Transformer-XL
Attentive Language Models

Beyond a Fixed-Length Context

Zihang Dai et al.

Abstract

트랜스포머는 language modeling 세팅에서 (sequence를 한 번에 보기 때문에) Fixed-length context로 제한되는 것이 있었다. (512가 안되는 문장은 패딩을 하고, 512를 넘는 문장은 자름)

Transformer-XL이라는 아키텍쳐를 제안한다.
- 바닐라 트랜스포머가 가진 longer-term dependency를 캐치할 수 있을 뿐만 아니라,
- Segment끼리의 정보가 전달되지 않는, context fragmentation problem까지 해결했다.

Results

Transformer XL은 5가지 데이터셋에서 SOTA를 달성했다.
기존 RNN과 바닐라 Transformer 대비 80~450%의 향상이 있었으며, 바닐라 Transformer보다 1800배 빠르다.

Method	enwiki8	text8	One Billion Word	WT-103	PTB (w/o finetuning)
Previous Best	1.06	1.13	23.7	20.5	55.5
Transformer-XL	**0.99**	**1.08**	**21.8**	**18.3**	**54.5**

Results

Tranformer-XL의 Attention score

Model	#Param	PPL
Grave et al. (2016b) - LSTM	-	48.7
Bai et al. (2018) - TCN	-	45.2
Dauphin et al. (2016) - GCNN-8	-	44.9
Grave et al. (2016b) - LSTM + Neural cache	-	40.8
Dauphin et al. (2016) - GCNN-14	-	37.2
Merity et al. (2018) - QRNN	151M	33.0
Rae et al. (2018) - Hebbian + Cache	-	29.9
Ours - Transformer-XL Standard	151M	**24.0**
Baevski and Auli (2018) - Adaptive Input°	247M	20.5
Ours - Transformer-XL Large	257M	**18.3**

Table 1: Comparison with state-of-the-art results on WikiText-103. ° indicates contemporary work.

Model	#Param	bpc
Ha et al. (2016) - LN HyperNetworks	27M	1.34
Chung et al. (2016) - LN HM-LSTM	35M	1.32
Zilly et al. (2016) - RHN	46M	1.27
Mujika et al. (2017) - FS-LSTM-4	47M	1.25
Krause et al. (2016) - Large mLSTM	46M	1.24
Knol (2017) - cmix v13	-	1.23
Al-Rfou et al. (2018) - 12L Transformer	44M	1.11
Ours - 12L Transformer-XL	41M	**1.06**
Al-Rfou et al. (2018) - 64L Transformer	235M	1.06
Ours - 18L Transformer-XL	88M	1.03
Ours - 24L Transformer-XL	277M	**0.99**

Table 2: Comparison with state-of-the-art results on enwik8.

Model	#Param	bpc
Cooijmans et al. (2016) - BN-LSTM	-	1.36
Chung et al. (2016) - LN HM-LSTM	35M	1.29
Zilly et al. (2016) - RHN	45M	1.27
Krause et al. (2016) - Large mLSTM	45M	1.27
Al-Rfou et al. (2018) - 12L Transformer	44M	1.18
Al-Rfou et al. (2018) - 64L Transformer	235M	1.13
Ours - 24L Transformer-XL	277M	**1.08**

Table 3: Comparison with state-of-the-art results on text8.

Model	#Param	PPL
Shazeer et al. (2014) - Sparse Non-Negative	33B	52.9
Chelba et al. (2013) - RNN-1024 + 9 Gram	20B	51.3
Kuchaiev and Ginsburg (2017) - G-LSTM-2	-	36.0
Dauphin et al. (2016) - GCNN-14 bottleneck	-	31.9
Jozefowicz et al. (2016) - LSTM	1.8B	30.6
Jozefowicz et al. (2016) - LSTM + CNN Input	1.04B	30.0
Shazeer et al. (2017) - Low-Budget MoE	~5B	34.1
Shazeer et al. (2017) - High-Budget MoE	~5B	28.0
Shazeer et al. (2018) - Mesh Tensorflow	4.9B	24.0
Baevski and Auli (2018) - Adaptive Input°	0.46B	24.1
Baevski and Auli (2018) - Adaptive Input°	1.0B	23.7
Ours - Transformer-XL Base	0.46B	23.5
Ours - Transformer-XL Large	0.8B	**21.8**

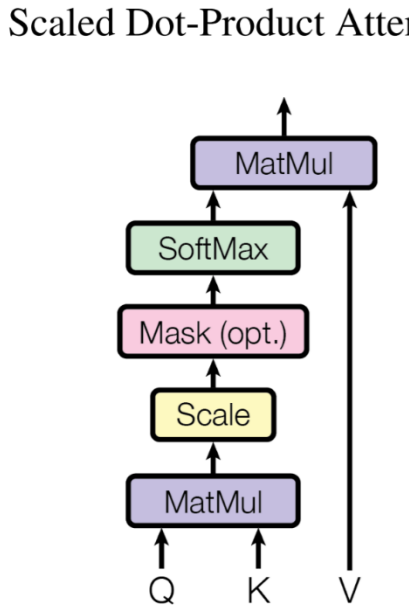
Table 4: Comparison with state-of-the-art results on One Billion Word. ° indicates contemporary work.

Deep Self-Attention Network

Attention은 Query(Q), Key(K), Value(V) pair – 모든 componen는 vecto이다 – 를 outpu으로 mappin하는 작업이다. Value를 모두 weighted sum (각 데이터에 가중치를 주는 것)한다.

이때 value에 적용되는 weigh는 Q와 K간 연관성이 얼마나 크냐, 얼마나 관련성이 높냐(확률)를 측정해서 계산된다. 이러한 과정을 scaled dot-product attention이라고 한다.

(모든 Key에 대해 Query를 내적하고, 각 결과값을 root-d_k로 나눈 다음, softmax를 적용해서 weight 계산)

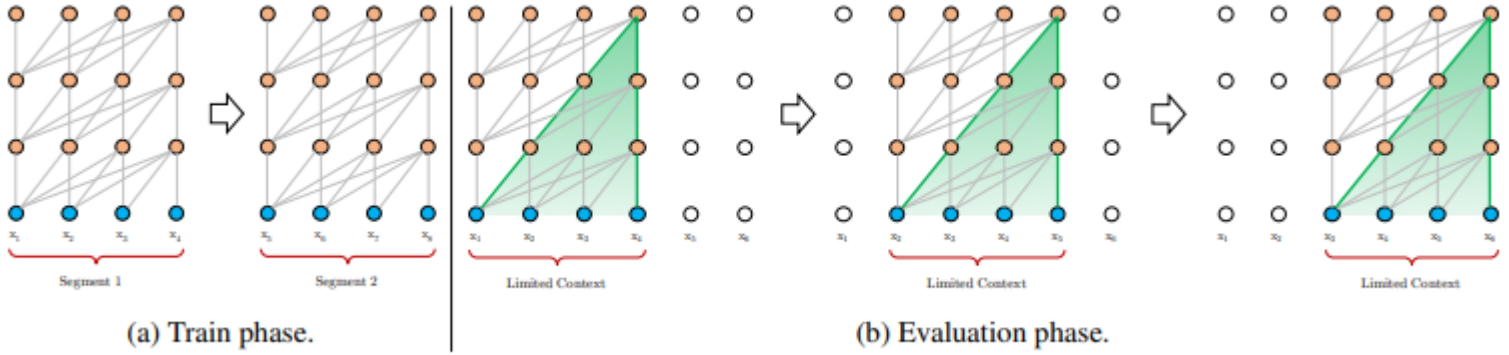


Vanilla Transformer

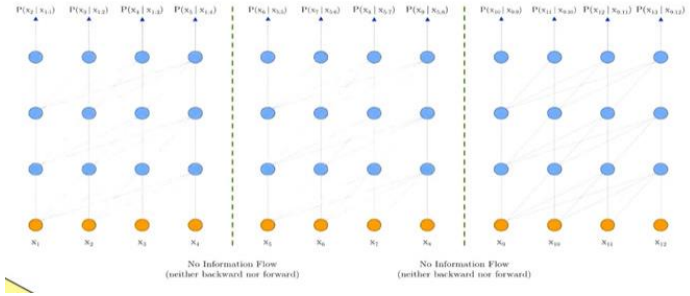
전체 sequence를 한 번에 보기 때문에 고정된 길이(512)어야만 했고, 더 긴 길이의 sequence 학습을 못했다.
문장이 길면 잘라서 넣어야 했고(absolute positional embedding), 때문에 정보가 끊어지는 문제가 발생했다.

기존의 모델은 정해진 길이로 segment를 4개로 쪼갠 다음, 1~4/5~8 segmen를 훈련시켰다.
또한 예측시에는 고정길이 4씩 sliding을 하는데, 1~4로 5번째의 것을 예측하고, 2~5로 6을 예측한다.

Context fragmentation Problem
각 segments간에 정보가 전달되지 않는다.



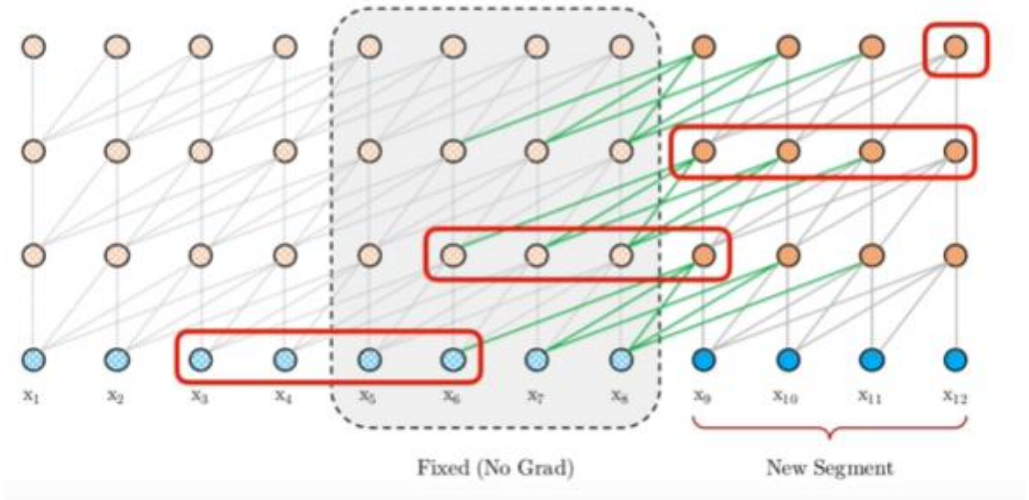
(a) Train phase. (b) Evaluation phase.

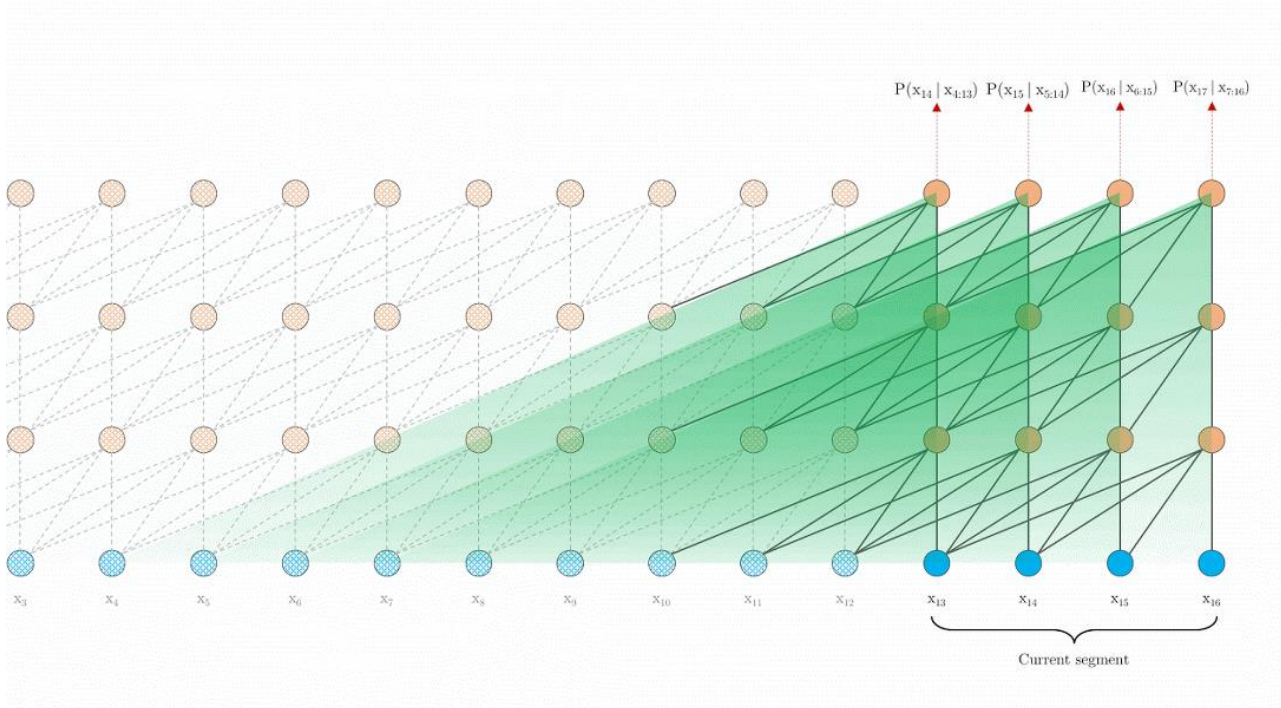


Transformer-XL Model

- recurrence를 활용해서 fixed-length를 해결하고,
- relative positional embeddings를 사용해서 context fragmentation proble을 해결하자

Segment를 학습할 때, 이전 segment를 계산할 때 사용한 state를 캐시로 갖고 있다.
(이는 속도 향상에도 기여했다.)





Transformer-XL Model

- recurrence를 활용해서 fixed-length를 해결하고,
- relative positional embeddings를 사용해서 context fragmentation proble을 해결하자

Segment를 학습할 때, 이전 segment를 계산할 때 사용한 state를 캐시로 갖고 있다.
(이는 속도 향상에도 기여했다.)

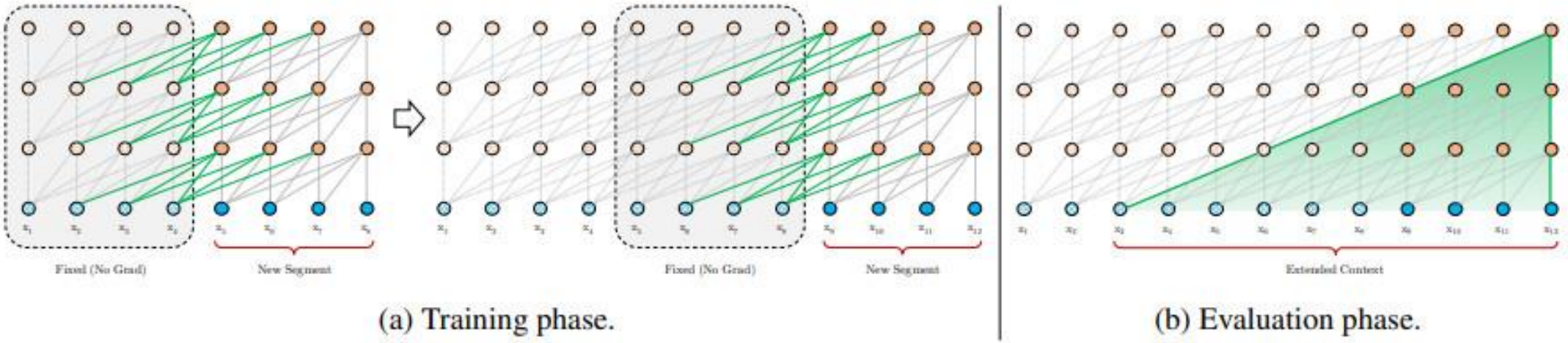


Figure 2: Illustration of the Transformer-XL model with a segment length 4.

Relative Positional Encoding

기존의 absolute positional encoding은 여러 segment에 대해서 recurrence를 학습시키기가 어렵다.

- Attention score in standard Transformer

$$\mathbf{A}_{ij}^{abs} = \underbrace{\mathbf{E}_{x_i}^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{E}_{x_j}}_{(a)} + \underbrace{\mathbf{E}_{x_i}^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{U}_j}_{(b)} + \underbrace{\mathbf{U}_i^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{E}_{x_j}}_{(c)} + \underbrace{\mathbf{U}_i^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{U}_j}_{(d)}$$

- Attention score with Relative Positional Encoding

$$\mathbf{A}_{ij}^{rel} = \underbrace{\mathbf{E}_{x_i}^\top \mathbf{W}_q^\top \mathbf{W}_{k,E} \mathbf{E}_{x_j}}_{(a)} + \underbrace{\mathbf{E}_{x_i}^\top \mathbf{W}_q^\top \mathbf{W}_{k,R} \mathbf{R}_{i-j}}_{(b)} + \underbrace{u^\top \mathbf{W}_{k,E} \mathbf{E}_{x_j}}_{(c)} + \underbrace{v^\top \mathbf{W}_{k,R} \mathbf{R}_{i-j}}_{(d)}$$

(a) : 콘텐츠 기반
(b) : 콘텐츠에 의존한 콘텐츠 bias
(c) : 글로벌 콘텐츠 bias
(d) : 글로벌 positional bias

: relative distance라서 음수가 없다
언제든지 여기에서 절대 위치를 얻을 수 있기 때문에, temporal information을 잃지 않는다.

Relative Positional Encoding

기존의 absolute positional encoding은 여러 segment에 대해서 recurrence를 학습시키기가 어렵다.

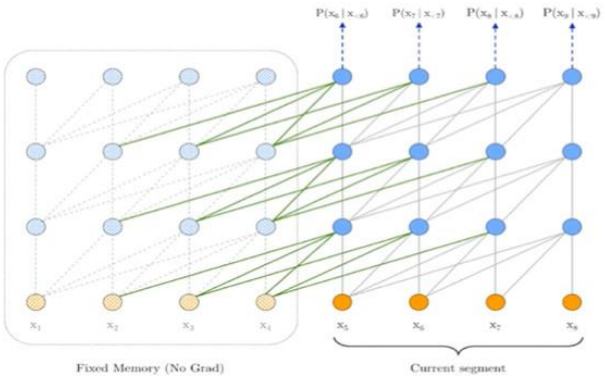
- Attention score in standard Transformer

$$\mathbf{A}_{ij}^{abs} = \underbrace{\mathbf{E}_{x_i}^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{E}_{x_j}}_{(a)} + \underbrace{\mathbf{E}_{x_i}^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{U}_j}_{(b)} + \underbrace{\mathbf{U}_i^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{E}_{x_j}}_{(c)} + \underbrace{\mathbf{U}_i^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{U}_j}_{(d)}$$

- Attention score with Relative Positional Encoding

$$\mathbf{A}_{ij}^{rel} = \underbrace{\mathbf{E}_{x_i}^\top \mathbf{W}_q^\top \mathbf{W}_{k,E} \mathbf{E}_{x_j}}_{(a)} + \underbrace{\mathbf{E}_{x_i}^\top \mathbf{W}_q^\top \mathbf{W}_{k,R} \mathbf{R}_{i-j}}_{(b)} + \underbrace{u^\top \mathbf{W}_{k,E} \mathbf{E}_{x_j}}_{(c)} + \underbrace{v^\top \mathbf{W}_{k,R} \mathbf{R}_{i-j}}_{(d)}$$

(a) : 콘텐츠 기반
(b) : 콘텐츠에 의존한 콘텐츠 bias
(c) : 글로벌 콘텐츠 bias
(d) : 글로벌 positional bias



References

1. PR-161: Transformer-XL (https://www.youtube.com/watch?v=lSTljZy8ag4)
2. Transformer-XL Explained (링크)
3. https://ai.googleblog.com/2019/01/transformer-xl-unleashing-potential-of.html
4. https://novdov.github.io/machnielearning/nlp/2019/07/13/XLNet-%EB%A6%AC%EB%B7%B0/
5. https://ratsgo.github.io/natural%20language%20processing/2019/09/11/xlnet/