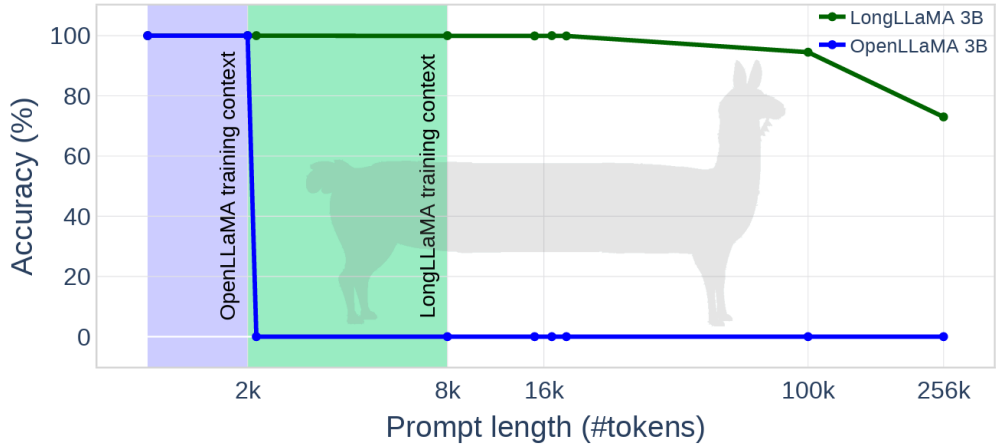
On Long Context Language Modeling

Guande He

2023.11.3

Overview

- Long context modeling is a key ability for LM's application (e.g., processing long documents, coherent multi-round dialogs…).
- Current LMs are pre-trained with context windows less than 10k tokens (LLaMA: 2048; LLaMA-2, ChatGPT: 4096)
- Pre-trained LMs need to be effectively and efficiently adapted for **long context language modeling**.



Picture from "Focused Transformer: Contrastive Training for Context Scaling"

Overview

- Currently, a decent way to deploy LLMs for practical use is Retrieval Augmented Generation (RAG).
  - Instead of tackling the 'internal' problems of LLMs, such as static, lack of domain-specific knowledge, hallucinations, and costly adaptations, we could retrieval grounding external context to LLMs for generation purpose.

- The ability to wield long context is crucial for LLMs!

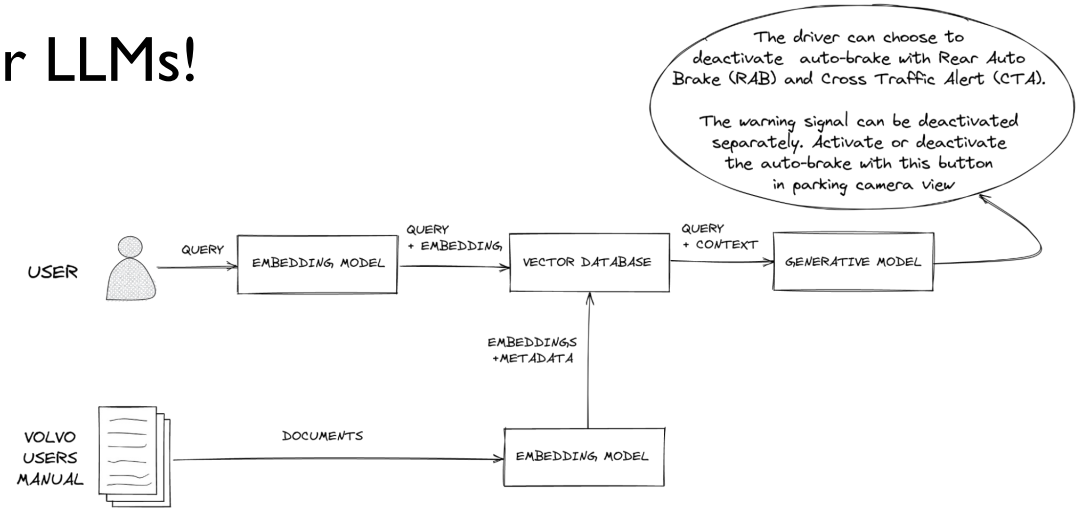


Image source: Retrieval Augmented Generation (RAG): Reducing Hallucinations in GenAI Applications (Pinecone)

Outline

- Model Refinement
  - **Positional encoding**
  - Attention scheme
- Data Mix
- Training
- Application-oriented Evaluation

Sinusoidal Positional Encoding

- The attention module has no intrinsic ability of order modeling, compared with convolution or recurrence.
- We need to inject some information about the relative and absolute position of the tokens in the sequence, namely positional encoding (PE).
- Formally, denote the dim of embeddings as d_{model} , the sinusoidal PE is defined as sine and cosine functions of different frequencies:

$$PE_{(pos,2i)} = \sin(\frac{pos}{10000^{2i/d_{\text{model}}}})$$

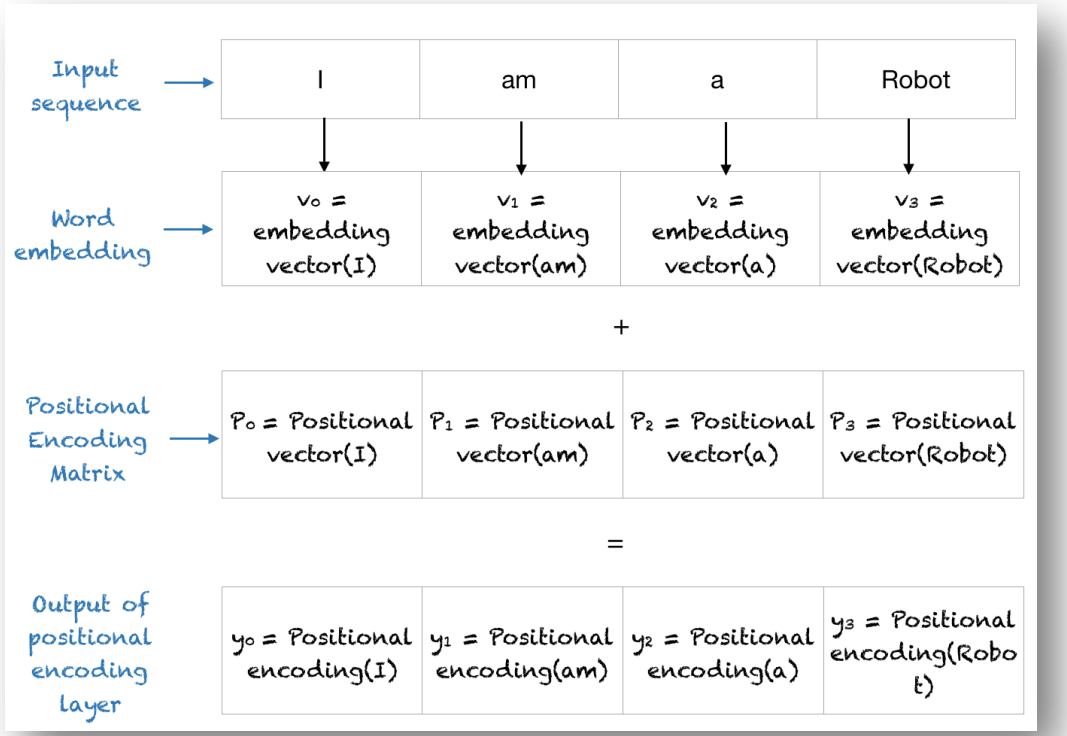
$$PE_{(pos,2i+1)} = \cos(\frac{pos}{10000^{2i/d_{\text{model}}}})$$

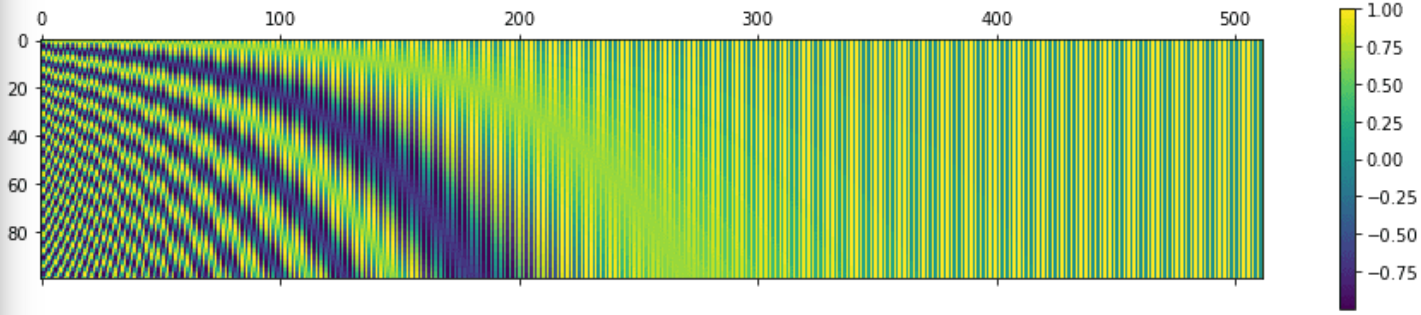
where pos is the position and i is the dimension.

Vaswani, Ashish, et al. "Attention is all you need." (NeurIPS 2017).

Sinusoidal Positional Encoding

- Input embedding = word embedding + positional encoding





Pictures from "A Gentle Introduction to Positional Encoding in Transformer Models"

Sinusoidal Positional Encoding

Some intuition about how it works

- Denote the model as a scalar function $f(\cdots, \boldsymbol{x}_m, \cdots, \boldsymbol{x}_n, \cdots)$, where x_m and x_n are m-th and n-th elements.
- Attention-based models will yield $f(\cdots, \boldsymbol{x}_m, \cdots, \boldsymbol{x}_n, \cdots) = f(\cdots, \boldsymbol{x}_n, \cdots, \boldsymbol{x}_m, \cdots)$
- Positional encoding could break such symmetry by:

$$\tilde{f}(\cdots, \boldsymbol{x}_m, \cdots, \boldsymbol{x}_n, \cdots) = f(\cdots, \boldsymbol{x}_m + \boldsymbol{p}_m, \cdots, \boldsymbol{x}_n + \boldsymbol{p}_n, \cdots)$$

- Question: How can we come up with a reasonable design for $\boldsymbol{p}$?
- Taylor expansion at $(\cdots, \boldsymbol{x}_m, \cdots, \boldsymbol{x}_n, \cdots)$:

$$\tilde{f} \approx f + \boldsymbol{p}_m^\top \frac{\partial f}{\partial \boldsymbol{x}_m} + \boldsymbol{p}_n^\top \frac{\partial f}{\partial \boldsymbol{x}_n} + \frac{1}{2}\boldsymbol{p}_m^\top \frac{\partial^2 f}{\partial \boldsymbol{x}_m^2}\boldsymbol{p}_m + \frac{1}{2}\boldsymbol{p}_n^\top \frac{\partial^2 f}{\partial \boldsymbol{x}_n^2}\boldsymbol{p}_n + \underbrace{\boldsymbol{p}_m^\top \frac{\partial^2 f}{\partial \boldsymbol{x}_m \partial \boldsymbol{x}_n}\boldsymbol{p}_n}_{\boldsymbol{p}_m^\top \boldsymbol{\mathcal{H}} \boldsymbol{p}_n}$$

苏剑林. “Transformer升级之路：1、Sinusoidal位置编码追根溯源 ”(2021).

Sinusoidal Positional Encoding

Some intuition about how it works

$$\tilde{f} \approx f + \boldsymbol{p}_m^\top \frac{\partial f}{\partial \boldsymbol{x}_m} + \boldsymbol{p}_n^\top \frac{\partial f}{\partial \boldsymbol{x}_n} + \frac{1}{2}\boldsymbol{p}_m^\top \frac{\partial^2 f}{\partial \boldsymbol{x}_m^2}\boldsymbol{p}_m + \frac{1}{2}\boldsymbol{p}_n^\top \frac{\partial^2 f}{\partial \boldsymbol{x}_n^2}\boldsymbol{p}_n + \underbrace{\boldsymbol{p}_m^\top \frac{\partial^2 f}{\partial \boldsymbol{x}_m \partial \boldsymbol{x}_n}\boldsymbol{p}_n}_{\boldsymbol{p}_m^\top \boldsymbol{\mathcal{H}} \boldsymbol{p}_n}$$

- In principle, an ideal positional encoding should contain the information of **relative position** between tokens.
- A promising term is $\boldsymbol{p}_m^\top \boldsymbol{\mathcal{H}} \boldsymbol{p}_n$. For simplicity, consider $d_{\text{model}} = 2, \boldsymbol{\mathcal{H}} = \boldsymbol{I}$, we want $\boldsymbol{p}_m^\top \boldsymbol{\mathcal{H}} \boldsymbol{p}_n$ could encode the relative position, i.e., $\langle \boldsymbol{p}_m, \boldsymbol{p}_n \rangle = g(m-n)$.
- Derive with complex numbers:

$$\langle \boldsymbol{p}_m, \boldsymbol{p}_n \rangle = \mathrm{Re}[\boldsymbol{p}_m \boldsymbol{p}_n^*] \quad \Leftarrow \quad \boldsymbol{p}_m \boldsymbol{p}_n^* = \boldsymbol{q}_{m-n}$$

$$\boldsymbol{p}_m = r_m e^{\mathrm{i}\phi_m}, \boldsymbol{p}_n^* = r_n e^{-\mathrm{i}\phi_n}, \boldsymbol{q}_{m-n} = R_{m-n} e^{\mathrm{i}\Phi_{m-n}}$$

$$r_m r_n e^{\mathrm{i}(\phi_m - \phi_n)} = R_{m-n} e^{\mathrm{i}\Phi_{m-n}} \quad \Rightarrow \quad \begin{cases} r_m r_n = R_{m-n} \\ \phi_m - \phi_n = \Phi_{m-n} \end{cases}$$

- One solution

$$\boldsymbol{p}_m = e^{\mathrm{i}m\theta} \quad \Leftrightarrow \quad \boldsymbol{p}_m = \begin{pmatrix} \cos m\theta \\ \sin m\theta \end{pmatrix}$$

苏剑林. “Transformer升级之路：1、Sinusoidal位置编码追根溯源 ”(2021).

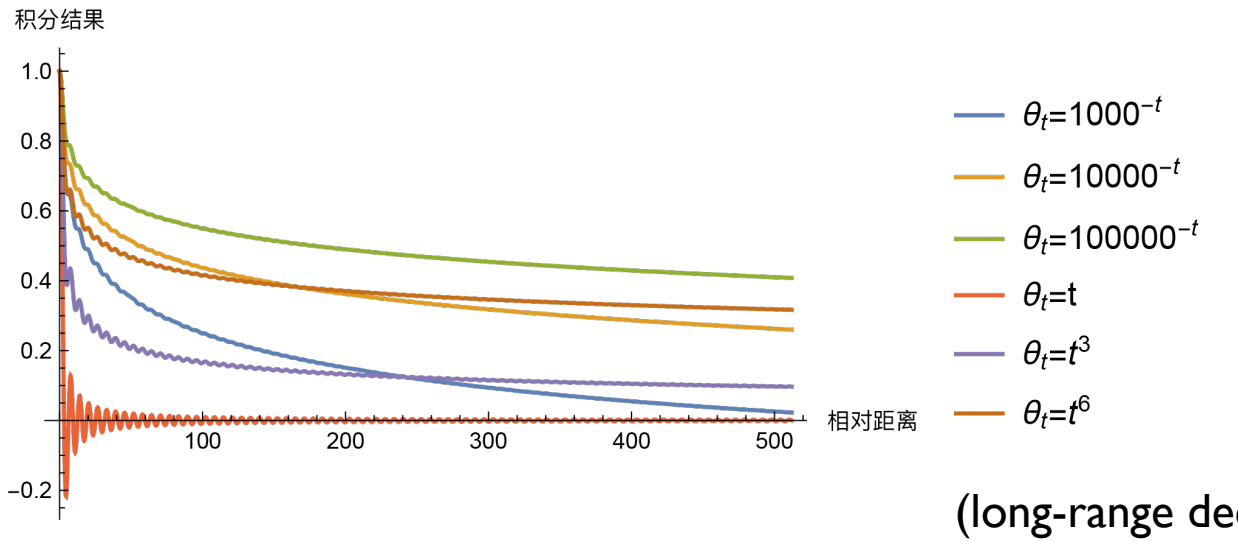
Sinusoidal Positional Encoding

Some intuition about how it works

- Extend to high dimension:

$$\boldsymbol{p}_m = \begin{pmatrix} e^{\mathrm{i}m\theta_0} \\ \vdots \\ e^{\mathrm{i}m\theta_{d/2-1}} \end{pmatrix} \quad \Leftrightarrow \quad \boldsymbol{p}_m = \begin{pmatrix} \cos m\theta_0 \\ \sin m\theta_0 \\ \vdots \\ \cos m\theta_{d/2-1} \\ \sin m\theta_{d/2-1} \end{pmatrix}$$

- Choose θ^i:



(long-range decay)

苏剑林. “Transformer升级之路：1、Sinusoidal位置编码追根溯源 ”(2021).

Relative Positional Encodings

- Consider the attention module with absolute positional encoding:

$$\begin{cases} \boldsymbol{q}_i = (\boldsymbol{x}_i + \boldsymbol{p}_i)\boldsymbol{W}_Q \\ \boldsymbol{k}_j = (\boldsymbol{x}_j + \boldsymbol{p}_j)\boldsymbol{W}_K \\ \boldsymbol{v}_j = \boldsymbol{x}_j \boldsymbol{W}_V \\ a_{i,j} = softmax\left(\boldsymbol{q}_i \boldsymbol{k}_j^\top\right) \\ \boldsymbol{o}_i = \sum_j a_{i,j} \boldsymbol{v}_j \end{cases}$$

- Expand the QK term:

$$\boldsymbol{q}_i \boldsymbol{k}_j^\top = (\boldsymbol{x}_i + \boldsymbol{p}_i)\boldsymbol{W}_Q \boldsymbol{W}_K^\top (\boldsymbol{x}_j + \boldsymbol{p}_j)^\top = \boldsymbol{x}_i \boldsymbol{W}_Q \boldsymbol{W}_K^\top \boldsymbol{x}_j^\top + \boldsymbol{x}_i \boldsymbol{W}_Q \boldsymbol{W}_K^\top \boldsymbol{p}_j^\top + \boldsymbol{p}_i \boldsymbol{W}_Q \boldsymbol{W}_K^\top \boldsymbol{x}_j^\top + \boldsymbol{p}_i \boldsymbol{W}_Q \boldsymbol{W}_K^\top \boldsymbol{p}_j^\top$$

  - Shaw et al., 2018: $\boldsymbol{x}_i \boldsymbol{W}_Q \boldsymbol{W}_K^\top \boldsymbol{x}_j^\top + \boldsymbol{x}_i \boldsymbol{W}_Q {\boldsymbol{R}_{i,j}^K}^\top$, where $\boldsymbol{R}_{i,j}^K$ is a function of $i - j$
  - XLNET: $\boldsymbol{x}_i \boldsymbol{W}_Q \boldsymbol{W}_K^\top \boldsymbol{x}_j^\top + \boldsymbol{x}_i \boldsymbol{W}_Q \boldsymbol{W}_{K,R}^\top \boldsymbol{R}_{i-j}^\top + \boldsymbol{u} \boldsymbol{W}_K^\top \boldsymbol{x}_j^\top + \boldsymbol{v} \boldsymbol{W}_{K,R}^\top \boldsymbol{R}_{i-j}^\top$
  - T5: $\boldsymbol{x}_i \boldsymbol{W}_Q \boldsymbol{W}_K^\top \boldsymbol{x}_j^\top + \boldsymbol{\beta}_{i,j}$
  - DeBERTa: $\boldsymbol{x}_i \boldsymbol{W}_Q \boldsymbol{W}_K^\top \boldsymbol{x}_j^\top + \boldsymbol{x}_i \boldsymbol{W}_Q \boldsymbol{W}_K^\top \boldsymbol{R}_{i,j}^\top + \boldsymbol{R}_{j,i} \boldsymbol{W}_Q \boldsymbol{W}_K^\top \boldsymbol{x}_j^\top$

Shaw, Peter, Jakob Uszkoreit, and Ashish Vaswani. "Self-attention with relative position representations." (NAACL 2018).
Dai, Zihang, et al. "Transformer-xl: Attentive language models beyond a fixed-length context." (ACL 2019).
Raffel, Colin, et al. "Exploring the limits of transfer learning with a unified text-to-text transformer." (JMLR 2020).
He, Pengcheng, et al. "Deberta: Decoding-enhanced bert with disentangled attention." (ICLR 2021).
苏剑林. "让研究人员绞尽脑汁的Transformer位置编码." (2021).

RoPE: Rotary Position Embedding

- Implement relative PE with absolute PE.
- Attention with absolute PE: $\tilde{\boldsymbol{q}}_m = \boldsymbol{f}(\boldsymbol{q}, m), \quad \tilde{\boldsymbol{k}}_n = \boldsymbol{f}(\boldsymbol{k}, n)$
- Find a tractable solution of: $\langle \boldsymbol{f}(\boldsymbol{q}, m), \boldsymbol{f}(\boldsymbol{k}, n) \rangle = g(\boldsymbol{q}, \boldsymbol{k}, m-n)$
  - Initial condition: $\boldsymbol{f}(\boldsymbol{q}, 0) = \boldsymbol{q}, \boldsymbol{f}(\boldsymbol{k}, 0) = \boldsymbol{k}$
  - Derive with complex number:

    $$\mathrm{Re}[\boldsymbol{f}(\boldsymbol{q}, m)\boldsymbol{f}^*(\boldsymbol{k}, n)] = g(\boldsymbol{q}, \boldsymbol{k}, m-n) \Leftarrow \boldsymbol{f}(\boldsymbol{q}, m)\boldsymbol{f}^*(\boldsymbol{k}, n) = \boldsymbol{g}(\boldsymbol{q}, \boldsymbol{k}, m-n)$$

  - Let:

    $$\boldsymbol{f}(\boldsymbol{q}, m) = R_f(\boldsymbol{q}, m)e^{\mathrm{i}\Theta_f(\boldsymbol{q}, m)}$$
    $$\boldsymbol{f}(\boldsymbol{k}, n) = R_f(\boldsymbol{k}, n)e^{\mathrm{i}\Theta_f(\boldsymbol{k}, n)}$$
    $$\boldsymbol{g}(\boldsymbol{q}, \boldsymbol{k}, m-n) = R_g(\boldsymbol{q}, \boldsymbol{k}, m-n)e^{\mathrm{i}\Theta_g(\boldsymbol{q}, \boldsymbol{k}, m-n)}$$

    Equation:

    $$R_f(\boldsymbol{q}, m)R_f(\boldsymbol{k}, n) = R_g(\boldsymbol{q}, \boldsymbol{k}, m-n)$$
    $$\Theta_f(\boldsymbol{q}, m) - \Theta_f(\boldsymbol{k}, n) = \Theta_g(\boldsymbol{q}, \boldsymbol{k}, m-n)$$

  - One solution: $\boldsymbol{f}(\boldsymbol{q}, m) = \boldsymbol{q}e^{\mathrm{i}m\theta}$, i.e., rotation operation $\boldsymbol{f}(\boldsymbol{q}, m) = \begin{pmatrix} \cos m\theta & -\sin m\theta \\ \sin m\theta & \cos m\theta \end{pmatrix} \begin{pmatrix} q_0 \\ q_1 \end{pmatrix}$

Su, Jianlin, et al. "Roformer: Enhanced transformer with rotary position embedding." (2021).
苏剑林. "Transformer升级之路：2、博采众长的旋转式位置编码"(2021).

RoPE: Rotary Position Embedding

- High dimension: $\theta_i = 10000^{-2i/d}$

$$\underbrace{\begin{pmatrix} \cos m\theta_0 & -\sin m\theta_0 & 0 & 0 & \cdots & 0 & 0 \\ \sin m\theta_0 & \cos m\theta_0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cos m\theta_1 & -\sin m\theta_1 & \cdots & 0 & 0 \\ 0 & 0 & \sin m\theta_1 & \cos m\theta_1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \cos m\theta_{d/2-1} & -\sin m\theta_{d/2-1} \\ 0 & 0 & 0 & 0 & \cdots & \sin m\theta_{d/2-1} & \cos m\theta_{d/2-1} \end{pmatrix}}_{\boldsymbol{\mathcal{R}}_m} \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \\ \vdots \\ q_{d-2} \\ q_{d-1} \end{pmatrix} \Leftrightarrow \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \\ \vdots \\ q_{d-2} \\ q_{d-1} \end{pmatrix} \otimes \begin{pmatrix} \cos m\theta_0 \\ \cos m\theta_0 \\ \cos m\theta_1 \\ \cos m\theta_1 \\ \vdots \\ \cos m\theta_{d/2-1} \\ \cos m\theta_{d/2-1} \end{pmatrix} + \begin{pmatrix} -q_1 \\ q_0 \\ -q_3 \\ q_2 \\ \vdots \\ -q_{d-1} \\ q_{d-2} \end{pmatrix} \otimes \begin{pmatrix} \sin m\theta_0 \\ \sin m\theta_0 \\ \sin m\theta_1 \\ \sin m\theta_1 \\ \vdots \\ \sin m\theta_{d/2-1} \\ \sin m\theta_{d/2-1} \end{pmatrix}$$

which satisfies: $(\boldsymbol{\mathcal{R}}_m \boldsymbol{q})^\top (\boldsymbol{\mathcal{R}}_n \boldsymbol{k}) = \boldsymbol{q}^\top \boldsymbol{\mathcal{R}}_m^\top \boldsymbol{\mathcal{R}}_n \boldsymbol{k} = \boldsymbol{q}^\top \boldsymbol{\mathcal{R}}_{n-m} \boldsymbol{k}$

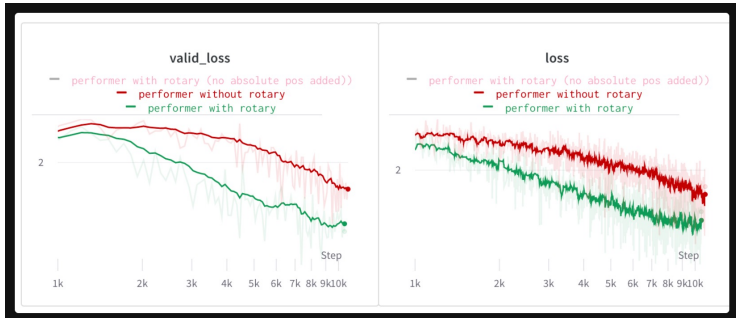
- Superior performance in terms of ppl. on a wide-range of model architectures.

Type	OWT2 Loss	OWT2 Ppl.
Learned Absolute	2.809	16.59
T5 RPE	2.801	16.46
Rotary	2.759	15.78

Final validation loss / ppl scores on OWT2 validation set at 55k steps (~30B tokens)

Type	Pile Loss	Pile Ppl.
Learned Absolute	2.240	9.393
T5 RPE	2.223	9.234
Rotary	2.173	8.784

Final validation loss / ppl scores on Pile validation set at 8k steps (~8B tokens)



Su, Jianlin, et al. “Roformer: Enhanced transformer with rotary position embedding.” (2021).
Biderman, Stella, et al. “Rotary Embeddings: A Relative Revolution.” (2021).
苏剑林. “Transformer升级之路：2、博采众长的旋转式位置编码”(2021).

Length Extrapolation / Generalization

- Length extrapolation: overcoming inconsistencies between training and inference
  - Unseen positional encoding during inference.
  - More attention participants during inference.

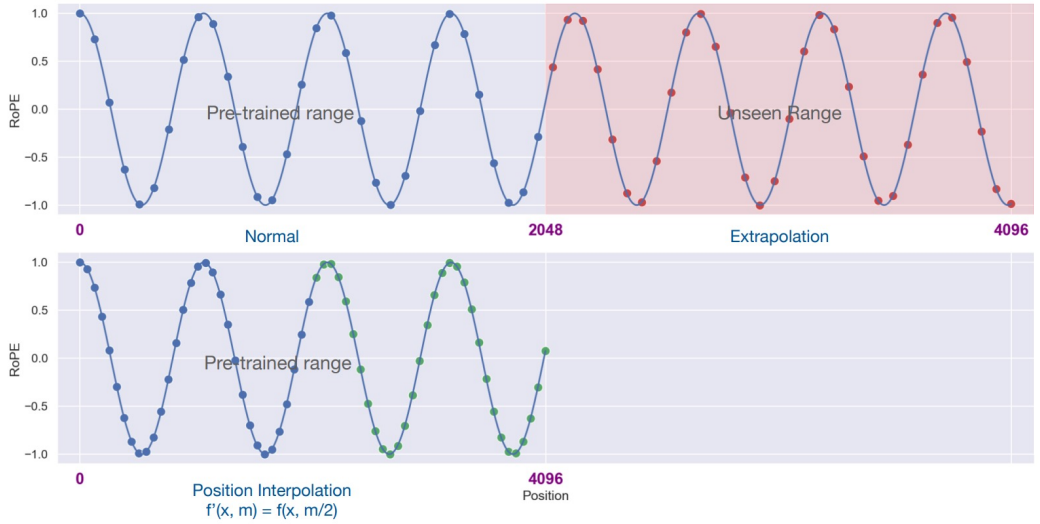
- For attention: confine the context window with attention mask (window attention), e.g., change $\boldsymbol{q}_m^\top \boldsymbol{k}_n$ to $\boldsymbol{q}_m^\top \boldsymbol{k}_n - \lambda|m-n|$.

- For PE: refine existing RoPE to support longer context while preserving performance within original context length.

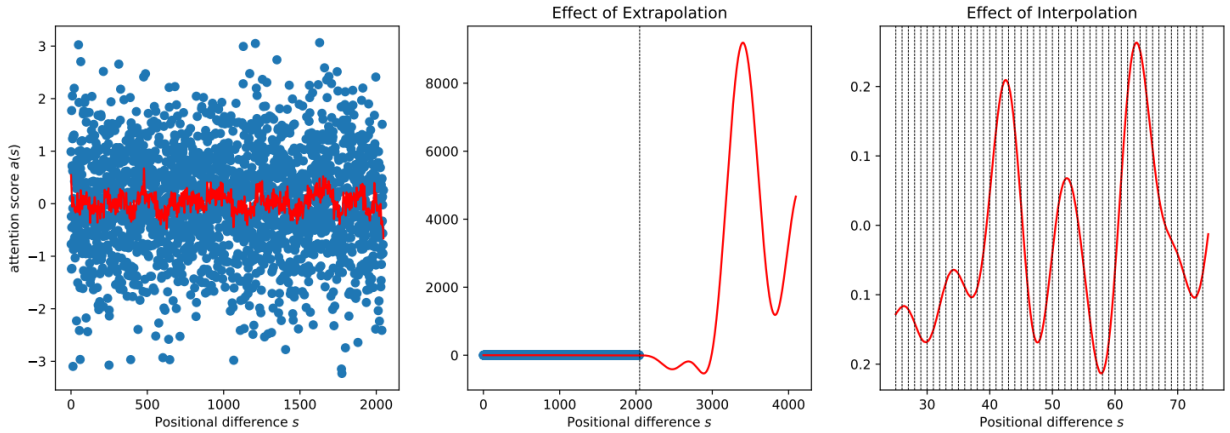
Press, Ofir, Noah A. Smith, and Mike Lewis. "Train short, test long: Attention with linear biases enables input length extrapolation." (ICLR 2022).
苏剑林. “Transformer升级之路：7、长度外推性与局部注意力.”(2023).

Position Interpolation for RoPE
Length extrapolation w/ minimal fine-tuning

- Extrapolating outside the training positions may produce catastrophic values.
- Since PE can be applied on non-integer positions, one could interpolate the position encodings at neighboring integer positions.



Extrapolation v.s. Interpolation



Catastrophic behavior of extrapolation

Chen, Shouyuan, et al. "Extending context window of large language models via positional interpolation." (2023).

Position Interpolation for RoPE
Length extrapolation w/ minimal fine-tuning

- Method: Given a pre-trained context length L and a new context length $L' > L$, position interpolation replaces RoPE by:

$$\boldsymbol{f}'(\boldsymbol{x}, m) = \boldsymbol{f}(\boldsymbol{x}, \frac{mL}{L'})$$

- Numerical bound: for unbounded $h_j := (q_{2j} + iq_{2j+1})(k_{2j} - ik_{2j+1})$, the upper bound for difference between the result attention score and the typical attention score for interpolation is ~600x smaller than extrapolation.

Chen, Shouyuan, et al. "Extending context window of large language models via positional interpolation." (2023).

Position Interpolation for RoPE
Length extrapolation w/ minimal fine-tuning

- Experiments:
  - Extend LLaMA (7B, 13B, 33B, 65B) to context window sizes up to 32768 using direct fine-tuning (10k steps) or Position Interpolation(1k steps).

Size	Model Context Window	Method	Evaluation Context Window Size 2048	4096	8192	16384	32768
7B	2048	None	7.20	$> 10^3$	$> 10^3$	$> 10^3$	$> 10^3$
7B	8192	FT	7.21	7.34	7.69	-	-
7B	8192	PI	7.13	6.96	6.95	-	-
7B	16384	PI	7.11	6.93	6.82	6.83	-
7B	32768	PI	7.23	7.04	6.91	6.80	6.77
13B	2048	None	6.59	-	-	-	-
13B	8192	FT	6.56	6.57	6.69	-	-
13B	8192	PI	6.55	6.42	6.42	-	-
13B	16384	PI	6.56	6.42	6.31	6.32	-
13B	32768	PI	6.54	6.40	6.28	6.18	6.09
33B	2048	None	5.82	-	-	-	-
33B	8192	FT	5.88	5.99	6.21	-	-
33B	8192	PI	5.82	5.69	5.71	-	-
33B	16384	PI	5.87	5.74	5.67	5.68	-
65B	2048	None	5.49	-	-	-	-
65B	8192	PI	5.42	5.32	5.37	-	-

Table 1: Evaluation perplexity on PG19 dataset (Rae et al., 2020). FT: Direct Fine-tuning. PI: Position Interpolation. Model fine-tuned with PI shows progressively lower perplexity with longer context window, showing that PI can leverage long context well, while the perplexity of FT increases over longer window. Note that overall the perplexity is higher compared to Table 2 since PG19 has very different writing styles.

Size	Model Context Window	Method	Fine-tuning steps 200	400	600	800	1000	10000
7B	8192	FT	1792	2048	2048	2048	2304	2560
33B	8192	FT	1792	2048	1792	2048	2304	-
7B	8192	PI	8192	8192	8192	8192	8192	-
7B	16384	PI	16384	16384	16384	16384	16384	-
7B	32768	PI	32768	32768	18432	32768	32768	-
33B	8192	PI	8192	8192	8192	8192	8192	-
33B	16384	PI	16384	16384	16384	16384	16384	-

Table 4: Effective context window sizes after fine-tuning. FT: Direct fine-tuning. PI: Position Interpolation.

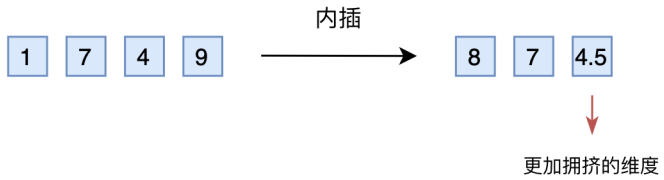
Model Size	Context Window	Fine-tune on	BoolQ	PIQA	Race-M	Race-H	WinoGrande
7B	2048	None	76.1	78.9	55.7	42.2	69.6
7B	8192	Pile	73.2	78.2	53.8	41.7	69.0
7B	16384	Pile	69.8	77.6	53.3	40.9	67.8
7B	32768	Pile	64.7	77.2	50.1	39.6	66.9
7B	8192	RedPajama	75.5	77.4	54.5	41.5	68.1
33B	2048	None	81.6	80.2	61.1	45.9	76.2
33B	8192	Pile	80.2	80.7	60.2	45.7	75.9

Table 5: Zero-shot performance on a subset of LLaMA Benchmarks. Models extended by Position Interpolation comparable performance as the original models, except for BoolQ dataset that may require models to pay close attention to word ordering in a short reference paragraph.

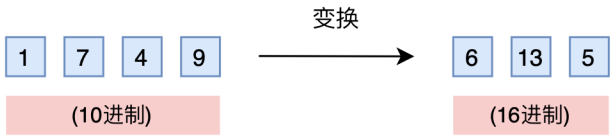
Chen, Shouyuan, et al. "Extending context window of large language models via positional interpolation." (2023).

NTK-Aware Scaled RoPE
Length extrapolation w/o fine-tuning

- Linear position interpolation ($\boldsymbol{f}'(\boldsymbol{x}, m) = \boldsymbol{f}(\boldsymbol{x}, \frac{mL}{L'})$) may overly narrow the spacing on the unit-digit of the position:



- Amortize such 'narrowing' pressure to all digits via numeral system conversion:



苏剑林. "Transformer升级之路：10、RoPE是一种β进制编码."(2023).
NTK-Aware Scaled RoPE allows LLaMA models to have extended (8k+) context size without any fine-tuning and minimal perplexity degradation. : LocalLLaMA (reddit.com)

NTK-Aware Scaled RoPE
Length extrapolation w/o fine-tuning

- Recall that for a β-numerical system, the m-th digit of a number n is:

$$\left\lfloor \frac{n}{\beta^{m-1}} \right\rfloor \bmod \beta$$

- RoPE is β-system for position n :

$$\left[\cos\left(\frac{n}{\beta^0}\right), \sin\left(\frac{n}{\beta^0}\right), \cos\left(\frac{n}{\beta^1}\right), \sin\left(\frac{n}{\beta^1}\right), \cdots, \cos\left(\frac{n}{\beta^{d/2-1}}\right), \sin\left(\frac{n}{\beta^{d/2-1}}\right)\right]$$

  where $\beta = 10000^{2/d}$.

- To expand the length to kL within the same number of digits d, one could adopt $\beta\sqrt[d]{k}$-system.
- For RoPE, we could adopt $\beta_k = 10000k^{2/d}$.

苏剑林. “Transformer升级之路：10、RoPE是一种β进制编码.”(2023).
NTK-Aware Scaled RoPE allows LLaMA models to have extended (8k+) context size without any fine-tuning and minimal perplexity degradation. : LocalLLaMA (reddit.com)

NTK-Aware Scaled RoPE
Length extrapolation w/o fine-tuning

- Another implementation: extrapolation for high-freq, interpolation for low-freq.
- Consider RoPE:

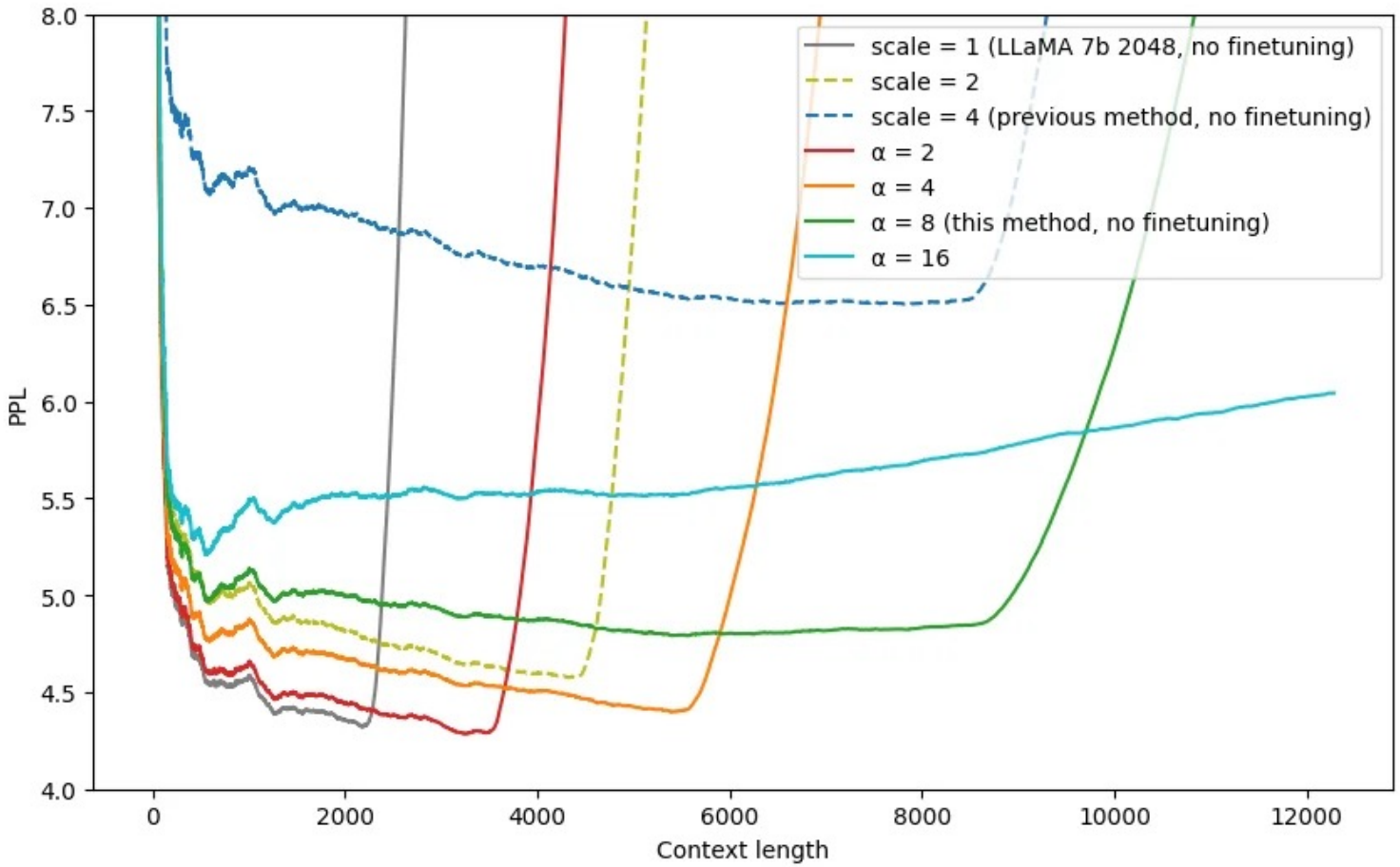
$$\left[\cos\left(\frac{n}{\beta^0}\right), \sin\left(\frac{n}{\beta^0}\right), \cos\left(\frac{n}{\beta^1}\right), \sin\left(\frac{n}{\beta^1}\right), \cdots, \cos\left(\frac{n}{\beta^{d/2-1}}\right), \sin\left(\frac{n}{\beta^{d/2-1}}\right)\right]$$

- The lowest-freq term is $\frac{n}{\beta^{d/2-1}}$, let $\frac{n}{(\beta\lambda)^{d/2-1}} = \frac{n/k}{\beta^{d/2-1}}$, then $\lambda = k^{2/(d-2)}$
- For the highest-freq term $\frac{n}{\beta}$, since $\lambda \approx 1$ when d is large. Hence, it's equivalent to direct position extrapolation.

苏剑林. “Transformer升级之路：10、RoPE是一种β进制编码.”(2023).
NTK-Aware Scaled RoPE allows LLaMA models to have extended (8k+) context size without any fine-tuning and minimal perplexity degradation. : LocalLLaMA (reddit.com)

NTK-Aware Scaled RoPE
Length extrapolation w/o fine-tuning

- Experiments:



NTK-Aware Scaled RoPE allows LLaMA models to have extended (8k+) context size without any fine-tuning and minimal perplexity degradation. : LocalLLaMA (reddit.com)

Extensive Study on RoPE for Long Context Modeling

- Vanilla RoPE imposes a large decay on attention scores for distant tokens.
- RoPE introduces large "oscillation" in the long-range regions.
- Experiment with several RoPE variants:
  - Vanilla: $\boldsymbol{x}_j e^{\mathrm{i}mb^{\frac{-2j}{d}}}$
  - Position Interpolation (PI): $\boldsymbol{x}_j e^{\mathrm{i}\alpha mb^{\frac{-2j}{d}}}$
  - Adjusted Base Frequency (ABF): $\boldsymbol{x}_j e^{\mathrm{i}m(\beta b)^{\frac{-2j}{d}}}$
  - Xpos + ABF: $\boldsymbol{x}_j e^{\xi_j m + \mathrm{i}m(\beta b)^{\frac{-2j}{d}}}$

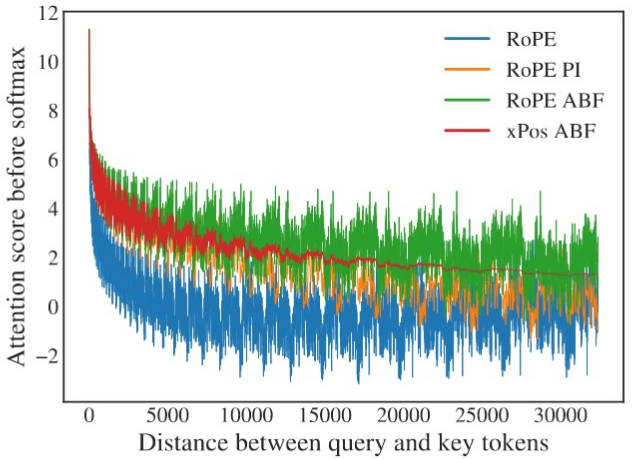
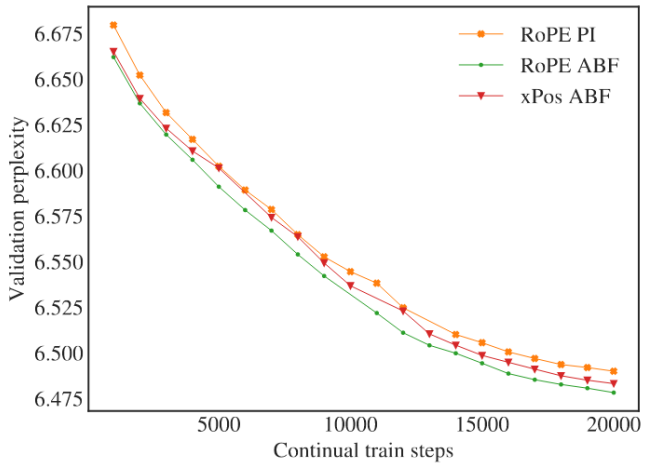


Figure 4: Decaying raw attention scores for distant tokens of explored positional encoding variants (assuming keys and queries are all-ones vectors).

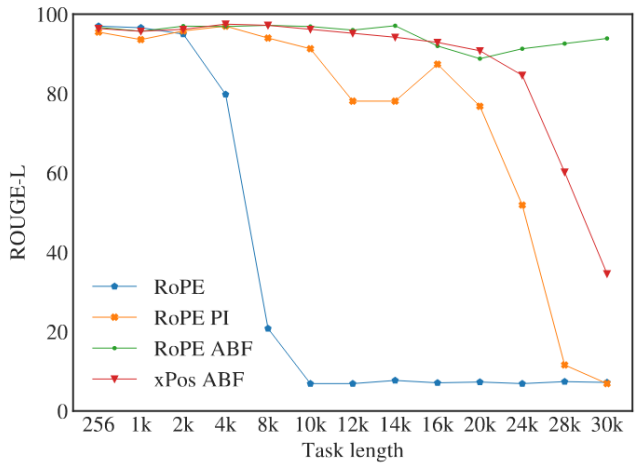
Xiong, Wenhan, et al. "Effective long-context scaling of foundation models." (2023).

Extensive Study on RoPE for Long Context Modeling

- Experiments:
  - Continual pre-training + instruction fine-tuning on pre-trained LLaMA-2 checkpoints



(a) Validation PPL (16k-token sequences) on a held-out long-context dataset.



(b) Performance on FIRST-SENTENCE-RETRIEVAL task.

Figure 5: Comparison of positional encoding variants on synthetic sentence retrieval task and validation perplexity evolution during continual pretraining.

	HumanEval	Math	MMLU	HellaSwag	TQA
RoPE	14.63	**3.62**	45.69	76.31	65.23
RoPE PI	15.24	3.08	45.84	76.65	65.96
RoPE-ABF	**17.07**	3.52	**46.24**	**76.73**	66.04
xPos-ABF	16.46	3.54	45.72	76.68	**66.14**

Table 6: The performance of models with different positional encoding variants on standard short-context benchmarks.

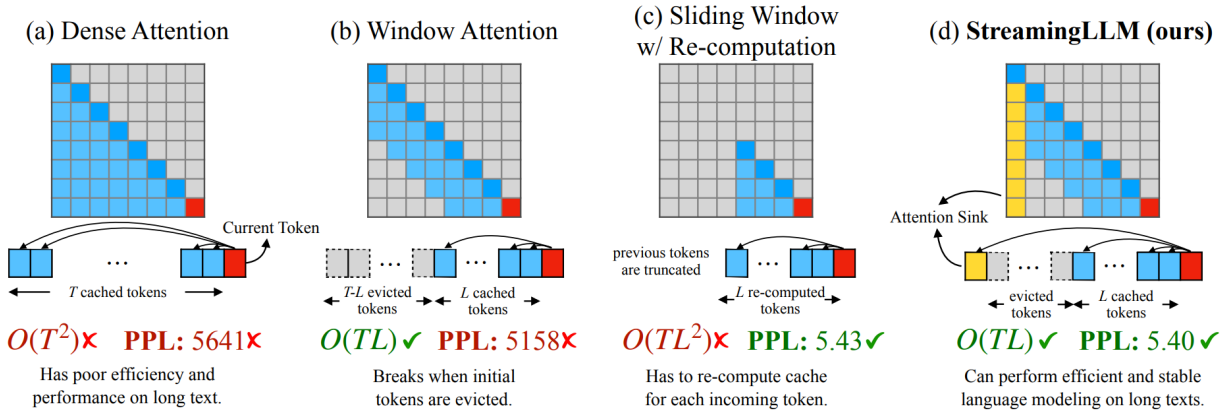
PE	Books	CC	Wikipedia
RoPE	6.548	6.816	3.802
RoPE PI	6.341	6.786	3.775
RoPE ABF	**6.323**	**6.780**	**3.771**
xPos ABF	6.331	**6.780**	**3.771**

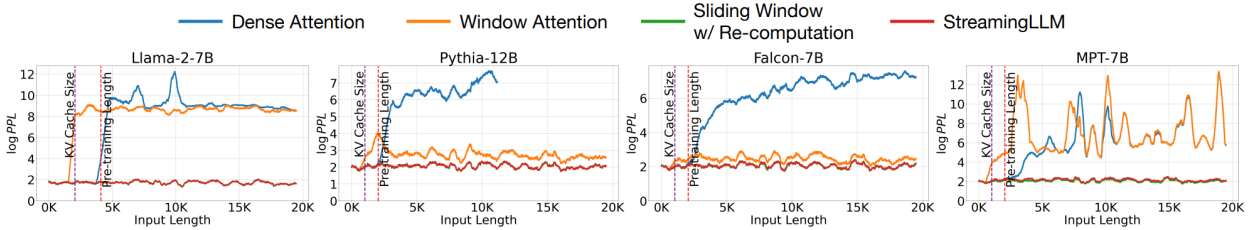
Table 5: Validation perplexity of models with different positional encoding variants. All samples are 32,768-token sequences (CC: CommonCrawl).

Xiong, Wenhan, et al. "Effective long-context scaling of foundation models." (2023).

Improved Attention Scheme for Long Context Modeling
Refine the window attention

- For long context language modeling, there are two main challenges:
  - Pre-trained LMs have limited length extrapolation capabilities.
  - The KV cache on decoding stage can lead to excessive memory or increasing decoding latency (if we re-compute some of them).
- A direct solution is window attention.
  - However, it is sub-optimal in quality or efficiency.
  - Window attention breaks when the first token is excluded.

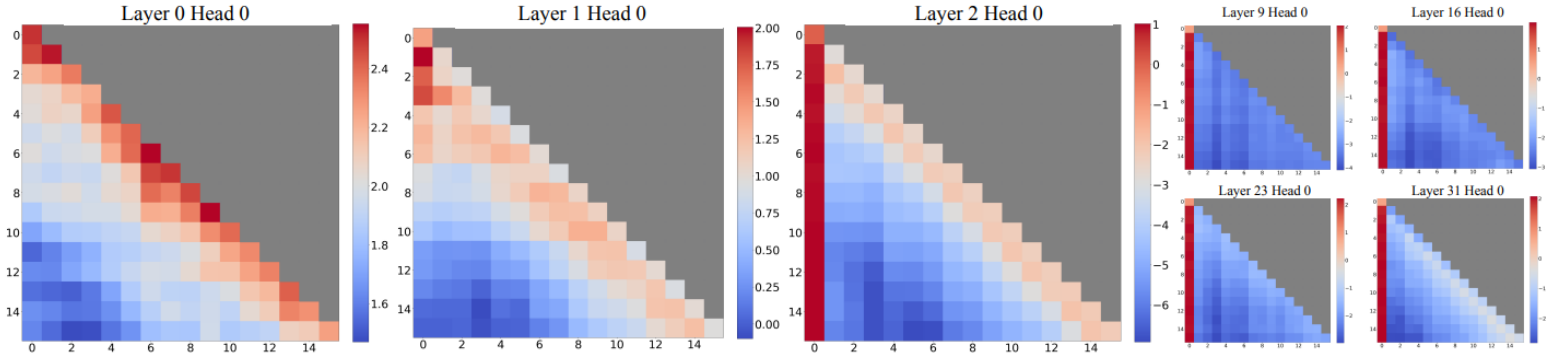




Xiao, Guangxuan, et al. "Efficient Streaming Language Models with Attention Sinks." (2023).

Improved Attention Scheme for Long Context Modeling
How window attention breaks?

- Empirically, LMs consistently focuses on the initial tokens across all layers and heads except the bottom two layers.
  - Even when substituting the first token with a line break "\n".



- Intuitively, the model learned a biased behavior that tends to dump unnecessary attention values to specific tokens.

Xiao, Guangxuan, et al. "Efficient Streaming Language Models with Attention Sinks." (2023).

Improved Attention Scheme for Long Context Modeling
Window attention with attention sinks

- Method: preserve several start tokens when performing window attention.
  - Use attention sinks (i.e., four initial tokens) to stabilize the computation of attention score.
  - Use rolling KV cache around the most recent tokens to save memory.

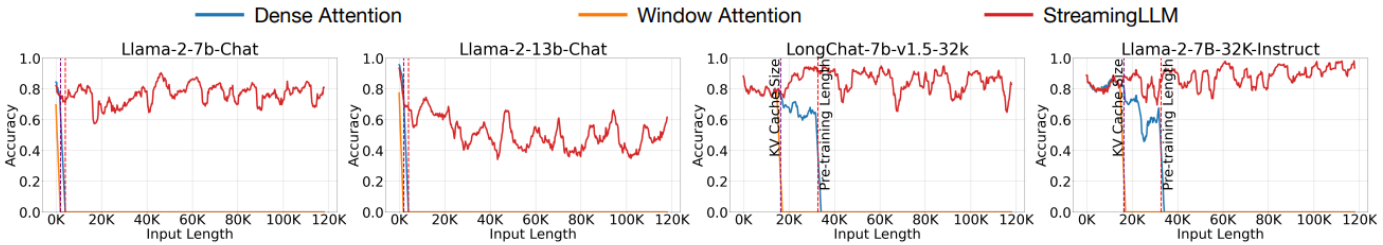


Figure 9: Performance on the StreamEval benchmark. Accuracies are averaged over 100 samples.

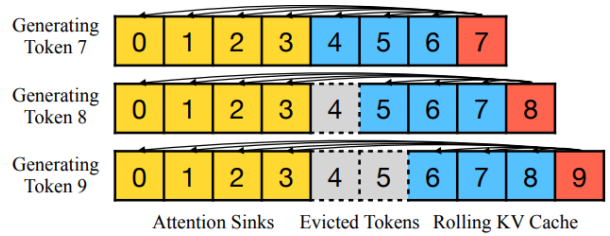


Figure 4: The KV cache of StreamingLLM.

Llama-2-13B	PPL (↓)
0 + 1024 (Window)	5158.07
4 + 1020	5.40
4"\n"+1020	5.60

Eval on 65k tokens

Cache Config	0+2048	1+2047	2+2046	4+2044	8+2040
Falcon-7B	17.90	12.12	12.12	12.12	12.12
MPT-7B	460.29	14.99	15.00	14.99	14.98
Pythia-12B	21.62	11.95	12.09	12.09	12.02
Cache Config	**0+4096**	**1+4095**	**2+4094**	**4+4092**	**8+4088**
Llama-2-7B	3359.95	11.88	10.51	9.59	9.54

Eval on 400k tokens

Xiao, Guangxuan, et al. "Efficient Streaming Language Models with Attention Sinks." (2023).

Improved Attention Scheme for Long Context Modeling
Explicitly build attention sink during pre-training

- Pre-training with attention sink:
  - Zero Sink: replace the regular Softmax in attention with Softmax-1: $\mathrm{SoftMax}_1(x)_i = \frac{e^{x_i}}{1 + \sum_{j=1}^{N} e^{x_j}}$
  - Learnable Sink: Add a common token at the beginning of all samples.

Cache Config	0+1024	1+1023	2+1022	4+1020
Vanilla	27.87	18.49	18.05	18.05
Zero Sink	29214	19.90	18.27	18.01
Learnable Sink	1235	**18.01**	18.01	18.02

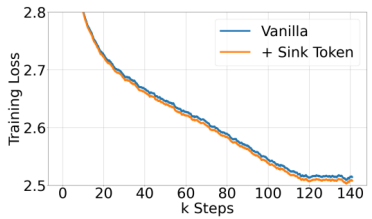
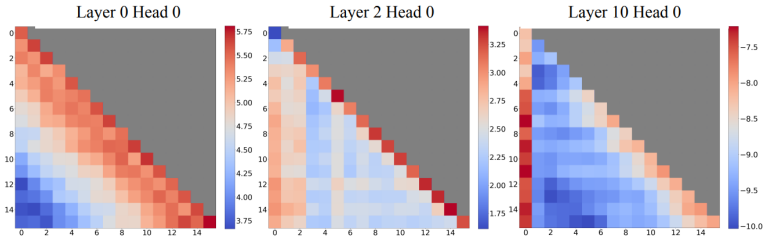


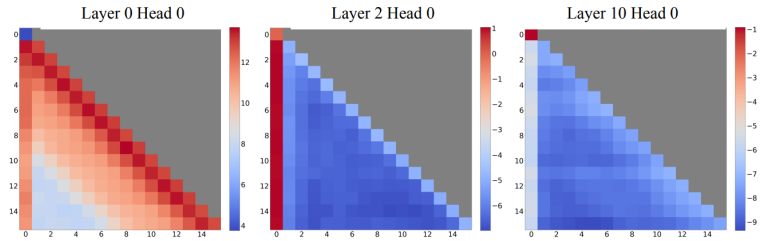
Figure 6: Pre-training loss curves of models w/ and w/o sink tokens. Two models have a similar convergence trend.

Table 4: Zero-shot accuracy (in %) across 7 NLP benchmarks, including ARC-[Challenge, Easy], HellaSwag, LAMBADA, OpenbookQA, PIQA, and Winogrande. The inclusion of a sink token during pre-training doesn't harm the model performance.

Methods	ARC-c	ARC-e	HS	LBD	OBQA	PIQA	WG
Vanilla	18.6	45.2	29.4	39.6	16.0	62.2	50.1
+Sink Token	**19.6**	**45.6**	**29.8**	**39.9**	**16.6**	**62.6**	**50.8**



Pre-Trained without Sink Token



Pre-Trained with Sink Token

Xiao, Guangxuan, et al. "Efficient Streaming Language Models with Attention Sinks." (2023).

Data Mix for Continual Pre-training

- The data for continual pre-training Long LLaMA-2 consists of two parts:
  - Existing pre-training data of LLaMA-2
  - New long text data
- Adjusting the length distribution of the pretrain data does not provide major benefits for long context modeling.
- Long LLMs can be trained with limited long text data. And the performance gain mainly comes from the data quality itself instead of length distribution.

Continual Pretrain Data	NarrativeQA Δ F1	Qasper Δ F1	Quality Δ EM	QMSum Δ ROUGE-geo
LLAMA 2 LONG data mix	**23.70%**	**43.64%**	**75.5%**	**45.70%**
LLAMA 2 data mix	18.23%	38.12%	60.3%	44.87%
- remove long text data	19.48%	39.14%	67.1%	36.60%
- upsample existing long text data	22.15%	36.82%	65.0%	42.83%

Table 7: Comparison of different pretraining data mix on long-context tasks. Instead of showing the absolute performance, we report relative improvements over the 7B LLAMA 2 which has a 4,096-token context window. All models are evaluated with prompts truncated at 16,384 tokens.

Continual Pretrain Data	HumanEval	Math	MMLU	HellaSwag	TQA
LLAMA 2 LONG data mix	**17.08**	**4.09**	**48.62**	76.74	66.24
LLAMA 2 data mix	15.24	3.61	46.30	76.63	**66.71**
- remove long text data	17.07	3.57	46.25	**76.76**	65.90
- upsample existing long text data	17.07	3.53	46.25	76.74	66.04

Table 8: Standard short task performance of long-context models with different pretrain data mix.

Xiong, Wenhan, et al. "Effective long-context scaling of foundation models." (2023).

Instruction Tuning

- Fine-tuning only on the short instruction data is enough to produce a decent long chat LM.
- Mixing long context pre-training data with instruction data could further boost LM's performance.
- Different from common instruction tuning that only calculates LM loss on the response part, calculating the LM loss on the whole sequence (i.e., instruction-response pair) is better for long context LMs.

Settings	Qasper	NarrativeQA	QuALITY	SummScreenFD	QMSum
LLAMA 2 CHAT baseline	12.2	9.13	56.7	10.5	14.4
LLAMA 2 LONG *finetuned* with:					
"RLHF V5"	22.3	13.2	71.4	14.8	16.9
"RLHF V5" mix pretrain	23.7	16.6	76.2	**15.7**	17.8
"RLHF V5" mix self-inst w/o LM loss	35.7	22.3	59.3	12.2	13.4
"RLHF V5" mix self-inst with LM loss	**38.9**	**23.3**	**77.3**	14.5	**18.5**

Table 9: Comparison of different instruction finetuning data mixes.

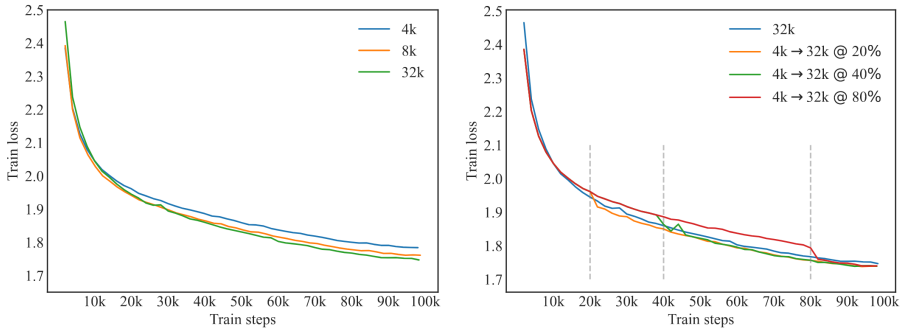
Xiong, Wenhan, et al. "Effective long-context scaling of foundation models." (2023).

Training Curriculum

- Question: does pre-training from scratch with long sequences yield better performance than continual pre-training?
- Experiment: two-stage training that begins with 4096 context length and switch to 32768 when the model completes 20%, 40%, 80% of whole training process.
  - Same number of training tokens.
  - Same number of tokens per each gradient update.
- Pre-trained LM could efficiently obtain long context modeling ability with continual pre-training (save ~40% FLOPs with no performance loss).

Pretrain Curriculum	FLOPs	NarrativeQA F1	Qasper F1	Quality EM	QMSum ROUGE-geo
32k from scratch	3.783×10^{22}	18.5	28.6	37.9	11.46
4k→32k @ 20%	3.405×10^{22}	20.0	28.1	38.8	12.09
4k→32k @ 40%	3.026×10^{22}	20.1	27.0	37.4	12.44
4k→32k @ 80%	2.270×10^{22}	18.5	25.0	38.3	11.00

Table 10: Comparison of models with different training curricula on long context QA tasks.



Xiong, Wenhan, et al. "Effective long-context scaling of foundation models." (2023).

How LMs use Long Contexts
Have they already met our needs?

- How well recent long LLMs utilize long texts.
- Models:
  - Open: MPT-30B-Instruct (uses AliBi, 8k), LongChat-13B (16k, RoPE)
  - Close: GPT-3.5-Turbo (4k & 16k), Claude-1.3 (100k)
- Task1: Multi-document QA task
  - Input: a user query and k documents (2.7k tokens each), where exactly **one** document contains the answer.
  - Variable: Total input context length and the position of relevant information.
  - The query is put after the given context.

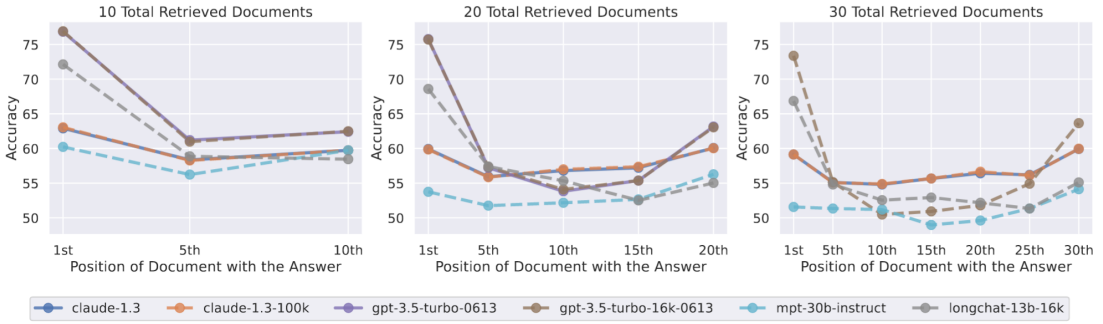


Figure 5: The effect of changing the position of relevant information (document containing the answer) on multi-document question answering performance. Lower positions are closer to the start of the input context. Performance is generally highest when relevant information is positioned at the very start or very end of the context, and rapidly degrades when models must reason over information in the middle of their input context.

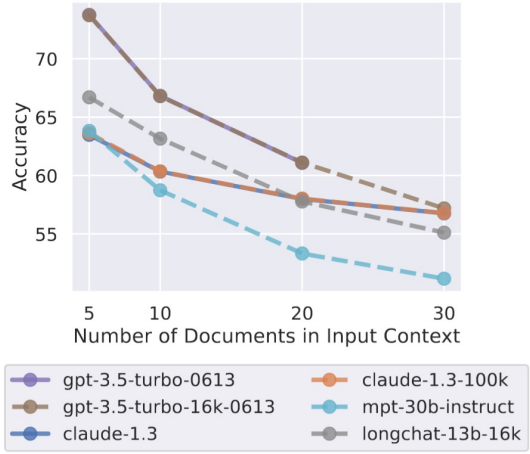


Figure 6: Language model performance (averaged across position of relevant information) on the multi-document question answering task decreases as the input context grows longer.

Liu, Nelson F., et al. "Lost in the middle: How language models use long contexts." (2023).

How LMs use Long Contexts

- Task2: Synthetic Key-Value retrieval task
  - A string-serialized JSON object with k key-value pairs.
  - Each key and value are randomly-generated UUIDs.
  - A particular key to be extracted.

Input Context

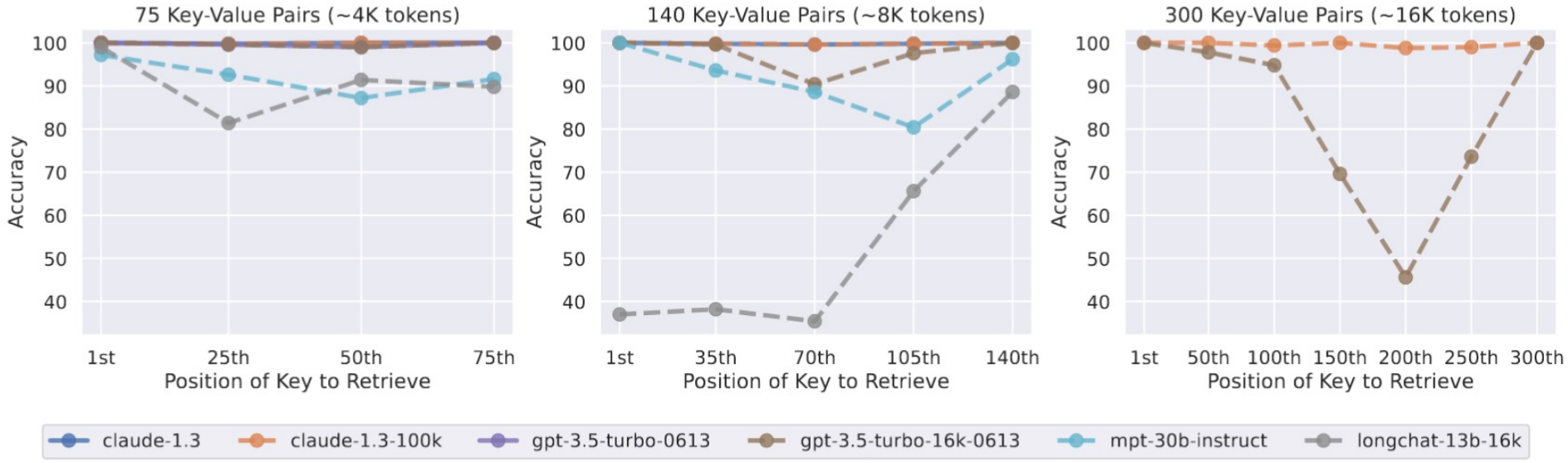
```
Extract the value corresponding to the specified key in the JSON object below.

JSON data:
{"2a8d601d-1d69-4e64-9f90-8ad825a74195": "bb3ba2a5-7de8-434b-a86e-a88bb9fa7289",
 "a54e2eed-e625-4570-9f74-3624e77d6684": "d1ff29be-4e2a-4208-a182-0cea716be3d4",
 "9f4a92b9-5f69-4725-ba1e-403f08dea695": "703a7ce5-f17f-4e6d-b895-5836ba5ec71c",
 "52a9c80c-da51-4fc9-bf70-4a4901bc2ac3": "b2f8ea3d-4b1b-49e0-a141-b9823991ebeb",
 "f4eb1c53-af0a-4dc4-a3a5-c2d50851a178": "d733b0d2-6af3-44e1-8592-e5637fdb76fb"}

Key: "9f4a92b9-5f69-4725-ba1e-403f08dea695"
Corresponding value:
```

Desired Output

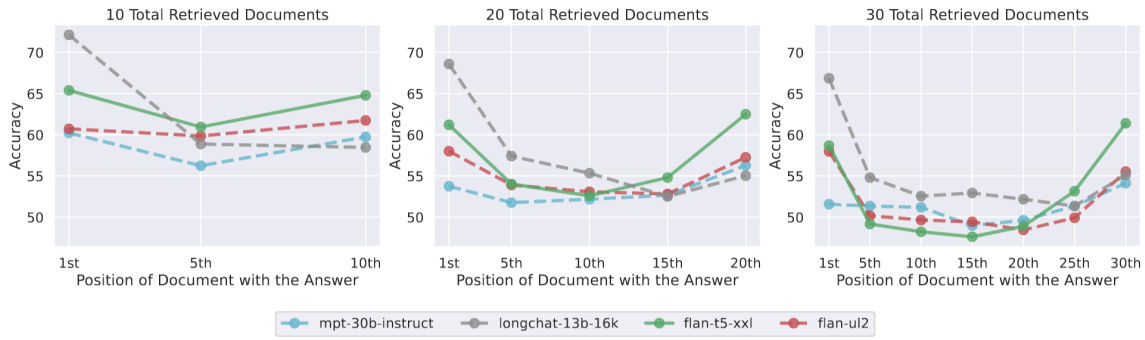
```
703a7ce5-f17f-4e6d-b895-5836ba5ec71c
```



Liu, Nelson F., et al. "Lost in the middle: How language models use long contexts." (2023).

Why LLMs "Lost in the Middle"?

- Model architecture:
  - Compare with the FLAN-T5 (Encoder-Decoder) model.



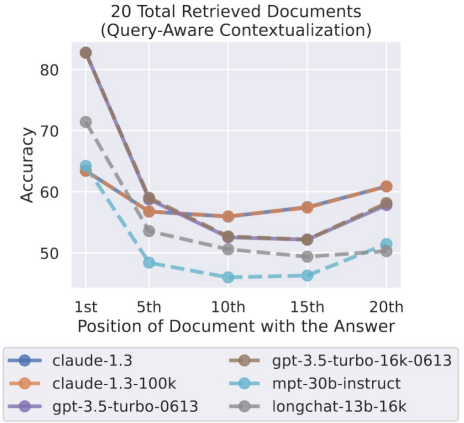


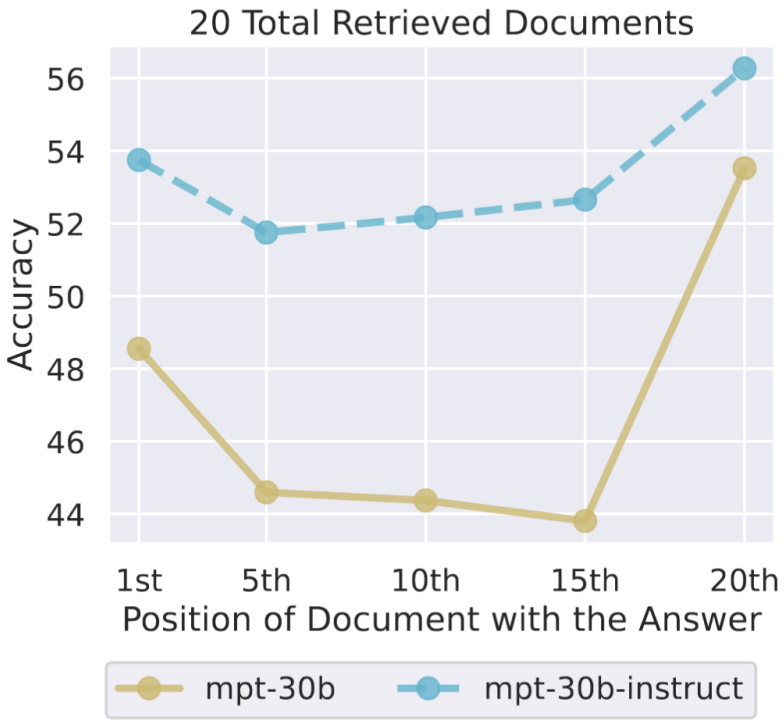
Figure 12: Query-aware contextualization (i.e., placing the question before *and* after the documents in the input context) improves multi-document QA performance when relevant information occurs at the very beginning, but slightly decreases performance otherwise.

  - FLANs suffer less from the "lost in the middle" effect.
  - Intuitively, since the query is put after the context, decoder-only LMs could not attend to the query when read through the context.
  - Put the query at both the beginning and ending of the context could achieve nearly prefect performance.

Liu, Nelson F., et al. "Lost in the middle: How language models use long contexts." (2023).

Why LLMs "Lost in the Middle"?

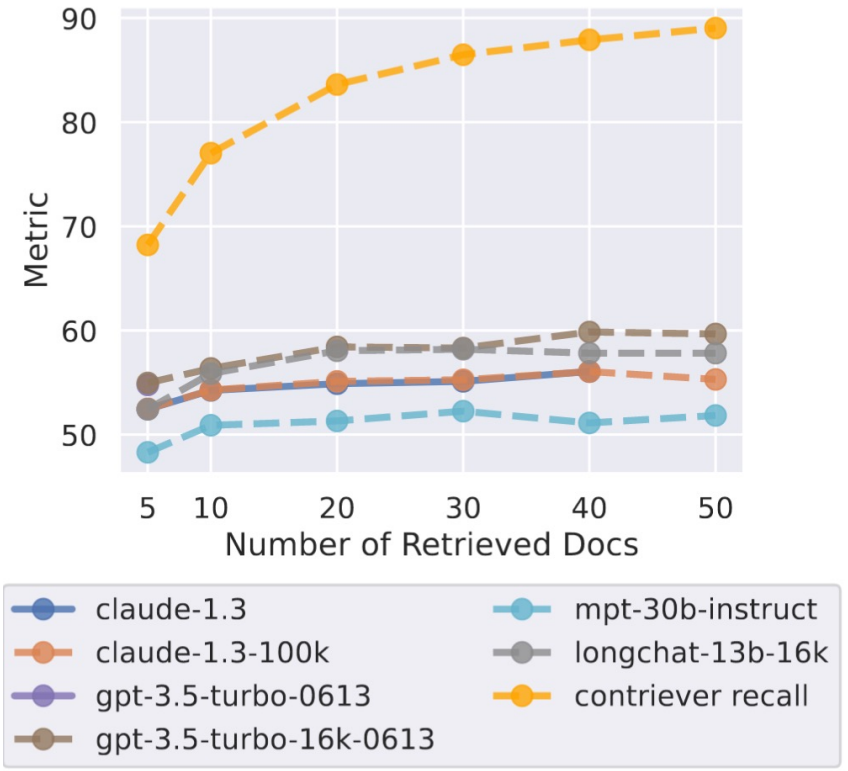
- Effect of instruction tuning:
  - Pre-trained and instruction-tuned LMs exhibit similar U-shape patterns.
  - Contradict to the traditional wisdom about pre-trained LMs that they suffer from recency bias, the result demonstrates that pre-trained LMs could effectively utilize long-range information.
  - Such ability may come from specific pre-training data, e.g., StackOverflow.



Liu, Nelson F., et al. "Lost in the middle: How language models use long contexts." (2023).

The more, the better?

- Experiment with standard retriever-reader setup.
  - A retrieval system takes an input query and return k documents from Wikipedia.
  - Evaluate reader accuracy and retriever recall.
- Using more than 20 documents only marginally improves the accuracy.
- It is a promising research direction for understanding and improving how LMs utilize a long information.



Liu, Nelson F., et al. "Lost in the middle: How language models use long contexts." (2023).

Summary

- With the combination of existing techniques, we can effectively and efficiently expand the context length of pre-trained models.

- However, there are still several remained questions:
  - Whether there are more principled ways to design position encoding.
  - Window attention seems to be an expediency for long context modeling.
  - LLMs do not treat a long context equally, they "lost in the middle".

Thanks!