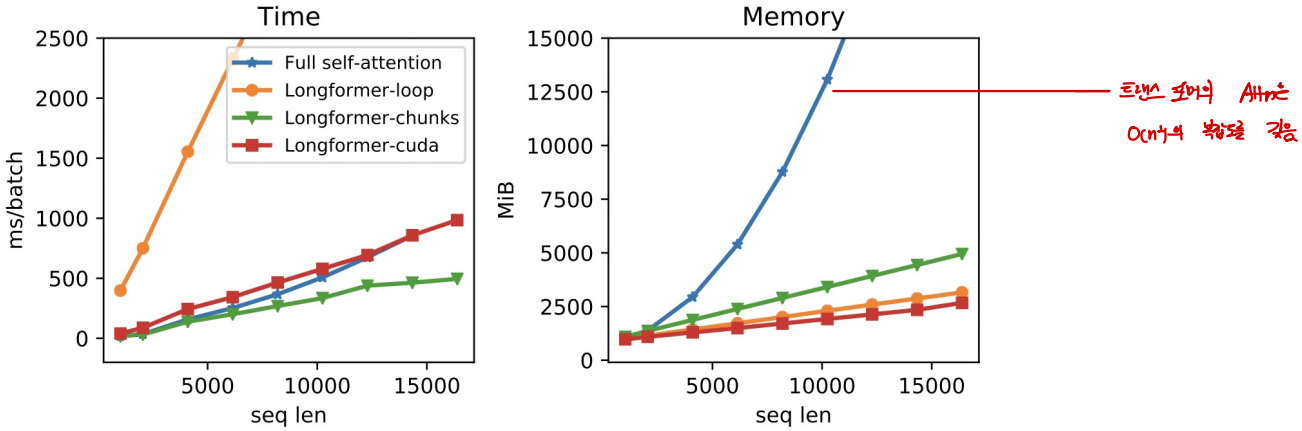


트랜스포머의 Attn은 O(n²)의 복잡도를 갖음

∴ 계산량, 시간, 메모리 소모↑

∴ 긴문장이 주어졌을때 Bert 같은경우 Truncated_Seq를 사용하게 되는데 일반적으로 512개의 토큰 리밋이 있다.

⇒ 512개의 제한과, 꼬리짤 때문에 문맥파악이 어려울수 있다.

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.

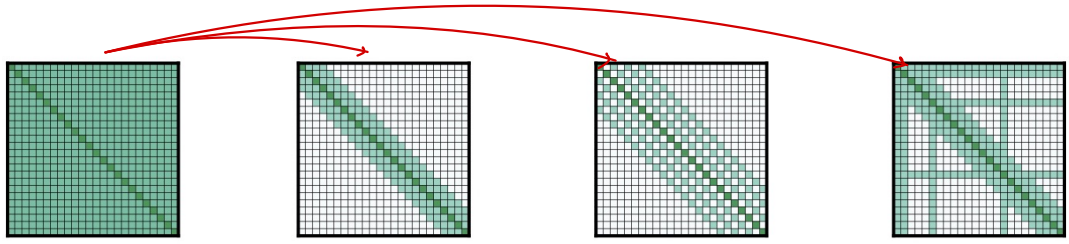
< 기 존 연 구 >

long Document는 : ltr : 긴 텍스트를 left → right 청크 단위로 임베딩. / Bidirectional 방법을 적용하는것에선 약함.

Sparse Attn 방식이 있다 : General한 방식/ Attn 방법도 맞춤.

⇒ Long Former.

복잡도를 줄이기 위한 방법들.

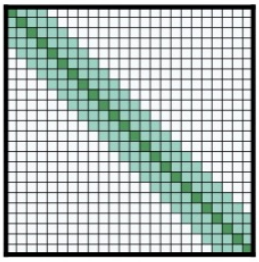


(a) Full n^2 attention (b) Sliding window attention (c) Dilated sliding window (d) Global+sliding window

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

Long Former는 문장이 길더라도 텍스트를 고려한 whole Contextual representation을 활용하면서 모델 아키텍처이 의존하지 않고 성능을 높일 수 있다.

1) Sliding Window.



(b) Sliding window attention

: 특정 근처에 있는 토큰들에 대해 Attn을 계산. 멀리떨어진 토큰에서는 계산하지 x.

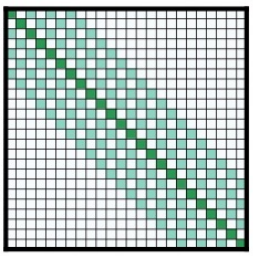
W: fixed window size.

⇒ Each Token attends $\frac{1}{2}w$ tokens on each side.

Computational Cost : $O(n \times w)$.

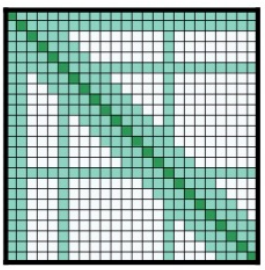
Receptive field : $l \times w$

2).



(c) Dilated sliding window

- 위의 슬라이딩 방법보다 좀더 효율적이고, 보통 컨텍스트를 좀더 확장하여 만든 방식 (더 넓은 영역)
- Receptive field : $l \times D \times w$,
- 특히 Multi-head Attn에서 서로다른 dialated 를 사용해서 Attn Score를 계산
  - Dialation이 없는 애들은 좀더 지역적인 애들을
  - Dalation이 있는 애들은 긴 문맥 정보를 처리함



(d) Global+sliding window

* 1). 위는 local Context 만 바라보기 때문에 컨텍스트에는 부족한 부분이 생김.
* 버트처럼 모델들은 문제들을 풀때 [sep] 같이 스페셜 토큰을 사용하는데 [CLS] 이러한 토큰들은 전체적인 문맥에 대해서 어텐션하므로 학습을 유리하게 만듦. (?!)

따라서 글로벌 어텐션 방식에서는 미리 선택해둔 스페셜 토큰위치에 대해서 전체 토큰에 대해서 어텐션 할 수 있도록 하였다.

스페셜 토큰의 개수가 윈도우 어텐션에 비해 상당히 적기 때문에 몇몇 토큰에 대해 글로벌 어텐션을 적용하는것은 계산에 있어서 무리가 안된다.

+ 복잡도 $O(n)$.

* 어떤 스페셜 토큰에 대해서만 풀 어텐션을 한다.

[CLS], [ ]

↓

스페셜 토큰에 대한 명확한 명명을 알려주지 않는다.

* Dialated → 레이어를 층층이 쌓는데 밑에는 촘촘한 정보 위로 갈수록 윈도우 사이즈를 크게..

CNN 같은데?

→ 밑 레이어 에서는, Window 내에서 fully attention을 하되.

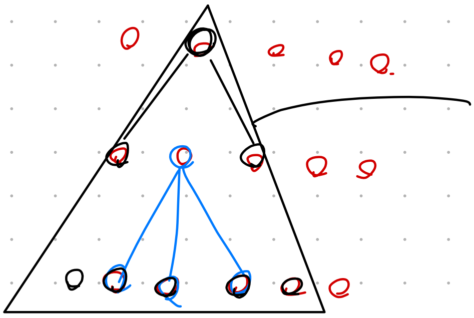
위로 갈수록 dialation을 준다.

영어 Ver.

* Transformer는 매우 harsh한 제약조건이 있어서 긴 문서는 문장 단위로 끊어서 봐야 함.

⇒ 각 문장은 독립적이 되어 버린다.

*. 예를 들어보자. (dilated).



위로 갈수록 더 넓게 정보를 획득 함.

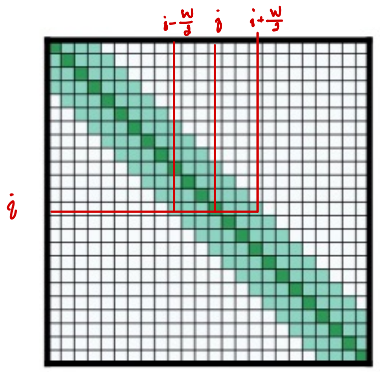
⇒. 낮은 차원에서는 local 한 정보를 얻는다. 반면 위로 갈수록 좀더 넓은 정보를 얻음.

let's say kernel size is 3.

* CNN과 비슷한 느낌인데 생각해보면 make sense 함.
왜냐하면 하나의 단어 주변에 있는 애들이 서로에게 중요한 역할을 하겠지.

* CNN에서도 되는것이 transformer에서도 된다.

* $O(n \cdot w)$
↑ window size.



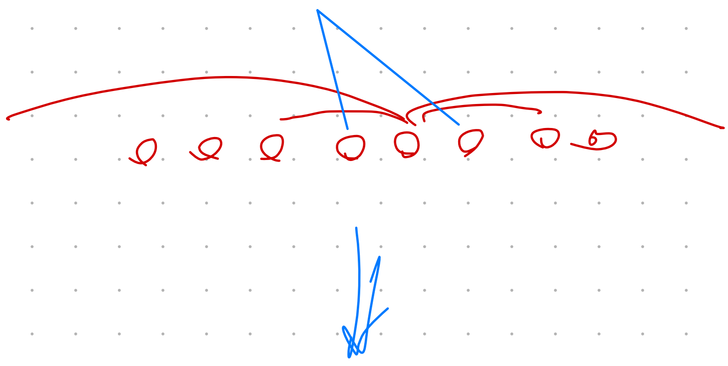
(b) Sliding window attention

위에서 말한 CNN 같은것은. 1개의 layer를 올라가면, 2w 만큼 context를 얻는다.

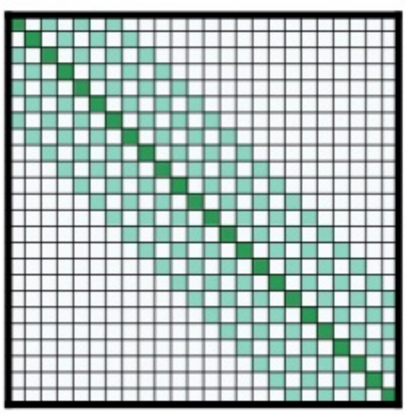
NLP는 주변 정보가 중요한거...

↕

Oh well! 주변 정보 안중요

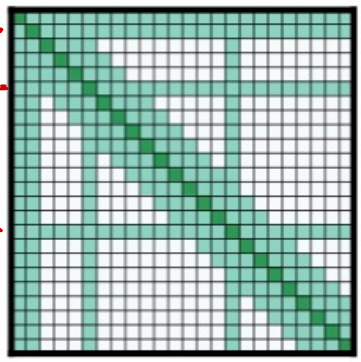


그래서 낮은 layer에서 (b) 쓰고.
높은 층에서 (c) 쓰자.



(c) Dilated sliding window

Those are Special Tokens to attend all units.



(d) Global+sliding window

여기는 Special Token이 있다고 함.