Longformer:
The Long-Document Transformer

최혜원

22.03.29

1. Information

- Transformers have achieved state-of-the-art results in a wide range of natural language tasks including generative language modeling and discriminative language understanding.
- 이런 성공의 이유: self-attention component
  -> enables the network to capture contextual information from the entire sequence.

1. Information

- Self-attention_limit: the memory and computational requirements of self-attention grow **quadratically** with sequence length. -> long sequences 에 부적합
- Solution: Longformer, a modified Transformer architecture with a self-attention operation that scales linearly with the sequence length.
- Advantage for long document classification, question answering(QA) and coreference resolution(대용어해소), where existing approaches partition, or shorten the long context into smaller sequences. (e.g., BERT-style, 512 token limit)

1. Information

- Partitioning -> loss of important information
- Longformer is able to build contextual representations of the entire context using multiple layers of attention, reducing the need for task-specific architectures.
- Longformer’s attention mechanism:
  windowed local-context self-attention + end task motivated global attention that encodes inductive bias about the task.
- Local attention: contextual representations,
  Global attention: full sequence representations.

1. Information

- Autoregressive character-level language modeling
  - windowed + a new dilated attention pattern
  - sequences of up to 32K characters
- Replacing the full self-attention operation of existing pre-trained models.(RoBERTa)
- introducing a variant of Longformer which instead of an encoder-only Transformer architecture, it follows an encoder-decoder architecture similar to the original Transformer model

2. Related Work

- Long-Document Transformers

Model	attention matrix	char-LM	other tasks	pretrain
Transformer-XL (2019)	ltr	yes	no	no
Adaptive Span (2019)	ltr	yes	no	no
Compressive (2020)	ltr	yes	no	no
Reformer (2020)	sparse	yes	no	no
Sparse (2019)	sparse	yes	no	no
Routing (2020)	sparse	yes	no	no
BP-Transformer (2019)	sparse	yes	MT	no
Blockwise (2019)	sparse	no	QA	yes
Our Longformer	sparse	yes	multiple	yes

Table 1: Summary of prior work on adapting Transformers for long documents. ltr: left-to-right.

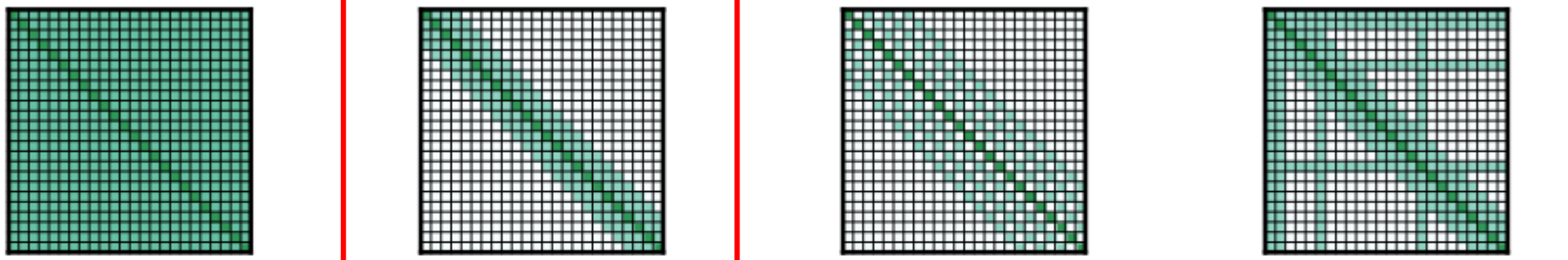
2. Related Work

- Task-specific Models for Long Documents
  - 512 limit of pretrained transformer models like BERT.
  - 주로 사용한 방식:
    - approach chunks the document into chunks of length 512
    - processes each chunk separately
    - two-stage model(the first stage retrieves relevant documents that are passed onto the second stage for answer extraction)
  - Longformer: no truncating, chunking -> concatenates the available context and processes it in a single pass.

3. Longformer

- 기존 transformer - O(n^2) / n= 입력 시퀀스 길이
- Longformer scales linearly with the input sequence, making it efficient for longer sequences according to an “attention pattern".

3.1 Attention Pattern

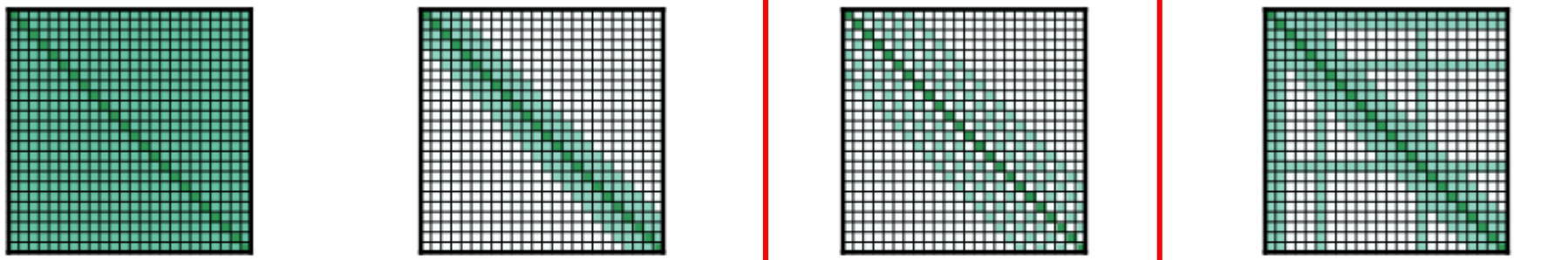


(a) Full n^2 attention (b) Sliding window attention (c) Dilated sliding window (d) Global+sliding window

Figure 2: Comparing the full self-attention pattern and the configuration of attention patterns in our Longformer.

- Sliding Window: fixed-size(w) window attention surrounding each token. Each token attends to 1/2w tokens on each side.
- O(nxw), n 의 scale 은 선형적으로 증가한다.
- Transformer with **l** layers, top layer is **l x w** .
- Depending on the application, w might be helpful to use different between efficiency model representation capacity.

3.1 Attention Pattern

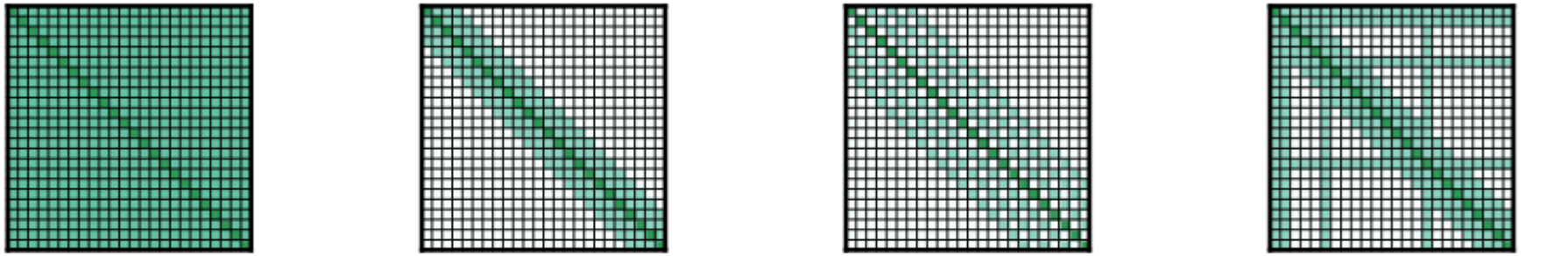


(a) Full n^2 attention (b) Sliding window attention (c) Dilated sliding window (d) Global+sliding window

Figure 2: Comparing the full self-attention pattern and the configuration of attention patterns in our Longformer.

- **Dilated Sliding Window**: the window has gaps of size dilation *d*
- Assuming a fixed d and w for all layers, the receptive field is l × d × w, which can reach tens of thousands of tokens even for small values of d.
- We found settings with different dilation configurations per head improves performance.

3.1 Attention Pattern



(a) Full n^2 attention (b) Sliding window attention (c) Dilated sliding window (d) Global+sliding window

Figure 2: Comparing the full self-attention pattern and the configuration of attention patterns in our Longformer.

- Global Attention
  - MLM: local context to predict the masked word
  - classification: the model aggregates the representation of the whole sequence into a special token ([CLS] in case of BERT)
  - QA: question / document 비교
  - We add “global attention” on few pre-selected input locations.
  - Operation symmetric, a token across the sequence and all tokens in the sequence attend to it.
  - O(n) :the number of such tokens is small relative to and independent of n

3.1 Attention Pattern

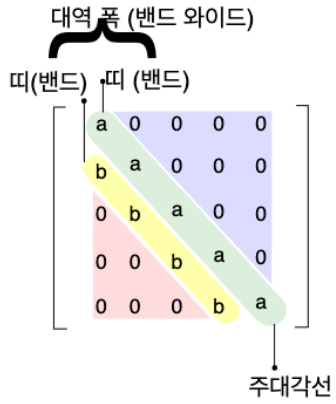
- Linear Projections for Global Attention.
- In transformer model, computing attention scores.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (1)$$

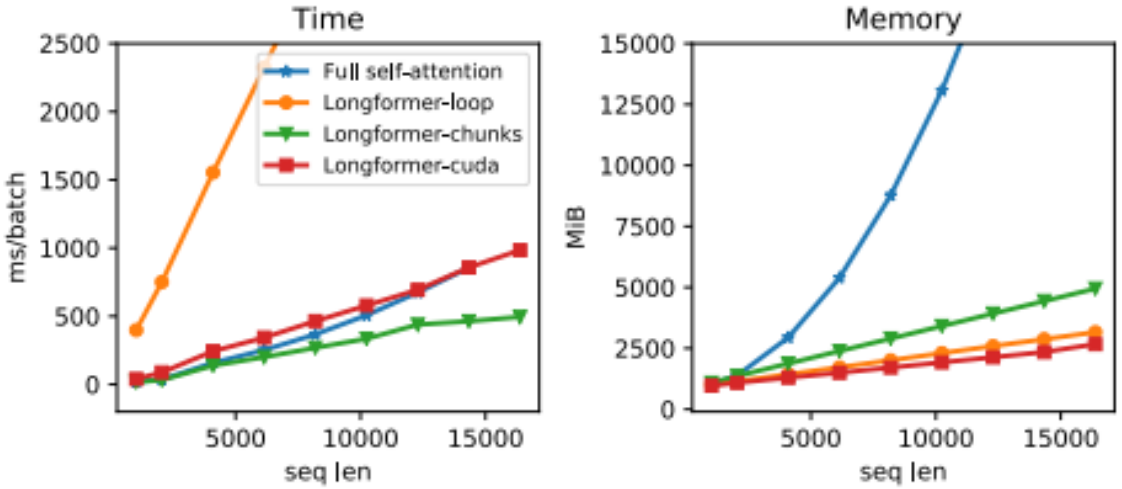
- Sliding window attention - Q_s, K_s, V_s
- Global attention - Q_g, K_g, V_g

3.2 Implementation

- The expensive operation is the matrix multiplication QK^t, because both Q and K have n(sequence length) projections.
- Longformer, the dilated sliding window attention computes only a fixed number of the diagonals of QK^T. -> n제곱 연산
- However, implementing it requires a form of banded matrix multiplication that is not supported in existing deep learning libraries like PyTorch/Tensorflow.



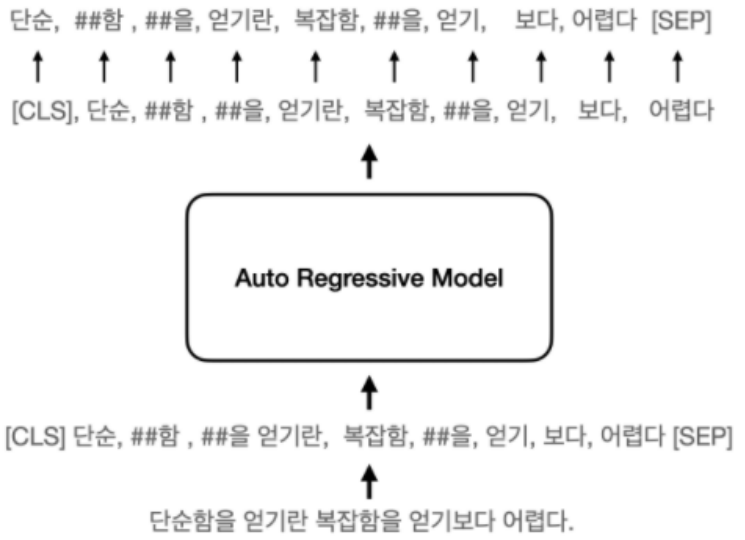
3.2 Implementation



- Loop, chunks, cuda 세 그래프 비교: self attention 다른 방식으로 연산 수행 시 runtime 과 memory 의 사용을 나타냄.
- Cuda is our fully functioning highly optimized custom CUDA kernel implemented using TVM

4. Autogressive Language Modeling

- Autoregressive or left-to-right language modeling is loosely defined as estimating the probability distribution of an existing token/character given its previous tokens/characters in an input sequence.- 이전 토큰/문자가 주어졌을 때 현재 문자의 확률 분포를 예측하는 것.



4.1 Attention Pattern

- Use dilated sliding window attention.
- 낮은 레이어에는 작은 window size -> to capture local information
- 높은 레이어는 window 사이즈 증가. -> to learn higher-level representation
- For the higher layers, we use a small amount of increasing dilation only on 2 heads.

4.2 Experiment Setup

- Training: a staged training procedure where we increase the attention window size and sequence length across multiple training phases.
- 시작은 short sequence length and window size -> double the window size and sequence length -> halve the learning rate.
- 모델 학습은 5단계. 시퀀스 길이 2048 -> 23040

4.2.1 Results

Model	#Param	Dev	Test
Dataset `text8`			
T12 (Al-Rfou et al., 2018)	44M	-	1.18
Adaptive (Sukhbaatar et al., 2019)	38M	1.05	1.11
BP-Transformer (Ye et al., 2019)	39M	-	1.11
Our Longformer	41M	1.04	**1.10**
Dataset `enwik8`			
T12 (Al-Rfou et al., 2018)	44M	-	1.11
Transformer-XL (Dai et al., 2019)	41M	-	1.06
Reformer (Kitaev et al., 2020)	-	-	1.05
Adaptive (Sukhbaatar et al., 2019)	39M	1.04	1.02
BP-Transformer (Ye et al., 2019)	38M	-	1.02
Our Longformer	41M	1.02	**1.00**

Table 2: *Small* model BPC on `text8` & `enwik8`

Model	#Param	Test BPC
Transformer-XL (18 layers)	88M	1.03
Sparse (Child et al., 2019)	≈100M	0.99
Transformer-XL (24 layers)	277M	0.99
Adaptive (Sukhbaatar et al., 2019)	209M	0.98
Compressive (Rae et al., 2020)	277M	0.97
Routing (Roy et al., 2020)	≈223M	0.99
Our Longformer	102M	0.99

Table 3: Performance of *large* models on `enwik8`

5. Pretraining and Finetunig

- Pretrained Longformer on a document corpus and finetune it for 6 task(classification, QA, coreference resolution)
- 4096 tokens (8times longer than BERT)
- Pretrain Longformer with masked language modeling (MLM), where the goal is to recover randomly masked tokens in a sequence.
- -> pretraing from RoBERTa released checkpoint.
- Note that our attention pattern can be plugged into any pretrained transformer model without the need to change the model architecture

5. Pretraining and Finetunig

- **Attention pattern**: window size 512인 sliding window attention ->RoBERTa와 같은 계산량
- **Position Embeddings**: RoBERTa 최대 512 -> 4096, initialize position embeddings by copying the 512 position embeddings from RoBERTa multiple times

Model	base	large
RoBERTa (seqlen: 512)	1.846	1.496
Longformer (seqlen: 4,096)	10.299	8.738
+ copy position embeddings	1.957	1.597
+ 2K gradient updates	1.753	1.414
+ 65K gradient updates	1.705	1.358
Longformer (train extra pos. embed. only)	1.850	1.504

Table 5: MLM BPC for RoBERTa and various pre-trained Longformer configurations.

5. Pretraining and Finetunig

- Continued MLM Pretraining
- 65k gradient updates with sequences length 4096,
- batchsize 64(2^18 tokens)
- maximum learning rate of 3e-5
- linear warmup of 500 steps
- a power 3 polynomial decay (learning rate schedule)
- 나머지는 RoBERTa 와 동일.

6. Tasks

Wordpieces	WH	TQA	HQA	ON	IMDB	HY
avg.	1,535	6,589	1,316	506	300	705
95th pctl.	3,627	17,126	1,889	1,147	705	1,975

Table 6: Average and 95th percentile of context length of datasets in wordpieces. WH: WikiHop, TQA: TriviaQA, HQA: HotpotQA, ON: OntoNotes, HY: Hyperpartisan news

- Evaluation datasets have contexts significantly longer than 512 wordpieces.
- The Longformer variant replaces the RoBERTa self-attention mechanism with our windowed attention used during pretraining, plus a task motivated **global attention**.

6. Tasks

- **QA**: WikiHop, TriviaQA, HotpotQA
- concatenate question and documents into one long sequence, run it through Longformer, then have a dataset-specific prediction layer.
- WikiHop uses a classification layer for the candidate
- TriviaQA uses the loss function Clark and Gardner to predict answer span.
- We include global attention to question tokens and answer candidates for WikiHop and to question tokens for TriviaQA.

6. Tasks

- **QA**: WikiHop, TriviaQA, HotpotQA
- HotpotQA is a multihop QA dataset that involves extracting answer spans and evidence sentences from 10 Wikipedia paragraphs, 2 of which are relevant and the rest are distractors.
- We train the models in a multi-task way to predict relevant paragraphs, evidence sentences, answer spans and question types (yes/no/span) jointly

6. Tasks

- Document Classification: IMDB
- IMDB is a standard sentiment classification datasets consisting of movie reviews.
- We use global attention on the [CLS] token.