Transformer XL

김재민, 신성진

Contents

TransformerXL - Introduction

언어에 대한 **Long-Term Dependency**는 항상 단골로 나오는 이슈

RNN의 **LSTM**계열: 평균 약 **200**자 정도 유지 **(Khandelwal et al. 2018)**

Attention을 통해 연관 단어의 직접적인 연결을 통해 문제 해결

Char-Transfomer모델로 **(Al-Rfou et al. 2018) LSTM**보다 좋은 성능도 냄

But, 기존 모델은

1.고정된 길이 **Segment**

2. 사전 정의된 컨텍스트 길이를 초과하는 장기의존성 캡쳐 불가

TransformerXL - Introduction

+ **Context fragment: Long-term Dependency 해결 X**

  고정된 길이의 **Segments:** 문장이나 다른 의미 경계를 고려하지 않고 연속적인 기호 덩어리들을 생성

  상황 정보가 부족하여 비효율적인 최적화 및 성능 저하를 초래

TransformerXL - Introduction

Transformer XL 제안

Gradient vanishing explosion을 해결하기 위해 제안

이전 **Segment**의 **Hidden**을 재 사용하여 **(**반복에 대한 메모리 역할**)** 문제를 해결

이전 **Segment**의 활용을 위한 기존의 **absolute positional embedding**에서 **relevant positional encodings**을 활용

TransformerXL - Introduction

Transformer보다 긴 Longer dependency 해결

RNN 대비 80%, vanilla Transformer대비 450%

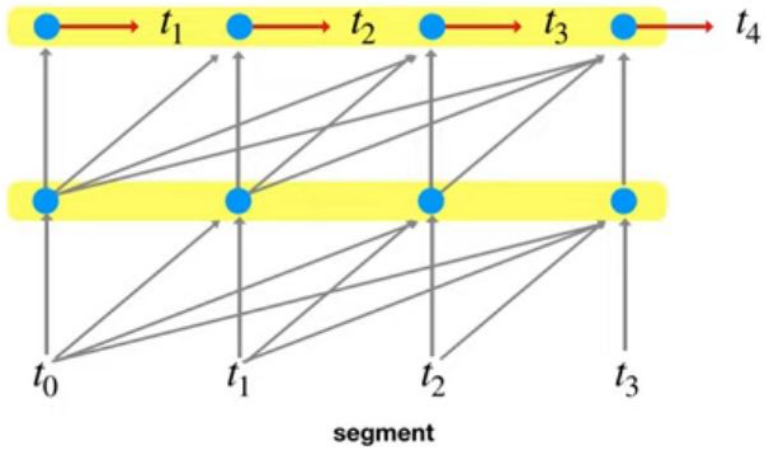
Article Generation으로 수천 단어 정도의 토큰 생성을 가능

TransformerXL - Related Work

- **Truncated Back Propagation Through Time (Truncated BPTT)**
  - **RNN** 기반의 모델
  - 이전 **batch**에서 **hidden states**를 가져와 현 **batch**에 넘겨준다
- **Character Level Language Modeling with Deeper self-attention**
  - **Transformer**를 **Char**로 접근
  - 매우 깊은 **(64 Layer)**를 쌓았음.

TransformerXL - Model - Vanilla Transformer Language Model

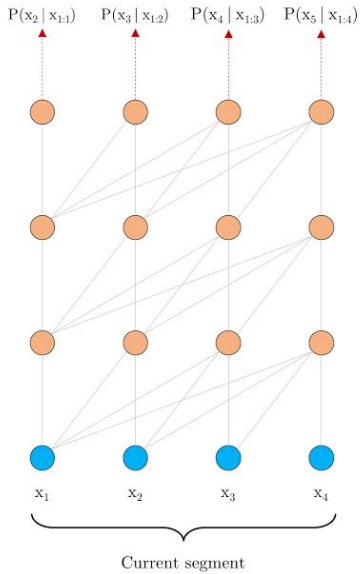
Vanilla Transformer (Char Transformer)



*Al-Rhou et al 2018, Character-Level Language Modeling with Deeper Self-Attention

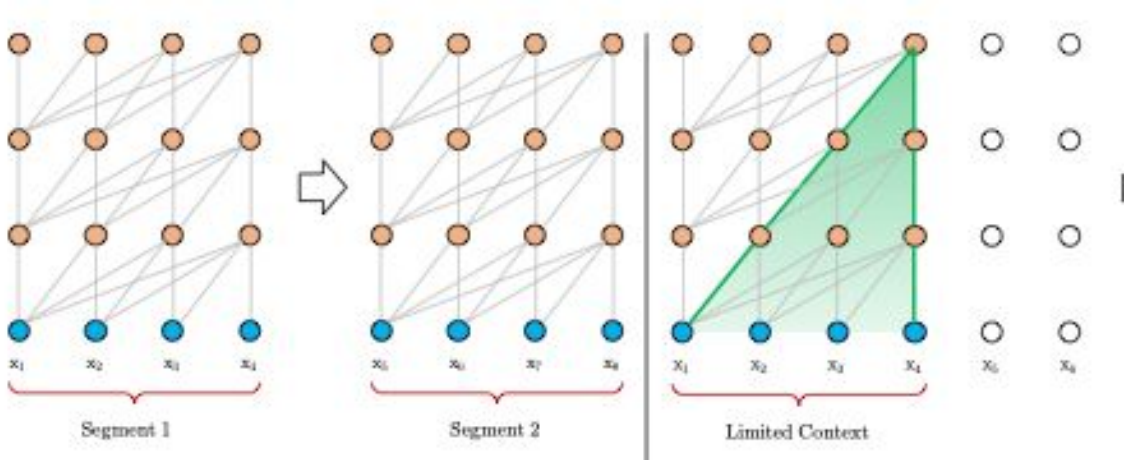
TransformerXL - Model - Vanilla Transformer Language Model

Valina Transformer: 정보가 **segments** 단위로 나눠져 전후 문맥을 볼 수가 없음



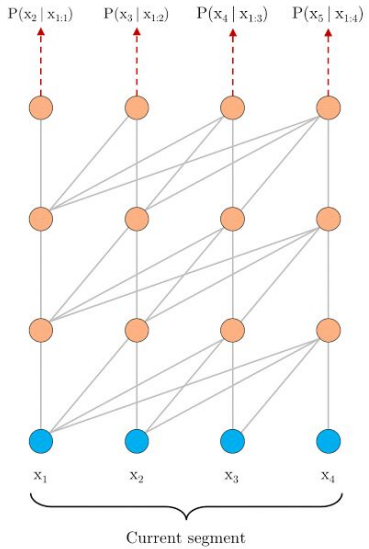
TransformerXL - Model - Vanilla Transformer Language Model

segment 단위로 language modeling을 학습하면, segment 크기를 넘어서는 long-term dependaency를 학습할 수 없고, 문장이나 의미를 고려하지 않고 segment가 나눠지는 context fragmentation 문제가 발생.



TransformerXL - Model - Segment-Level Recurrence with State Reuse

XL의 문제 해결법: Segment Recurrence



TransformerXL - Model - Segment-Level Recurrence with State Reuse

Advantages

X9를 예측 할때, 정보가 없어 예측하기 힘든 것을 강화

기존에 이슈가 되었던 **Seg**끼리 정보가 전파가 안되는 점을 해결

No Grad (모든 히든을 **Cache**로 가지고 있음) -> 계산이 빠름

TransformerXL - Model - Segment-Level Recurrence with State Reuse

일반

관련 수식

Stop Gradient 후 Cache화 + 현재와 Concat

$$\widetilde{\mathbf{h}}_{\tau+1}^{n-1} = \left[\mathrm{SG}(\mathbf{h}_{\tau}^{n-1}) \circ \mathbf{h}_{\tau+1}^{n-1}\right],$$

[Extended Context]

$$\mathbf{q}_{\tau+1}^{n}, \mathbf{k}_{\tau+1}^{n}, \mathbf{v}_{\tau+1}^{n} = \mathbf{h}_{\tau+1}^{n-1}\mathbf{W}_{q}^{\top}, \widetilde{\mathbf{h}}_{\tau+1}^{n-1}\mathbf{W}_{k}^{\top}, \widetilde{\mathbf{h}}_{\tau+1}^{n-1}\mathbf{W}_{v}^{\top},$$

[Query, Key, Value Vector]

$$\mathbf{h}_{\tau+1}^{n} = \text{Transformer-Layer}\left(\mathbf{q}_{\tau+1}^{n}, \mathbf{k}_{\tau+1}^{n}, \mathbf{v}_{\tau+1}^{n}\right).$$

[Self-attention + FF]

TransformerXL - Model - Segment-Level Recurrence with State Reuse

일반

Code

```python
def _update_mems(self, hids, mems, qlen, mlen):
    # does not deal with None
    if mems is None: return None

    # mems is not None
    assert len(hids) == len(mems), 'len(hids) != len(mems)'

    # There are `mlen + qlen` steps that can be cached into mems
    # For the next step, the last `ext_len` of the `qlen` tokens
    # will be used as the extended context. Hence, we only cache
    # the tokens from `mlen + qlen - self.ext_len - self.mem_len
    # to `mlen + qlen - self.ext_len`.
    with torch.no_grad():
        new_mems = []
        end_idx = mlen + max(0, qlen - 0 - self.ext_len)
        beg_idx = max(0, end_idx - self.mem_len)
        for i in range(len(hids)):

            cat = torch.cat([mems[i], hids[i]], dim=0)
            new_mems.append(cat[beg_idx:end_idx].detach())

    return new_mems
```

```python
def forward(self, w, r, r_w_bias, r_r_bias, attn_mask=None, mems=None):
    qlen, rlen, bsz = w.size(0), r.size(0), w.size(1)
    # W : [36, 4, 200]
    # r : [36, 1, 200] if mems is None else [72, 1, 200]
    # mems : [36, 4, 200]

    if mems is not None:
        cat = torch.cat([mems, w], 0)
        # cat : [72, 4(batch_size), 200]
        if self.pre_lnorm:
            w_heads = self.qkv_net(self.layer_norm(cat))
        else:
            w_heads = self.qkv_net(cat)
        r_head_k = self.r_net(r)

        # w_heads : [72, 4, 12]
        w_head_q, w_head_k, w_head_v = torch.chunk(w_heads, 3, dim=-1)
        w_head_q = w_head_q[-qlen:]
```

TransformerXL- Model - Relative Positional Encoding

Segment Recurrence로 인한 **Absolute Position**의 문제를 해결하고자 제안

기존 **Positional Embedding**으로는 해결이 되지 않음 (위치가 겹침)

[0,1,2,3]	[0,1,2,3]	[0,1,2,3]
Segment 1	**Segment 2**	**Segment 3**

상대적 위치를 계산하여 문제 해결! **(key vector**와 **query vector (i - j)**

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

TransformerXL - Model - Relative Positional Encoding

일반

기존

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

Query = (E + U) * W_q

Key = (E + U) * W_K

↓ 임베딩 ↳ 포지션정보

Attention 수식 $Q^T K$

Attention*

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

① Attention 수식 대입

Q^T : $(E^T W_q^T + W_q^T U^T)$

K : $(E W_K + W_K U)$

(a) $E^T W_q^T W_K E + E^T W_q^T W_K U$

(c) $U^T W_q^T W_K E + U^T W_q W_K U_j$

TransformerXL - Model - Relative Positional Encoding

개선

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

TransformerXL - Model - Relative Positional Encoding

Quadratic 시간 복잡도를 Linear하게 푸는 예)

Q를 WR의 곱으로 만들기 위해, Q를 Reversed order로 정의

정의된 Q matrix를 q와 곱하고, left-shift를 취한다.

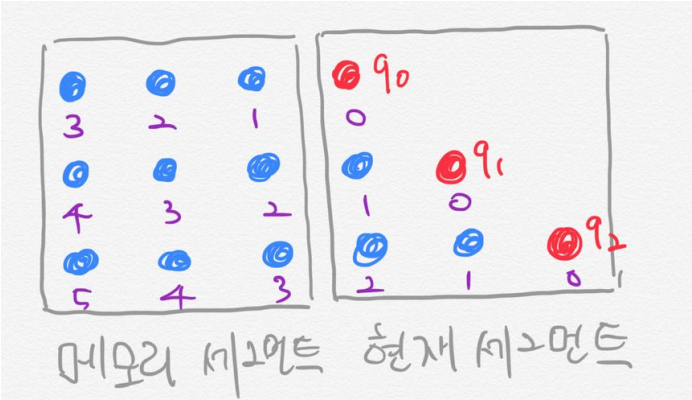
ing that a naive way to compute $\mathbf{A}$ requires computing $\mathbf{W}^n_{k,R}\mathbf{R}_{i-j}$ for all pairs (i, j), whose cost is quadratic w.r.t. the sequence length. However, noticing that the value of $i - j$ only ranges from zero to the sequence length, we show a simple computation procedure in Appendix B, which reduces the cost to be linear w.r.t. the sequence length.

TransformerXL - Model - Relative Positional Encoding

일반

발 없는 말이 천리 간다 -> [발, 없는, 말, 이, 천리, 간다]

연산이 Quadratic...



메모리 세그먼트: [발, 없는, 말]

현재 세그먼트: [이, 천리, 간다]

q0: 이

q1: 천리

q2: 간다

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

ing that a naive way to compute $\mathbf{A}$ requires computing $\mathbf{W}^{n}_{k,R}\mathbf{R}_{i-j}$ for all pairs (i, j), whose cost is quadratic w.r.t. the sequence length. However, noticing that the value of $i - j$ only ranges from zero to the sequence length, we show a simple computation procedure in Appendix B, which reduces the cost to be linear w.r.t. the sequence length.

TransformerXL - Model - Relative Positional Encoding

Quadratic 시간 복잡도를 Linear하게 푸는 예)

$$\mathbf{Q} := \begin{bmatrix} \mathbf{R}_{M+L-1}^{\top} \\ \mathbf{R}_{M+L-2}^{\top} \\ \vdots \\ \mathbf{R}_{1}^{\top} \\ \mathbf{R}_{0}^{\top} \end{bmatrix} \mathbf{W}_{k,R}{}^{\top} = \begin{bmatrix} [\mathbf{W}_{k,R}\mathbf{R}_{M+L-1}]^{\top} \\ [\mathbf{W}_{k,R}\mathbf{R}_{M+L-2}]^{\top} \\ \vdots \\ [\mathbf{W}_{k,R}\mathbf{R}_{1}]^{\top} \\ [\mathbf{W}_{k,R}\mathbf{R}_{0}]^{\top} \end{bmatrix} \in \mathbb{R}^{(M+L)\times d}$$

$$\mathbf{Q} = \begin{bmatrix} [\mathbf{W}_{k,R}\mathbf{R}_{5}]^{T} \\ [\mathbf{W}_{k,R}\mathbf{R}_{4}]^{T} \\ [\mathbf{W}_{k,R}\mathbf{R}_{3}]^{T} \\ [\mathbf{W}_{k,R}\mathbf{R}_{2}]^{T} \\ [\mathbf{W}_{k,R}\mathbf{R}_{1}]^{T} \\ [\mathbf{W}_{k,R}\mathbf{R}_{0}]^{T} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_{0}{}^{T} \\ \mathbf{Q}_{1}{}^{T} \\ \mathbf{Q}_{2}{}^{T} \\ \mathbf{Q}_{3}{}^{T} \\ \mathbf{Q}_{4}{}^{T} \\ \mathbf{Q}_{5}{}^{T} \end{bmatrix}$$

TransformerXL - Model - Relative Positional Encoding

일반

Quadratic 시간 복잡도를 Linear하게 푸는 예)

Next, we collect the term (b) for all possible i, j into the following $L \times (M + L)$ matrix,

$$\mathbf{B} = \begin{bmatrix} q_0^\top \mathbf{W}_{k,R}\mathbf{R}_M & \cdots & q_0^\top \mathbf{W}_{k,R}\mathbf{R}_0 & 0 & \cdots & 0 \\ q_1^\top \mathbf{W}_{k,R}\mathbf{R}_{M+1} & \cdots & q_1^\top \mathbf{W}_{k,R}\mathbf{R}_1 & q_1^\top \mathbf{W}_{k,R}\mathbf{R}_0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ q_{L-1}^\top \mathbf{W}_{k,R}\mathbf{R}_{M+L-1} & \cdots & q_{L-1}^\top \mathbf{W}_{k,R}\mathbf{R}_{M+L-1} & q_{L-1}^\top \mathbf{W}_{k,R}\mathbf{R}_{L-1} & \cdots & q_{L-1}^\top \mathbf{W}_{k,R}\mathbf{R}_0 \end{bmatrix}$$

$$= \begin{bmatrix} q_0^\top \mathbf{Q}_{L-1} & \cdots & q_0^\top \mathbf{Q}_{M+L-1} & 0 & \cdots & 0 \\ q_1^\top \mathbf{Q}_{L-2} & \cdots & q_1^\top \mathbf{Q}_{M+L-2} & q_1^\top \mathbf{Q}_{M+L-1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ q_{L-1}^\top \mathbf{Q}_0 & \cdots & q_{L-1}^\top \mathbf{Q}_M & q_{L-1}^\top \mathbf{Q}_{M+1} & \cdots & q_{L-1}^\top \mathbf{Q}_{M+L-1} \end{bmatrix}$$

Then, we further define

$$\widetilde{\mathbf{B}} = \mathbf{q}\mathbf{Q}^\top = \begin{bmatrix} q_0^\top \mathbf{Q}_0 & \cdots & q_0^\top \mathbf{Q}_M & q_0^\top \mathbf{Q}_{M+1} & \cdots & q_0^\top \mathbf{Q}_{M+L-1} \\ q_1^\top \mathbf{Q}_0 & \cdots & q_1^\top \mathbf{Q}_M & q_1^\top \mathbf{Q}_{M+1} & \cdots & q_1^\top \mathbf{Q}_{M+L-1} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ q_{L-1}^\top \mathbf{Q}_0 & \cdots & q_{L-1}^\top \mathbf{Q}_M & q_{L-1}^\top \mathbf{Q}_{M+1} & \cdots & q_{L-1}^\top \mathbf{Q}_{M+L-1} \end{bmatrix}.$$

Now, it is easy to see an immediate relationship between $\mathbf{B}$ and $\widetilde{\mathbf{B}}$, where the i-th row of $\mathbf{B}$ is simply a left-shifted version of i-th row of $\widetilde{\mathbf{B}}$. Hence, the computation of $\mathbf{B}$ only requires a matrix multiplication $\mathbf{q}\mathbf{Q}^\top$ to compute $\widetilde{\mathbf{B}}$ and then a set of left-shifts.

TransformerXL - Model - Relative Positional Encoding

일반

그림1

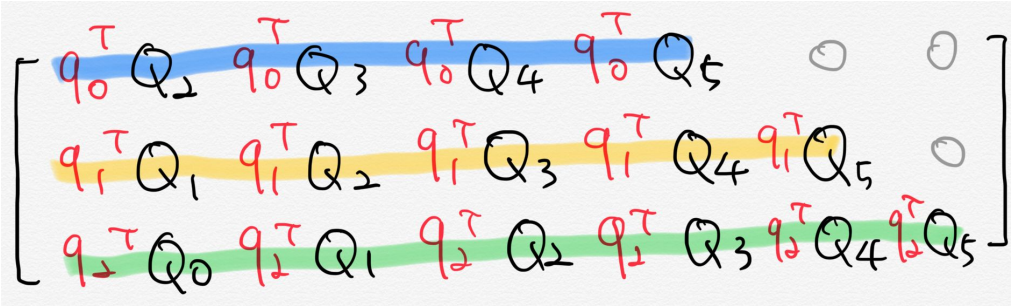
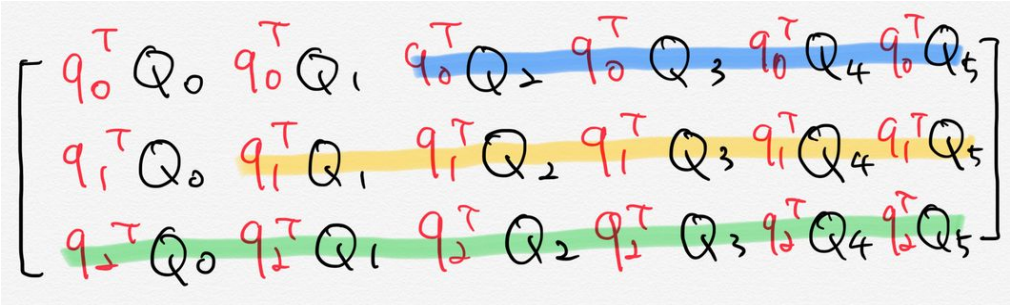


그림2



$$\mathbf{Q} = \begin{bmatrix} [\mathbf{W}_{k,R}\mathbf{R}_5]^T \\ [\mathbf{W}_{k,R}\mathbf{R}_4]^T \\ [\mathbf{W}_{k,R}\mathbf{R}_3]^T \\ [\mathbf{W}_{k,R}\mathbf{R}_2]^T \\ [\mathbf{W}_{k,R}\mathbf{R}_1]^T \\ [\mathbf{W}_{k,R}\mathbf{R}_0]^T \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_0{}^T \\ \mathbf{Q}_1{}^T \\ \mathbf{Q}_2{}^T \\ \mathbf{Q}_3{}^T \\ \mathbf{Q}_4{}^T \\ \mathbf{Q}_5{}^T \end{bmatrix}$$

TransformerXL - Model - Relative Positional Encoding

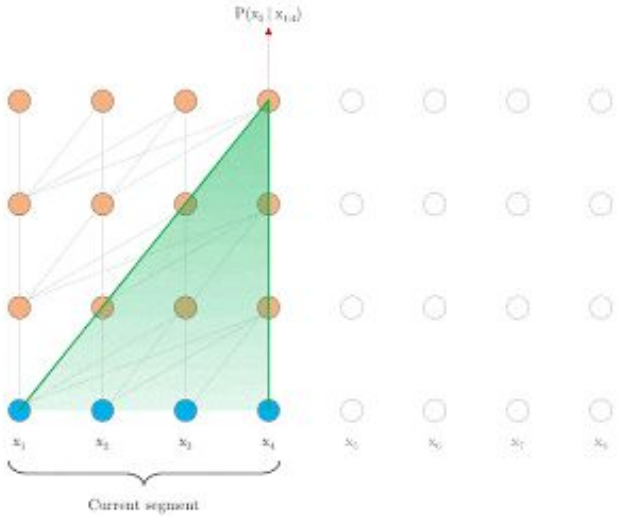
최종적으로 Transformer-XL의 계산 과정은 아래와 같음.

$$
\begin{aligned}
\widetilde{\mathbf{h}}_{\tau}^{n-1} &= \left[\mathrm{SG}(\mathbf{m}_{\tau}^{n-1}) \circ \mathbf{h}_{\tau}^{n-1}\right] \\
\mathbf{q}_{\tau}^{n}, \mathbf{k}_{\tau}^{n}, \mathbf{v}_{\tau}^{n} &= \mathbf{h}_{\tau}^{n-1}\mathbf{W}_{q}^{n\top}, \widetilde{\mathbf{h}}_{\tau}^{n-1}\mathbf{W}_{k,E}^{n\top}, \widetilde{\mathbf{h}}_{\tau}^{n-1}\mathbf{W}_{v}^{n\top} \\
\mathbf{A}_{\tau,i,j}^{n} &= \mathbf{q}_{\tau,i}^{n\top}\mathbf{k}_{\tau,j}^{n} + \mathbf{q}_{\tau,i}^{n\top}\mathbf{W}_{k,R}^{n}\mathbf{R}_{i-j} \\
&\quad + u^{\top}\mathbf{k}_{\tau,j} + v^{\top}\mathbf{W}_{k,R}^{n}\mathbf{R}_{i-j} \\
\mathbf{a}_{\tau}^{n} &= \text{Masked-Softmax}(\mathbf{A}_{\tau}^{n})\mathbf{v}_{\tau}^{n} \\
\mathbf{o}_{\tau}^{n} &= \text{LayerNorm}(\text{Linear}(\mathbf{a}_{\tau}^{n}) + \mathbf{h}_{\tau}^{n-1}) \\
\mathbf{h}_{\tau}^{n} &= \text{Positionwise-Feed-Forward}(\mathbf{o}_{\tau}^{n})
\end{aligned}
$$

TransformerXL - Model - Prediction

Vanilla Prediction

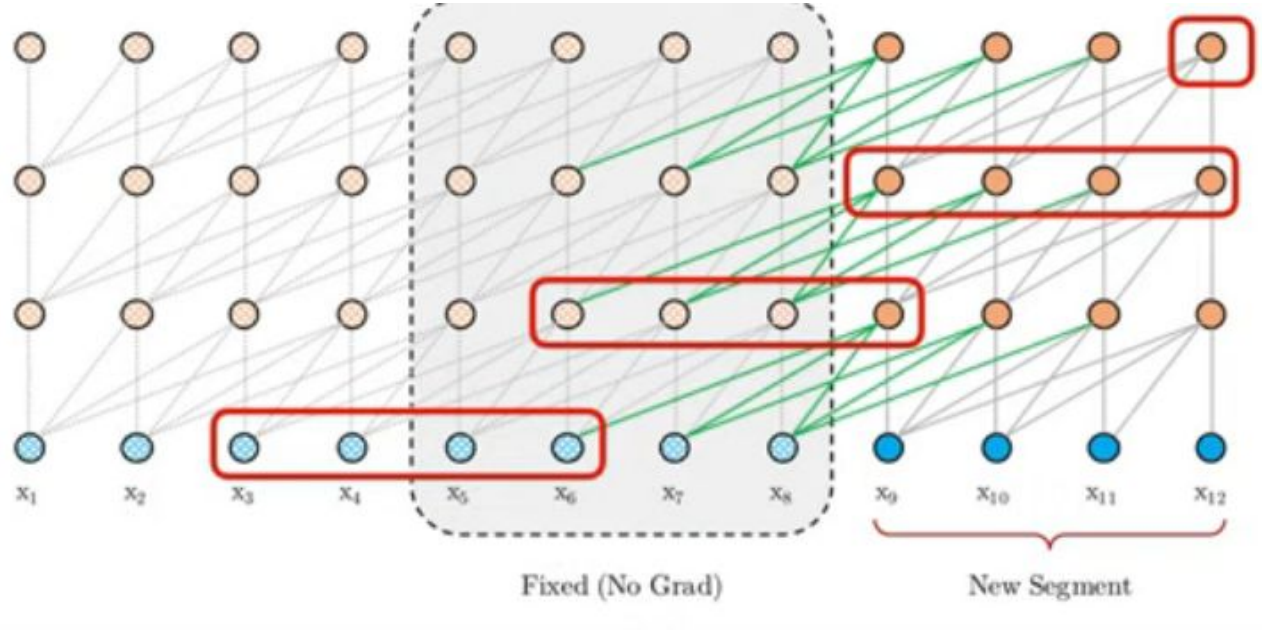
기존은 포지션 당 1개의 예측만 가능 **(Extremly Expensive)**



TransformerXL - Model - Prediction

TransformerXL Prediction

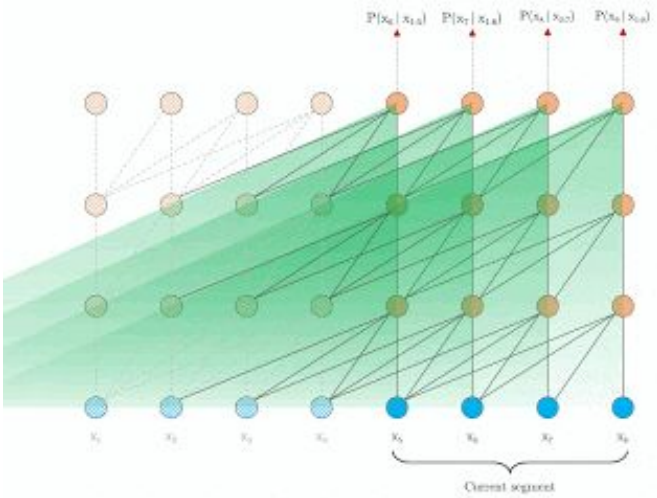
이전 segments에서 메모리를 사용하기 때문에, 한 segments에 대한 결과가 한꺼번에 연산 가능



TransformerXL - Model - Prediction

TransformerXL Prediction

이전 segments에서 메모리를 사용하기 때문에, 한 segments에 대한 결과가 한꺼번에 가능



TransformerXL - Experiments - Main Result

Dataset 및 결과

- WikiText-103*
  - **Word-level** dataset with long-term dependency
  - 103M training tokens from 28K articles, average length of 3.6K tokens per article
- enwiki-8
  - 100M bytes of unprocessed Wikipedia text
- text-8
  - 100M processed Wikipedia **characters**
- One Billion Word
  - Shuffled sentences (**No long-term dependency**)

TransformerXL - Experiments - Main Result

평가방법

- Perplexity (PPL)

$$PPL(W) = P(w_1 w_2 \dots w_N)^{-\frac{1}{N}}$$

- Bit Per Character (bpc)

$$BPC = Average(-\log_2 P(x_{t+1} \mid y_t))$$

- **Relative Effective Context Length (RECL)**
  - Effective Context Length* : longest length to which increasing the context span would lead to a gain more than a threshold
  - RECL : relative improvement over the best short context model

TransformerXL - Experiments - Main Result

평가 결과 (1)

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$^{\diamond}$	247M	20.5
Ours - Transformer-XL Large	257M	**18.3**

Table 1: Comparison with state-of-the-art results on WikiText-103. $^{\diamond}$ 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.

LM모델

Best

Char-Level도 좋은 성능

TransformerXL - Experiments - Main Result

평가 결과 (2)

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.

Char-level 모델

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$^\diamond$	0.46B	24.1
Baevski and Auli (2018) - Adaptive Input$^\diamond$	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. $^\diamond$ indicates contemporary work.

Long-term + Short-term 보다 좋음

TransformerXL - Experiments - Relative Effective Context Length

Inference 속도의 차이 - 학습과 모델 크기의 한계는 있지만 Generation으로는 최적화 모델이 아닐까?

Model	$r = 0.1$	$r = 0.5$	$r = 1.0$
Transformer-XL 151M	**900**	**800**	**700**
QRNN	500	400	300
LSTM	400	300	200
Transformer-XL 128M	**700**	**600**	**500**
- use Shaw et al. (2018) encoding	400	400	300
- remove recurrence	300	300	300
Transformer	128	128	128

Table 8: Relative effective context length (RECL) comparison. See text for the definition of RECL and r. The first three models and the last four models are compared as two *model groups* when we calculate RECL (RECL is computed on a model group rather than a single model). Each group has the same parameter budget.

How Long

기존 Char-transformer

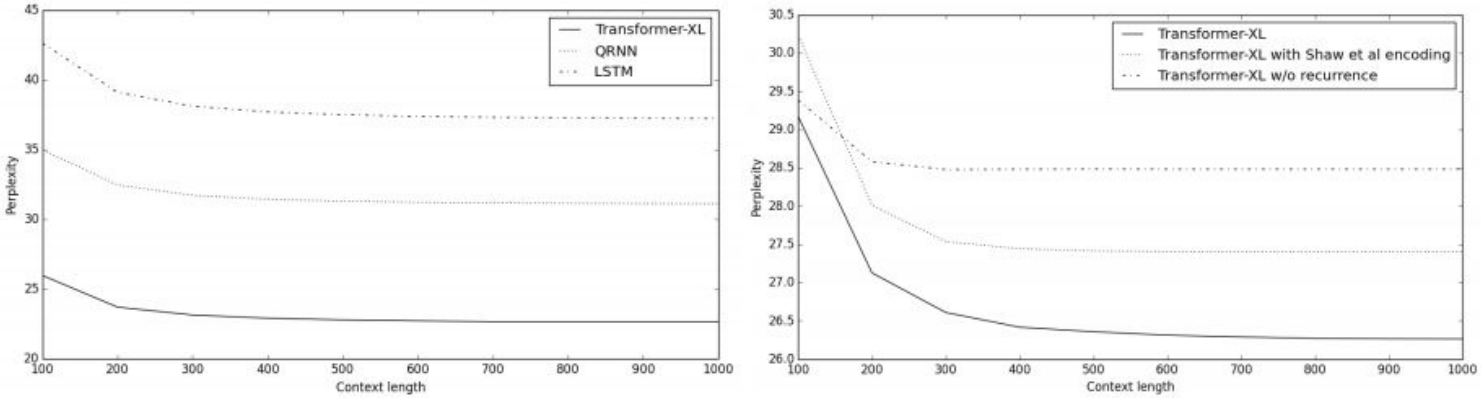
Attn Len	How much Al-Rfou et al. (2018) is slower
3,800	1,874x
2,800	1,409x
1,800	773x
800	363x

Table 9: Slowdown in terms of running time during evaluation. Evaluation is based on per-token time on one GPU.

How Fast

TransformerXL - Experiments - Relative Effective Context Length

Relative position encoding과 recurrent 메카니즘 효과



(a) Transformer-XL vs RNNs

(b) Transformer-XL vs Baseline

Figure 4: Perplexity vs context length.

TransformerXL - Experiments - Generation Result

Generation의 예

Context:
Kershaw started the 2010 season by posting a 3.07 ERA in April, but did so by walking 22 batters in 29 innings. On May 4, he had his worst start of his career against the Milwaukee Brewers at Dodger Stadium, throwing just 57 pitches in 11 / 3 innings, while retiring only four of the 13 batters he faced — including the pitcher. He was booed loudly upon being pulled from the game. Kershaw said after the game, " I didn't give our team any kind of chance. It's just not a good feeling to let your teammates down, let everybody down. It stings, it hurts. I 've got to figure things out. " Kershaw rebounded his next start by pitching an 8 inning two-hitter and out-dueling the then undefeated Ubaldo Jiménez. He credited his control of the slider being the major turning point for him. Later in the season, he was suspended for five games after hitting Aaron Rowand of the Giants with a pitch in a game on July 20. The incident occurred after both teams were given a warning following Giants ace Tim Lincecum hitting Matt Kemp earlier in the game. He threw his first career complete game shutout on September 14, 2010 also against San Francisco and finished the season with a record of 13 – 10 and a 2.91 ERA in 32 starts, pitching 2041 / 3 innings and recording 212 strikeouts.
= = = = 2011 season: 1st Cy Young Award = = = =
After finishing the 2010 season strong, the Dodgers named Kershaw as the Opening Day Starter for the 2011 season. On May 29, he pitched the second complete-game shutout of his career, striking out 10 while winning a two-hitter against the Florida Marlins, 8 – 0; he also had two singles and an RBI, scoring twice in the game. He produced his third career shutout on June 20, a two-hit, 11-strikeout effort against the Detroit Tigers. Kershaw became the first Dodgers starter to strike out the side in the 9th inning since Sandy Koufax's perfect game. In his next start, on June 26, Kershaw pitched another complete game (against the Los Angeles Angels of Anaheim). He became the first Dodger starter to have back-to-back complete game victories since Jeff Weaver in the 2005 season and the first Dodger to have double-digit strikeouts in consecutive starts since Chan-Ho Park in the 2000 season. He was awarded the National League Player of the Week award for the week of June 20 – 26 as a result of those two starts. Midway through June, Kershaw had amassed 32 career victories

XL Generation:
, tying him with Luis Castillo for the third best start total in the league and trailing only Johnny Bench and Eddie Mathews (37). The Dodgers signed Kershaw to a one-year deal on July 14, 2011, with a player option for 2012, in a deal worth just under $ 300,000. The Dodgers entered the 2011 season as the major league leaders, and Kershaw had the best record in the National League in saves (39), games started (78) and walks (120); he also had the lowest win / loss ratio in the league (1.87), and he had the worst on-base percentage (.349). He had the best ERA (2.00), most wins (16), or most shutouts (29) in his career (25), while leading the league in losses per nine innings pitched (4.01). Following the 2011 season, the Dodgers traded Kershaw to the Kansas City Royals on July 29 for pitcher Javier Torres.
= = = Kansas City Royals = = =
= = = = 2012 season = = = =
During spring training, Kershaw played very well. He was selected to spring training as a relief pitcher for the Royals for the 2012 season. After an injury to closer Javier Vázquez, he was activated on April 29 to replace Matt Holliday in the Royals ' starting rotation. In his only start with the Royals, on August 6, 2012, Kershaw struck out five batters in seven innings pitched to help the Royals to their first victory in franchise history. On September 27, 2012, it appeared Kershaw was going to pitch a complete game shutout against the Detroit Tigers, but did not manage to do so since the Tigers won 3 – 1. At the conclusion of the season, Kershaw was named Major League Baseball's Most Valuable Player, was chosen to the All-Star Game at Busch Stadium and was named to the All-Star Game as the starting pitcher at shortstop. The Royals announced on February 4, 2013 that Kershaw would spend 2013 as starting pitcher, though he was expected to miss the season as a result of a shoulder injury.
= = = = 2013 season = = = =
On May 17, 2013, Kershaw sustained another back injury and did not start in August and October 2013. He appeared in 22 starts, all starts, finishing with a strikeout-to-walk ratio of 1.50 and a 2.91 ERA. He also had the third most strikeouts in the league: 10. On May 20, 2013, he

Reference:
, a 3.15 ERA and 593 career strikeouts in 568.2 innings. According to the Elias Sports Bureau, Kershaw was the first 23-year-old pitcher to have that many victories, an ERA that low and an average of more than one strikeout per inning since ERA became an official statistic in 1910. Kershaw was selected to the National League team for the 2011 Major League Baseball All-Star Game, his first All-Star selection. In the month of July, Kershaw was 4 – 1 with a 2.02 ERA and NL-leading 45 strikeouts, earning him the National League Pitcher of the Month Award. On August 23, he struck out Matt Holliday of the St. Louis Cardinals for his 200th strikeout of the season and became the 10th Dodger pitcher to record back-to-back 200 strikeout seasons and the first since Chan-Ho Park did it in the 2001 season. Kershaw finished the 2011 season by leading the NL with 21 wins, 248 strikeouts and a 2.28 ERA, winning the NL pitching Triple Crown, the first Triple Crown winner since Jake Peavy of the 2007 San Diego Padres and the first Dodger since Sandy Koufax won it in the 1966 season. Justin Verlander of the Detroit Tigers won the American League Triple Crown the same season, marking the first major-league season since 1924 to feature Triple Crown-winning pitchers in both leagues. Kershaw's 21 wins were the most by a Dodger pitcher since Orel Hershiser won 23 during the 1988 season. His ERA was the lowest by a Dodger since Hershiser's 2.03 in the 1985 season, his strikeouts were the most by a Dodger since Koufax's 317 in 1966 and his 233 1 / 3 innings pitched were the most since Chan Ho Park pitched 234 in 2001. Since 1965 when Koufax did it, Peavy and Kershaw are only two pitchers in the National League have led the league in wins, strikeouts, ERA, and WHIP (walks plus hits per inning pitched). Kershaw also became just the second <unk> to have a 240-plus strikeouts in a season before the age of 24, joining Vida Blue. After the season, Kershaw was awarded the Warren Spahn Award as the best left-handed pitcher in 2011, the Players Choice Award for Most Outstanding National League pitcher, the Gold Glove Award as the top fielding pitcher in the NL and the Sporting News (TSN) National League Pitcher of the Year. He was additionally selected as the starting pitcher for the TSN NL All-Star Team. On November 17, he was honored with the National League Cy Young Award,

요약하자면,

Kersaw(커쇼)라는 LA 다져스 야구 선수인데,

1. 커쇼는 **Royals**라는 팀에 가지 않았는데도, 상상하여 관련 문장을 생성
2. 모델이 **===2011===**부분의 시간 순서를 인식하여, **===2012===, ===2013===**을 생성한 것.
3. **2012**년에 **injury** (부상)이라는 컨텍스트를 이해하여, **2013**년에 **"another back injuery"**라는 컨텍스트를 생성

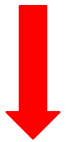
TransformerXL - 마무리

Limitation (Paper 거절 이유)

Better language 모델이지만, downstream task에서의 자료가 없음

Document 생성 잘 된다고 했는데 없음..

OpenGPT2가 관련 모델을 이겼음



XLNet