Cramming 1568 Tokens into a Single Vector and Back Again:

Exploring the Limits of Embedding Space Capacity

Yuri Kuratov[1,2] Mikhail Arkhipov Aydar Bulatov[1,2] Mikhail Burtsev[3]

contact: yurii.kuratov@phystech.edu

Motivation

Embeddings in LLMs are huge:

e.g., LLama-1B: 2,048-dim x 16bit =32,768 bits

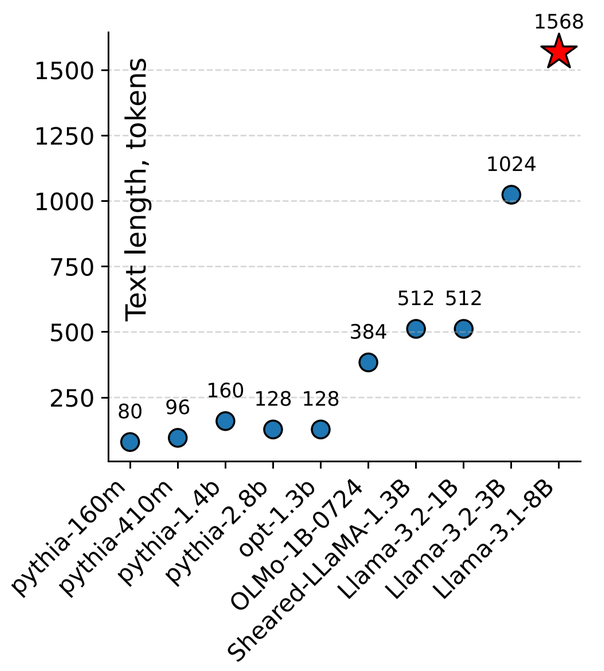
and **one embedding** represents just a **single token**!

But, **one embedding** can encode **many tokens**:

2,048-dim x 16bit = 32,768 bits ~ 1,930 tokens from 128k vocabulary V:

$$\frac{d_{model} \times b}{\log_2 |\mathcal{V}|}$$

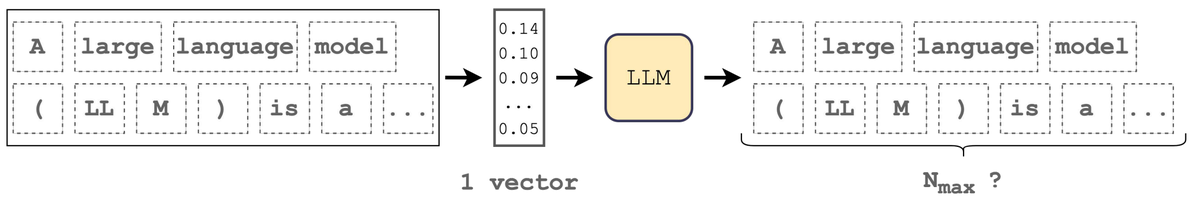
How many tokens fit into a single input vector?



Research Questions

- *Can LMs utilize the latent capacity of input vectors more effectively, potentially encoding and processing multiple tokens with a single vector?*
- *What are achievable limits of input vectors capacity?*

- A **single trainable vector** enables LMs to produce **surprisingly long, targeted text sequences.**
- **Llama-3-8B** reconstructs **1,568 tokens** from a **single vector.**



Method

text: $[t_1,...,t_N]$

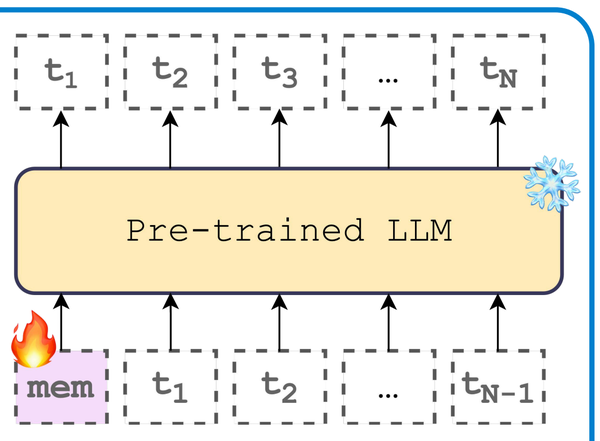
[mem] = $[m_1,...,m_K]$ *in most experiments K = 1*

Per-sample optimization of [mem] vectors:

[mem] is trained to produce text $[t_1,...,t_N]$ token by token:

$$P_{LLM}(t_i \mid [mem], t_1,...,t_{i-1})$$

[mem] vectors are trained for each text separately.



Per-sample optimization is a Prompt Tuning [1] for a single text sample.

Key Metrics and Boundaries

Decoding Capacity (in tokens):

$$L_{\max} = \max\{L \mid \mathrm{Acc}(\mathrm{LM}(t_{[1:L]} \mid [\mathrm{mem}])) > \mathrm{thr}\}$$

Longest text we can reconstruct from [mem] vector.

Token Gain:

$$C_{\text{tokens}} = C^{\text{LM+[mem]}}_{\text{tokens}} - C^{\text{LM}}_{\text{tokens}}$$

Extra tokens recovered only because of [mem], beyond the LM's "knowing of language".

Information Gain:

$$C_{\text{H}} = H_{\text{LM}} - H_{\text{LM+[mem]}}$$

Extra entropy: the same as Token Gain, but measured in entropy, not tokens.

Upper Bound on Token Gain:

$$L \leq \frac{d_{model} \times b}{\log_2 |\mathcal{V}|}$$

V – vocabulary, b - size in bits

Theoretical bound on tokens a single vector can ever encode.

Compression capacity across different text sources

		Pythia-160M	Pythia-410M	Pythia-1.4B	Llama-3.2-1B	Llama-3.2-3B	Llama-3.1-8B
PG-19	Max, tokens	80	96	160	512	1024	1568
	Gain, tokens	70.9±11.0	81.3±12.0	158.0±29.1	426.2±79.2	720.3±80.2	1094.1±127.6
	Information Gain	396.4±46.0	431.4±51.6	792.8±143.4	2119.9±364.8	3292.2±320.0	4865.7±546.6
Fanfics	Max, tokens	80	96	192	512	1024	1568
	Gain, tokens	70.9±10.5	81.2±11.6	152.9±28.0	449.6±83.7	734.1±85.0	1071.8±168.6
	Information Gain	378.1±45.9	429.8±46.2	776.9±132.5	2213.8±365.8	3354.5±344.9	4768.9±622.6
Random	Max, tokens	65	72	139	316	460	792
	Gain, tokens	61.3±6.6	76.9±8.7	144.4±17.5	294.9±64.8	456.9±72.1	623.2±97.3
	Information Gain	500.8±38.9	630.4±65.2	1108.2±136.2	2265.2±498.7	3382.6±585.2	4541.2±758.6

***Information Gain* remains similar across all text sources for each model** (except random for Pythia). For PG-19[2] and fanfics[3], LMs leverage their ability to predict natural language, so the *Decoding Capacity (in Tokens)* generally exceeds the *Token Gain*. Furthermore, we find no evidence that the models benefit from potentially having PG-19 in their pre-training data, as their performance on PG-19 is not significantly better than on fanfics. In contrast, random text offers no predictable structure, making these two metrics nearly identical. This allows us to distinguish how many tokens model can predict by itself compared to decoding from trainable input vector. Larger models consistently show greater compression capacity across all metrics.

PG-19 - natural texts, books [2]

Fanfics - natural texts, never seen by models, published after October 2024 [3]

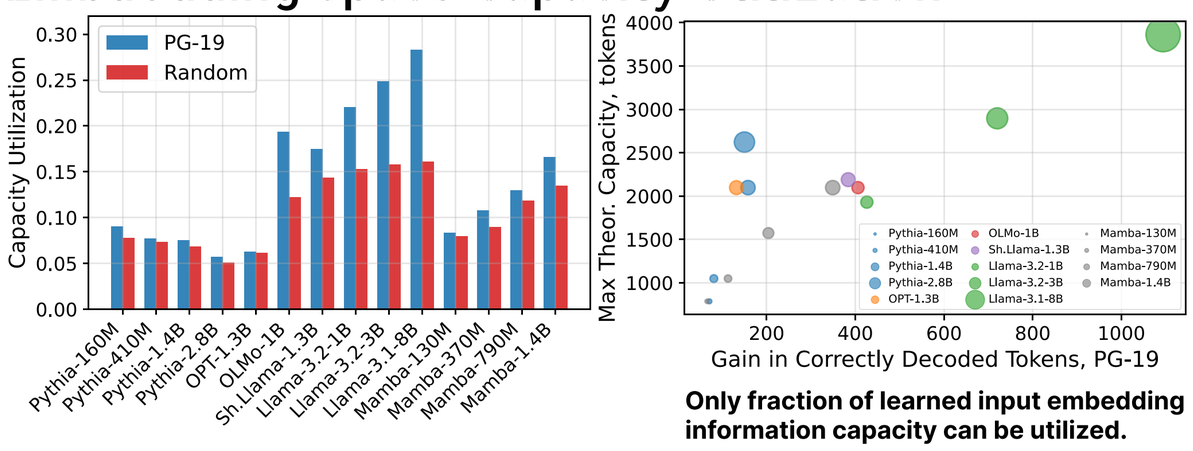
Random - random word sequences from vocab [4]

[1] Lester, Brian, Rami Al-Rfou, and Noah Constant. "The Power of Scale for Parameter-Efficient Prompt Tuning." Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. 2021.

[2] Rae, Jack W., Anna Potapenko, Siddhant M. Jayakumar, Chloe Hillier, and Timothy P. Lillicrap. "Compressive Transformers for Long-Range Sequence Modelling." In International Conference on Learning Representations.

[3] https://archiveofourown.org/ [4] https://nlp.stanford.edu/data/glove.6B.zip

Embdedding Space Capacity Utilization

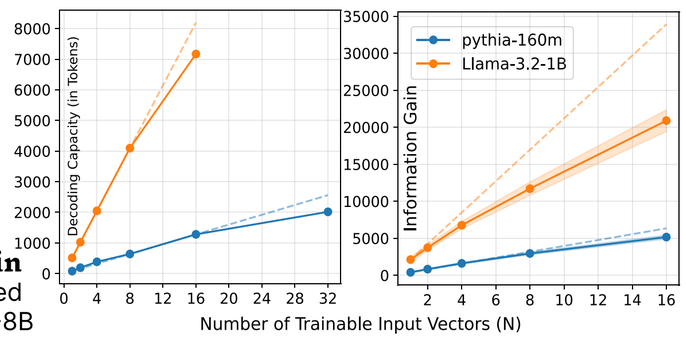


Only fraction of learned input embedding information capacity can be utilized.

Left. Capacity utilization for natural and random texts. Right. Maximum token capacity (see Eq. (1)) against gain in correctly decoded tokens shows differences in utilization of learned memory embedding for studied models.

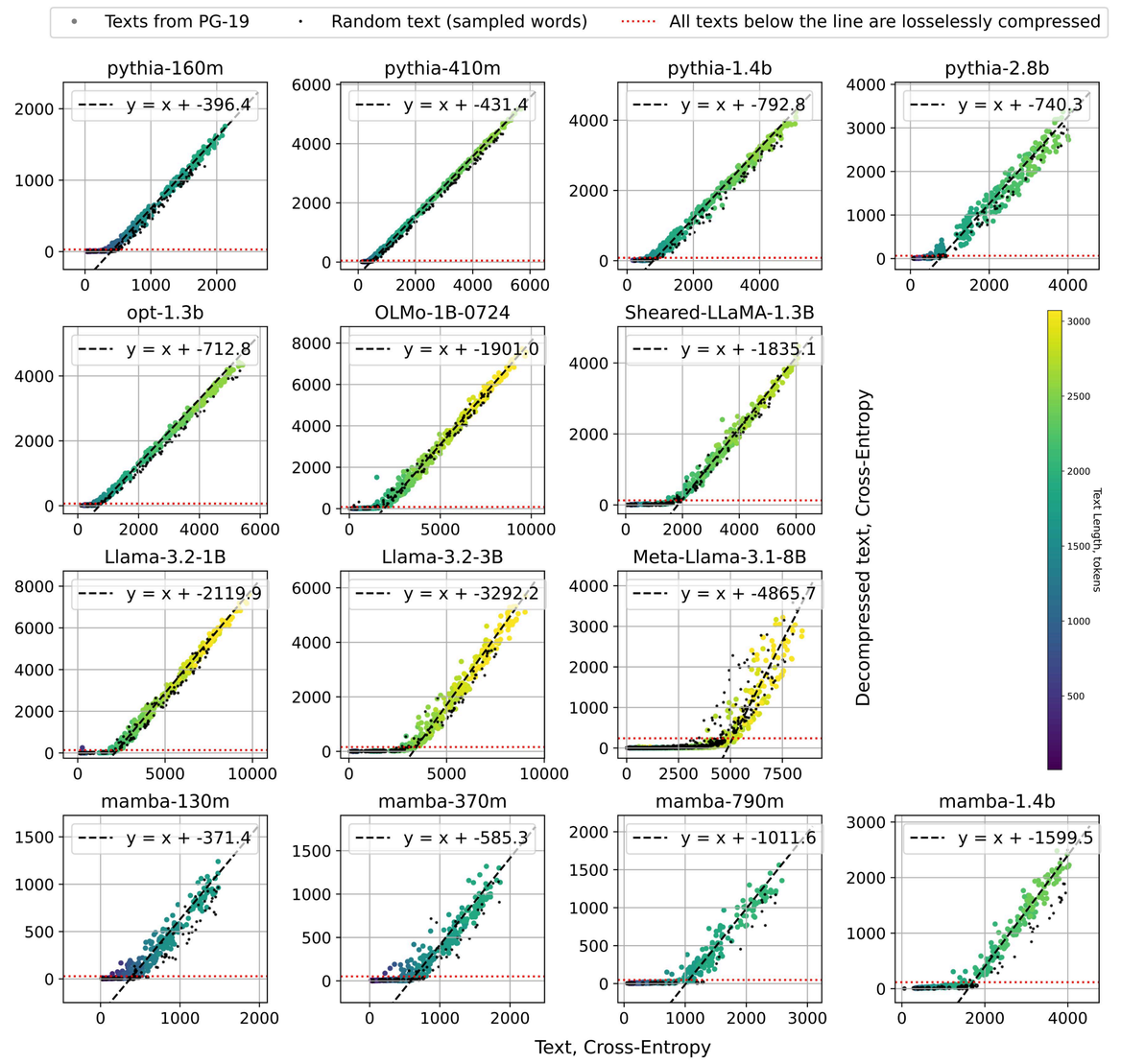
Compression scales linearly with the number of trainable [mem] vectors.

Dashed lines represent ideal linear scaling. Pythia-160m reaches its maximum input context length of 2048 tokens and can successfully encode texts of up to 2016 tokens into 32 [mem] input vectors. Llama-3.2-1B can perfectly decode texts of 7168 tokens from just 16 input vectors.



The Hobbit, or There and Back Again (approx. 120,000 tokens) can be compressed into only 128 input vectors using Llama-3.1-8B and into 256 vectors using Llama-3.2-1B.

Cross-entropy gain



Information gain of text compression to [mem] vector doesn't depend on language understanding capabilities of models. Compression results for various language models show the relationship between the cross-entropy (CE) of the original and decompressed texts. If the text CE falls below a model-specific threshold (red line), the text is losslessly compressed. This value is a input vector capacity in terms of entropy (Information Gain). For texts that are not perfectly compressed, the compression process reduces their CE to a consistent, model-specific value (bias of the black dashed line).

Larger models (e.g., Llama-3.1-8B) can handle longer texts before reaching the compression threshold, due to their greater capacity compared to smaller models (e.g., Pythia-160M). This behavior holds for both natural texts (*PG-19*) and unnatural *random* texts(random word sequences).

Takeaways

- lim(🦙) We show the capacity limits for representations of modern LLMs
- [mem] 🤜 **One** vector packs **1,568 tokens losslessly** in Llama-3.1-8B.
- 📏 Cross-entropy gap directly measures input vector capacity.
- 💡 Capacity stays **consistent across lengths, domains, and architectures***: LMs can decode both **natural** and **random** texts that fit cross-entropy threshold. *Transformers and SSM-based Mamba
- 📈 Capacity **scales near-linear** with [mem] size.
- 🪫 Today's models use **only 10–30%** of theoretical capacity.