MEGABYTE: Predicting Million-byte Sequences with Multiscale Transformers

MITON Times
Radek Bartyzal

Why?

Current state of Transformers:

- Transformers use self-attention
- Self-attention scales quadratically with number of elements
- => use tokenization to reduce the number of elements

Tokenization:

- many forms
- separately trained = form of preprocessing
- generally pain

MegaByte

Idea:

- let’s work directly with bytes of the input
- => ditch tokenization
- problem = $O(n^2)$ scaling with number of bytes

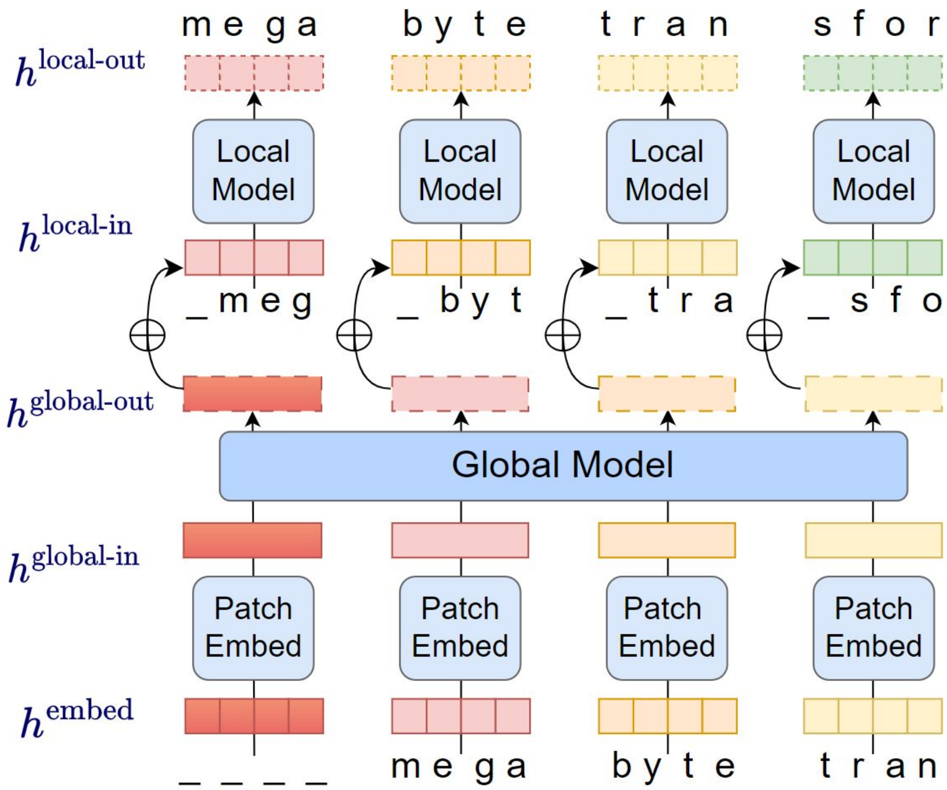
Megabyte solution:

- multi-scale architecture
- global model works with patches of bytes -> local model works on bytes

MegaByte architecture

- patch = 4 bytes
- global model embeds each patch
- local model gets:
  - patch bytes
  - global model embedding of the patch

- local model predicts next byte in its patch



MegaByte architecture

1. embed each byte
2. chunk byte embeddings into K patches of size P = 4
3. global model outputs patch representations = decoder-only Transformer
    - works on K patches
4. local model = smaller decoder-only Transformer
    - works on P elements of a single patch
    - each element = sum of:
        - output from the global model for this patch (global patch representation)
        - embedding of the previous byte in the sequence

MegaByte architecture

Benefits:

- sub-quadratic self-attention = with splits into patches
- per-patch feed-forward layers
    - MEGABYTE uses large feedforward layers per-patch rather than per-position, enabling much larger and more expressive models for the same cost
- parallelization of decoding:
    - generate representation of patches in parallel
    - MEGABYTE model with 1.5B parameters can generate sequences 40% faster than a standard 350M Transformer

Experiments: Comparison to sub-word models

	Tokenizer	Vocab Size	Context Length	Validation	Test
TransformerXL (Rae et al., 2019a)	SentencePiece	32k	512+1024 (subwords)	45.5	36.3
CompressiveTransformer (Rae et al., 2019a)	SentencePiece	32k	512+512+2x512 (subwords)	43.4	33.6
PerceiverAR (Hawthorne et al., 2022)	SentencePiece	32k	2048 (subwords)	45.9	28.9
BlockRecurrent (Hutchins et al., 2022)	SentencePiece	32k	1024+recurrence (subwords)	-	**26.5**
Transformer byte-level (ours)	Bytes	256	2048 (bytes)	81.6	69.4
PerceiverAR byte-level (ours)	Bytes	256	8192 (bytes)	119.1	88.8
MEGABYTE	Bytes	256	8192 (bytes)	**42.8**	36.4

Table 3. Larger scale experiments on PG19, converting bits-per-byte to word-level perplexities for comparison with prior work. Results below the line are compute-matched. MEGABYTE outperforms other byte models by a wide margin, and gives results competitive with state-of-the-art models trained on subwords.

Experiments: Scaling to 1M bytes

	Context	Image64	Image256	Image640
Total len		12288	196608	1228800
Transformer	1024	3.62	3.801	2.847
Perceiver AR	12000	3.55	3.373	2.345
MEGABYTE	Full	**3.52**	**3.158**	**2.282**

Table 5. Bits per byte (bpb) on ImageNet with different resolutions. All models use the same compute and data. MEGABYTE scales well to sequences of over 1M tokens.

Sources

- https://arxiv.org/abs/2305.07185
-