2023), the $\mathrm{FFN}(x)$ is defined as

$$\Big(\mathrm{SiLU}\big(W_{gate}(x)\big) \cdot W_{up}(x)\Big) W_{down}, \quad (3)$$

where $x \in \mathbb{R}^{l \times d_{model}}$, $W_{gate} \in \mathbb{R}^{d_{model} \times d_{inter}}$, $W_{down} \in \mathbb{R}^{d_{inter} \times d_{model}}$. We denote hidden embedding before W_{down} as h_{ffn}. When deactivating the k-th neuron of W_{up},

$$\begin{aligned} &\mathrm{Imp}(W_{up}[:,k]|x) = \|\hat{\mathrm{FFN}}(x) - \mathrm{FFN}(x)\|_2 \\ &= \Big\|\big(h_{\text{ffn}} \cdot \texttt{Mask}[k]\big) W_{down}(x)\Big\|_2, \end{aligned} \quad (4)$$

where $\texttt{Mask}[k]$ is a vector of length d_{inter} with the k-th element as 1 and others as 0. For calculating $\mathrm{Imp}(W_{up}[:,k]|x)$ for all neurons in W_{up} parallelly, we introduce a diagonal mask matrix of size (d_{inter}, d_{inter}), denoted as $\texttt{Mask}$. Therefore,

$$\mathrm{Imp}(W_{up}|x) = \|(h_{\text{ffn}} \cdot \texttt{Mask}) W_{down}(x)\|_2. \quad (5)$$

Furthermore, we find that deactivating the k-th neuron of W_{down} is equivalent to deactivating the k-th neuron in W_{up} as they all set $h_{\text{ffn}}[k] = 0$. Therefore $\mathrm{Imp}(W_{down}|x)$ can be obtain by Equation (5).

Self-Attention Layer For the input x of length l, the self-attention layer is defined as

$$\mathrm{Softmax}\Big(\frac{W_Q(x) W_K^T(x)}{\sqrt{d}}\Big) W_V(x), \quad (6)$$

where $W_Q \in \mathbb{R}^{d_{model} \times d_{mid}}$, $W_K \in \mathbb{R}^{d_{model} \times d_{mid}}$, $W_V \in \mathbb{R}^{d_{model} \times d_{mid}}$.[2] As $W_V(x)$ is a linear layer, $\mathrm{Imp}(W_V|x)$ can be obtained following Equation (5). In the case of W_Q, when deactivating the k-th neuron, $\hat{W}_Q \leftarrow W_Q[:,k] = 0$, we aim to obtain $\mathrm{Imp}(W_Q[:,k]|x)$. Firstly, we calculate the difference in attention weight, i.e., $W_Q(x) W_K^T(x)$.

$$\begin{aligned} \Delta_k &= \hat{W}_Q(x) W_K^T(x) - W_Q(x) W_K^T(x) \\ &= W_Q(x)[:,k] W_K(x)[k,:] \in \mathbb{R}^{l \times l} \end{aligned} \quad (7)$$

Then, the importance of $W_Q[:,k]$ can be defined as

$$\begin{aligned} &\mathrm{Imp}(W_Q[k,:]|x) \\ &\approx \|\hat{\mathrm{attention}}(x) - \mathrm{attention}(x)\|_2 \\ &\approx \Big\|\mathrm{softmax}\Big(\frac{W_Q(x) W_K^T(x) - \Delta_k}{\sqrt{d}}\Big) - \\ &\quad \mathrm{softmax}\Big(\frac{W_Q(x) W_K^T(x)}{\sqrt{d}}\Big)\Big\|_2 \end{aligned} \quad (8)$$

This process can also be calculated parallelly, i.e.,

$$\begin{aligned} \Delta &= \hat{W}_Q(x) W_K^T(x) - W_Q(x) W_K^T(x) \\ &= W_Q(x).resize(l, 1, d_{mid}) \times \\ &\quad W_K(x).resize(1, l, d_{mid}) \in \mathbb{R}^{l \times l \times d_{mid}} \end{aligned} \quad (9)$$

Then, the importance of W_Q can be defined as

$$\begin{aligned} \mathrm{Imp}(W_Q|x) \approx \Big\|&\mathrm{softmax}\Big(\frac{W_Q(x) W_K^T(x) - \Delta}{\sqrt{d}}\Big) - \\ &\mathrm{softmax}\Big(\frac{W_Q(x) W_K^T(x)}{\sqrt{d}}\Big)\Big\|_2. \end{aligned}$$

$\mathrm{Imp}(W_K|x)$ can be calculated the same way.

3 Investigate Language-Specific Neurons

In this section, we apply the $\texttt{PLND}$ method to selected languages and models in order to confirm the existence of language-specific neurons and investigate the relationships between languages.

[2] In Vicuna and Mistral, $d_{model} = d_{mid}$, but we use different notations to avoid ambiguity.

- How to validate such a framework: deactivate **relevant neurons**
- We propose a method to detect **language-specific neuron** with pure free text (aka unlabeled data) of certain languages

	Method	Fr	Zh	Es	Ru	Avg.
Vicuna	Original	14.2	61.1	10.4	20.8	26.6
	Deact-Rand.	14.1	61.6	10.4	20.8	26.7
	Deact-Lang.	**0.83**	**0.00**	**0.24**	**0.42**	**0.37**
Mistral	Original	15.2	56.4	10.6	21.0	25.8
	Deact-Rand.	15.4	55.9	10.2	21.2	25.7
	Deact-Lang.	**0.21**	**0.39**	**0.15**	**0.07**	**0.21**

- Just deactivating around **0.13%** neurons, LLMs almost lose multilingual capabilities (26.6 => 0.37)

Verify the framework

Approach: deactivate certain language-specific neurons of certain structures and observe the performance gap for English and Non-English tasks

- comparisons: language-specific neurons v.s. random neurons
- metrics:
  - The gap between the original performance and performance after deactivation for English (ΔEng) and averaged non-English languages (Δn-Eng)
  - A single metric **Δ = ΔEng − Δn-Eng**, where a high value indicates such deactivation operation does not bring much impact to the English performance but lead to performance drop in non-English.

Verify the framework - Understanding

Model	Deactivating Method					Performance				
	Under	S-ATTN	S-FFN	Gen	Neuron	Eng	n-Eng	Δ_{Eng}	$\Delta_{\text{n-Eng}}$	$\Delta\uparrow$
Vicuna	✓	✗	✗	✗	Random	57.8	53.9	+0.3	−0.1	+0.4
	✓	✓	✓	✓	Random	57.9	54.2	+0.4	+0.3	+0.1
	✓	✗	✗	✗	Lang-Spec	56.5	46.0	−0.5	−7.9	+7.4
	✗	✓	✓	✗	Lang-Spec	40.9	38.6	−15.9	−15.3	−0.6
	✗	✗	✗	✓	Lang-Spec	57.9	52.8	−0.4	−1.1	+0.7
Mistral	✓	✗	✗	✗	Random	58.1	55.5	+1.0	−0.2	+1.2
	✓	✓	✓	✓	Random	57.6	55.5	+0.5	−0.2	+0.7
	✓	✗	✗	✗	Lang-Spec	56.2	48.3	−0.9	−7.4	+6.5
	✗	✓	✓	✗	Lang-Spec	53.2	47.0	−3.9	−8.7	+4.8
	✗	✗	✗	✓	Lang-Spec	56.4	54.6	−0.7	−1.0	+0.3

randomly deactivating neurons (wherever they are) => almost unaffected

(i) neurons randomly selected from the understanding layers
(ii) neurons randomly chosen across all layers
(iii) language-specific neurons within the understanding layers
(iv) language-specific neurons in the task-solving layers
(v) language-specific neurons in the generation layers.

Verify the framework - Understanding

Model	Under	S-ATTN	S-FFN	Gen	Neuron	Eng	n-Eng	Δ_{Eng}	$\Delta_{\text{n-Eng}}$	$\Delta\uparrow$
	Deactivating Method					Performance				
Vicuna	✓	✗	✗	✗	Random	57.8	53.9	+0.3	−0.1	+0.4
	✓	✓	✓	✓	Random	57.9	54.2	+0.4	+0.3	+0.1
	✓	✗	✗	✗	Lang-Spec	56.5	46.0	−0.5	−7.9	+7.4
	✗	✓	✓	✗	Lang-Spec	40.9	38.6	−15.9	−15.3	−0.6
	✗	✗	✗	✓	Lang-Spec	57.9	52.8	−0.4	−1.1	+0.7
Mistral	✓	✗	✗	✗	Random	58.1	55.5	+1.0	−0.2	+1.2
	✓	✓	✓	✓	Random	57.6	55.5	+0.5	−0.2	+0.7
	✓	✗	✗	✗	Lang-Spec	56.2	48.3	−0.9	−7.4	+6.5
	✗	✓	✓	✗	Lang-Spec	53.2	47.0	−3.9	−8.7	+4.8
	✗	✗	✗	✓	Lang-Spec	56.4	54.6	−0.7	−1.0	+0.3

all performance drop

almost unaffected

(i) neurons randomly selected from the understanding layers

(ii) neurons randomly chosen across all layers

(iii) language-specific neurons within the understanding layers

(iv) language-specific neurons in the task-solving layers

(v) language-specific neurons in the generation layers.

Verify the framework - Understanding

Model	Under	S-ATTN	S-FFN	Gen	Neuron	Eng	n-Eng	Δ_{Eng}	$\Delta_{\text{n-Eng}}$	$\Delta\uparrow$
	Deactivating Method					**Performance**				
Vicuna	✓	✗	✗	✗	Random	57.8	53.9	+0.3	−0.1	+0.4
	✓	✓	✓	✓	Random	57.9	54.2	+0.4	+0.3	+0.1
	✓	✗	✗	✗	Lang-Spec	56.5	46.0	−0.5	−7.9	+7.4
	✗	✓	✓	✗	Lang-Spec	40.9	38.6	−15.9	−15.3	−0.6
	✗	✗	✗	✓	Lang-Spec	57.9	52.8	−0.4	−1.1	+0.7
Mistral	✓	✗	✗	✗	Random	58.1	55.5	+1.0	−0.2	+1.2
	✓	✓	✓	✓	Random	57.6	55.5	+0.5	−0.2	+0.7
	✓	✗	✗	✗	Lang-Spec	56.2	48.3	−0.9	−7.4	+6.5
	✗	✓	✓	✗	Lang-Spec	53.2	47.0	−3.9	−8.7	+4.8
	✗	✗	✗	✓	Lang-Spec	56.4	54.6	−0.7	−1.0	+0.3

English unaffected, but target languages are greatly impacted

✅ prove our 1st hypothesis

(i) neurons randomly selected from the understanding layers

(ii) neurons randomly chosen across all layers

(iii) language-specific neurons within the understanding layers

(iv) language-specific neurons in the task-solving layers

(v) language-specific neurons in the generation layers.

Verify the framework - Reasoning

Model	Deactivating Method					Performance				
	Under	S-ATTN	S-FFN	Gen	Neuron	Eng	n-Eng	Δ_{Eng}	$\Delta_{\text{n-Eng}}$	$\Delta \uparrow$
Vicuna	✗	✓	✗	✗	Random	20.0	11.3	−0.4	−1.8	+1.4
	✗	✓	✓	✗	Random	18.4	12.2	−2.0	−1.0	−1.0
	✓	✓	✓	✓	Random	19.6	12.5	−0.8	−0.7	−0.1
	✗	✓	✓	✗	Lang-Spec	7.2	3.4	−13.2	−9.8	−3.4
	✓	✗	✗	✓	Lang-Spec	18.1	8.3	−2.3	−4.9	+2.6
	✓	✗	✓	✓	Lang-Spec	19.0	7.8	−1.4	−5.4	+4.0
Mistral	✗	✓	✗	✗	Random	40.8	23.4	−5.2	−2.9	−2.3
	✗	✓	✓	✗	Random	39.2	24.0	−6.8	−2.3	−4.5
	✓	✓	✓	✓	Random	45.2	26.8	−0.8	+0.5	−1.3
	✗	✓	✓	✗	Lang-Spec	38.2	18.4	−7.8	−7.9	+0.1
	✓	✗	✗	✓	Lang-Spec	44.0	18.1	−2.0	−8.2	+6.2
	✓	✗	✓	✓	Lang-Spec	46.2	18.3	+0.2	−8.0	+8.2

randomly deactivating neurons in task-specific layer matters most

Verify the framework - Reasoning

Model	Deactivating Method					Performance				
	Under	S-ATTN	S-FFN	Gen	Neuron	Eng	n-Eng	Δ_{Eng}	$\Delta_{\text{n-Eng}}$	$\Delta\uparrow$
Vicuna	✗	✓	✗	✗	Random	20.0	11.3	−0.4	−1.8	+1.4
	✗	✓	✓	✗	Random	18.4	12.2	−2.0	−1.0	−1.0
	✓	✓	✓	✓	Random	19.6	12.5	−0.8	−0.7	−0.1
	✗	✓	✓	✗	Lang-Spec	7.2	3.4	−13.2	−9.8	−3.4
	✓	✗	✗	✓	Lang-Spec	18.1	8.3	−2.3	−4.9	+2.6
	✓	✗	✓	✓	Lang-Spec	19.0	7.8	−1.4	−5.4	+4.0
Mistral	✗	✓	✗	✗	Random	40.8	23.4	−5.2	−2.9	−2.3
	✗	✓	✓	✗	Random	39.2	24.0	−6.8	−2.3	−4.5
	✓	✓	✓	✓	Random	45.2	26.8	−0.8	+0.5	−1.3
	✗	✓	✓	✗	Lang-Spec	38.2	18.4	−7.8	−7.9	+0.1
	✓	✗	✗	✓	Lang-Spec	44.0	18.1	−2.0	−8.2	+6.2
	✓	✗	✓	✓	Lang-Spec	46.2	18.3	+0.2	−8.0	+8.2

English is also destroyed if deactivating both attention and FFN layers

But it can be preserved if we only deactivate the FFN layers

Verify the framework - Multilingual Knowledge

Model	Deactivating Method					Performance				
	Under	S-ATTN	S-FFN	Gen	Neuron	Eng	n-Eng	Δ_{Eng}	$\Delta_{\text{n-Eng}}$	$\Delta\uparrow$
Vicuna	✗	✗	✓	✗	Random	57.5	39.5	-0.3	$+0.0$	-0.3
	✗	✓	✓	✗	Random	56.0	38.7	-1.8	-0.8	-1.0
	✓	✓	✓	✓	Random	57.7	39.6	-0.1	$+0.1$	-0.2
	✗	✓	✗	✗	Lang-Spec	33.7	30.3	-24.1	-9.2	-14.9
	✗	✗	✓	✗	Lang-Spec	57.5	37.5	-0.3	-2.0	$+1.7$
Mistral	✗	✗	✓	✗	Random	61.0	37.0	-0.3	-0.5	$+0.2$
	✗	✓	✓	✗	Random	60.7	36.3	-0.6	-1.2	$+0.6$
	✓	✓	✓	✓	Random	61.8	37.4	$+0.1$	-0.1	$+0.2$
	✗	✓	✗	✗	Lang-Spec	51.2	28.9	-10.1	-8.6	-1.5
	✗	✗	✓	✗	Lang-Spec	61.2	35.1	-0.1	-2.4	$+2.3$

Table 6: Results of the knowledge question answering. The highest performance reduction difference (Δ) is achieved by disabling all language-specific neurons in the feed-forward structure within the task-solving layer.

Verify the framework - Generation

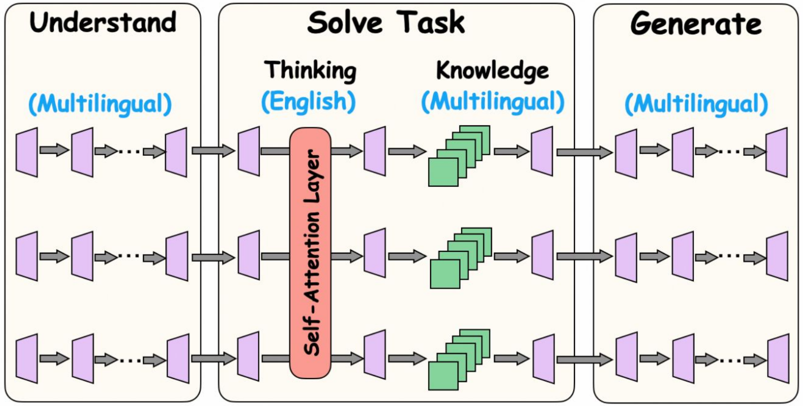
Model	Deactivating Method					Performance				
	Under	S-ATTN	S-FFN	Gen	Neuron	Eng	n-Eng	Δ_{Eng}	$\Delta_{\text{n-Eng}}$	$\Delta \uparrow$
Vicuna	✗	✗	✗	✓	Random	13.2	26.8	+0.1	+0.1	+0.0
	✓	✓	✓	✓	Random	13.0	26.7	−0.1	+0.0	−0.1
	✗	✗	✗	✓	Lang-Spec	13.1	25.7	+0.0	−1.1	+1.1
Mistral	✗	✗	✗	✓	Random	13.6	25.9	+0.1	+0.1	+0.0
	✓	✓	✓	✓	Random	13.6	25.7	+0.1	−0.2	+0.3
	✗	✗	✗	✓	Lang-Spec	13.8	24.3	+0.3	−1.5	+1.8

Table 7: Results of the generation task following neuron deactivation. The highest performance reduction difference (Δ) is achieved by disabling all language-specific neurons in the generation layer.

How can we utilize such a framework: Enhancement!

We have (basically) verified the proposed framework via deactivating certain neurons.

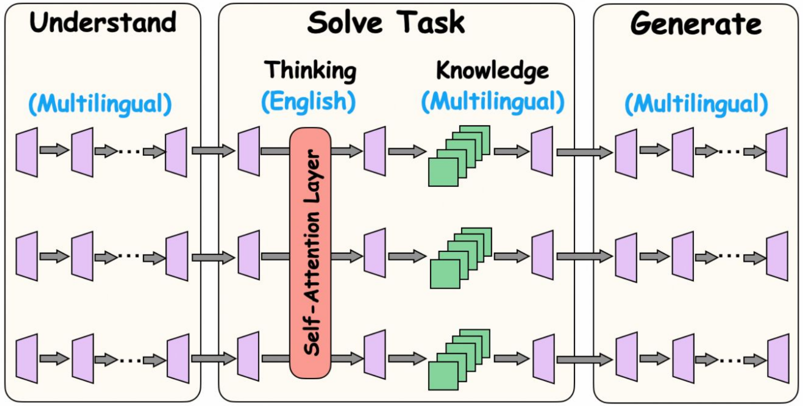
- ❑ We can also enhance their multilingual ability



How can we utilize such a framework: Enhancement!

We have (basically) verified the proposed framework via deactivating certain neurons.

- ❑ We can also enhance their multilingual ability
- ❑ Mainly focus on the understanding and generation ability first, since extending the reasoning abilities or broadening the knowledge base may require more specific data preparation
- ❑ Approach: tune language-specific neuron with only <1k documents!



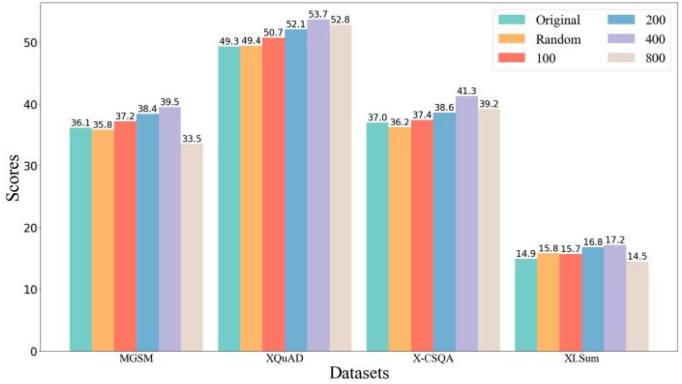


Figure 4: Enhancement results on high-resource languages, while the number is average among languages.