KOREA UNIVERSITY

DSBA Transformer survey paper study

A Survey of Transformers

#4 : Attention 3

arXiv preprint

DSBA
Data Science & Business Analytics

고려대학교 산업경영공학과

Data Science & Business Analytics Lab

이유경, 김명섭, 윤훈상, 김지나, 허재혁, 김수빈

발표자 : 이유경

04 발표 목차

Contents

04 Low Rank Self-Attention

Introduction

Some empirical and theoretical analyses
report the self-attention matrix $\boldsymbol{A} \in \boldsymbol{R}^{\boldsymbol{T}\times\boldsymbol{T}}$ is often low-rank

(1) The Low-rank property could be explicitly modeled with parameterization
(2) The self-attention matrix could be replaced by a low-rank approximation

04 Low Rank Self-Attention

Introduction

Some empirical and theoretical analyses
report the self-attention matrix $A \in R^{T \times T}$ is often low-rank

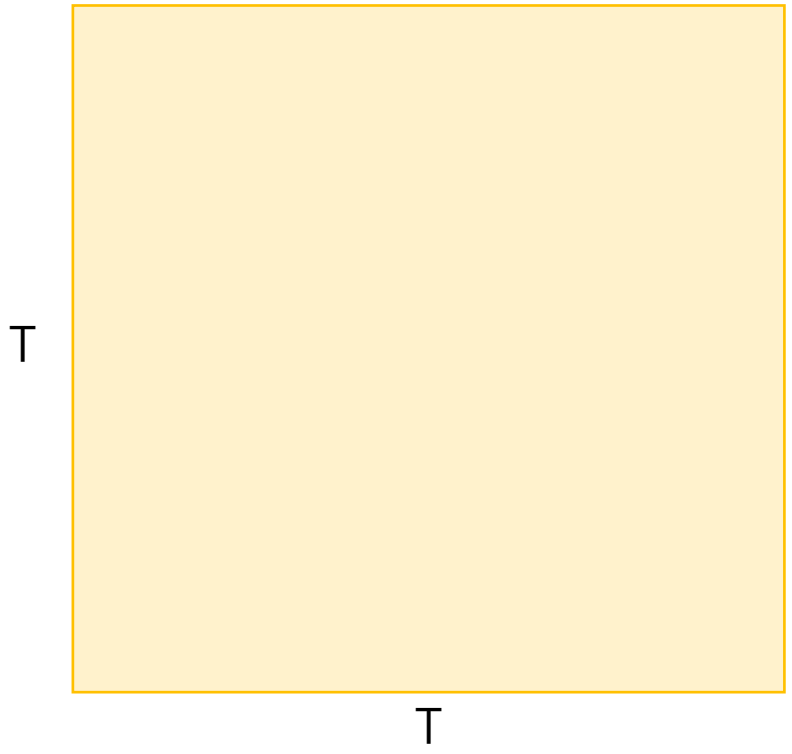
(1) The Low-rank property could be explicitly modeled with parameterization
(2) The self-attention matrix could be replaced by a low-rank approximation

~~감자기요 ?!~~

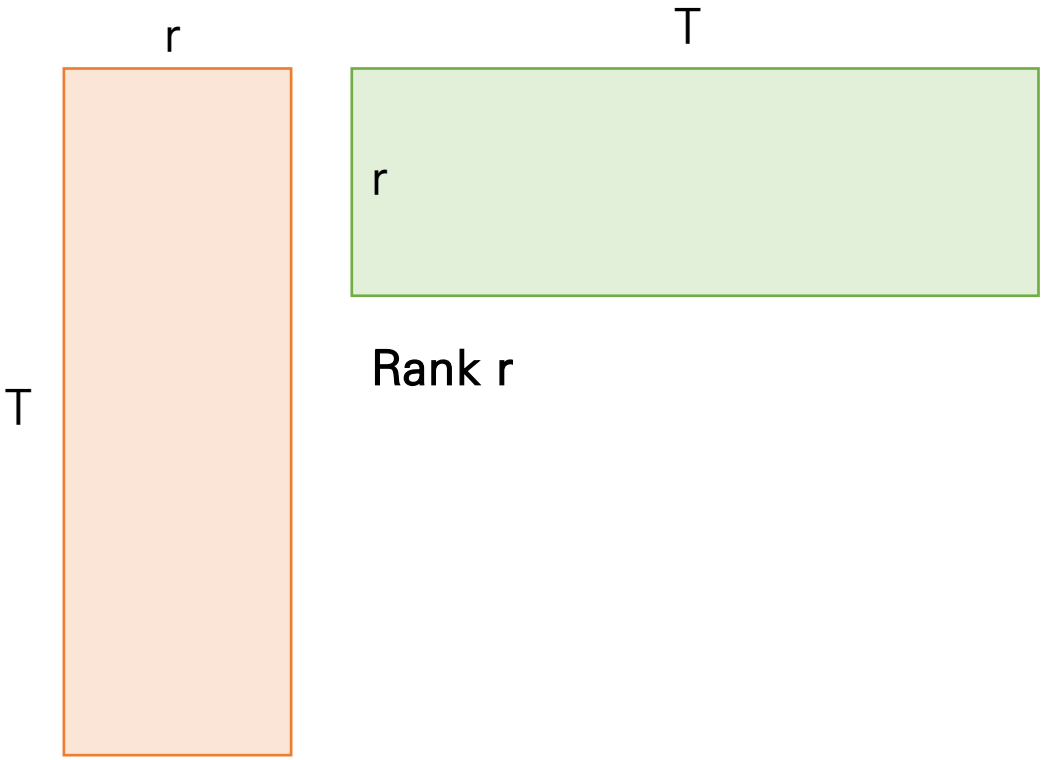
04 Low Rank Self-Attention

Introduction

Self attention matrix



Low rank matrix



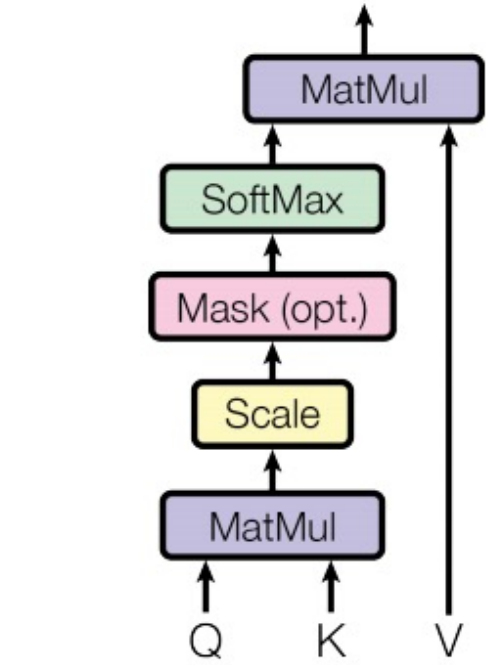
Rank r

Low Rank decomposition： https://www.youtube.com/watch?v=_FmoIBCUo9M

04 Low Rank Self-Attention

Introduction

- **Low Rank Self-Attention이 나오게 된 이유를 알아봅시다**
- Self-Attention mechanism의 특징 (Recap)



1. self-attention structure is powerful
2. It often requires more training data to learn parameters sufficiently
3. It is easy to overfit the data
4. Weak inductive bias

04 Low Rank Self-Attention

Introduction

https://gruuuuu.github.io/machine-learning/cnn-doc/#
https://wikidocs.net/22886

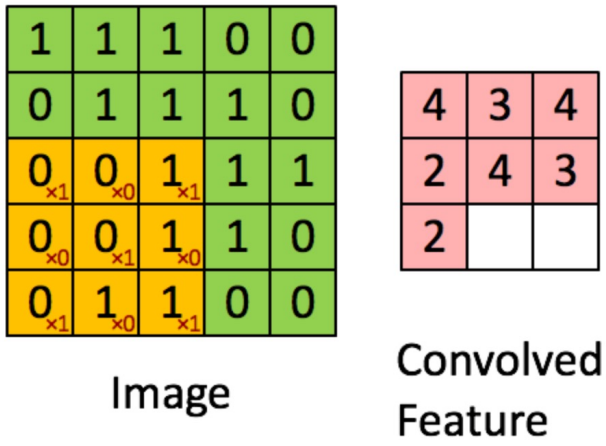
- Low Rank Self-Attention이 나오게 된 이유를 알아봅시다
- Self-Attention mechanism의 특징 (Recap)

Low-Rank and Locality Constrained Self-Attention for Sequence Modeling

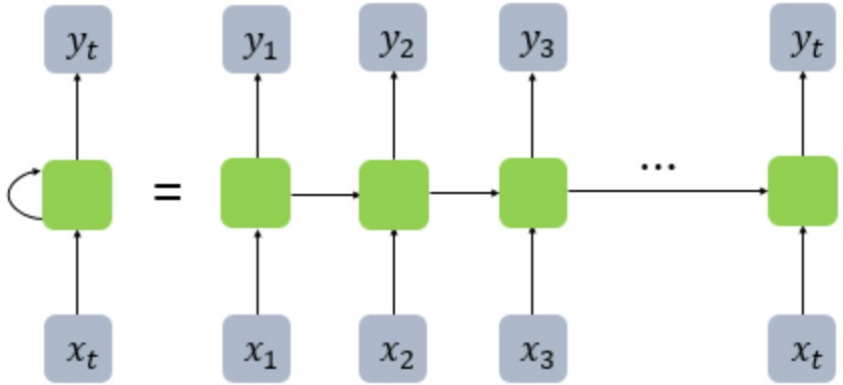
⋮

4 Weak inductive bias

- CNN, RNN ?



[CNN : local, neighbor]



[RNN : sequentially dependent]

: Inability of capturing the long-range and non-local dependency

04 Low Rank Self-Attention

Introduction

- **Low Rank Self-Attention이 나오게 된 이유를 알아봅시다**
- Self-Attention mechanism의 특징 (Recap)

Low-Rank and Locality Constrained Self-Attention for Sequence Modeling

1. self-attention structure is powerful
2. It often requires more training data to learn parameters sufficiently
3. It is easy to overfit the data
4. Weak inductive bias

: 시퀀스 내의 token pair가 correlate되었음을 가정함

: local , non-local dependency를 모두 고려 할 수 있음

04 Low Rank Self-Attention

Introduction

- **Low Rank Self-Attention이 나오게 된 이유를 알아봅시다**
- Self-Attention mechanism의 특징 (Recap)

Low-Rank and Locality Constrained Self-Attention for Sequence Modeling

1. self-attention structure is powerful
2. It often requires more training data to learn parameters sufficiently
3. It is easy to overfit the data
4. Weak inductive bias

- 하지만 Transformer는 위와 같은 특징으로
  Moderate-sized dataset에서는 generalization ability가 떨어짐

"Low rank의 관점에서"

- 그렇다면 알맞는 inductive bias를 주면 되겠다 ! – (1)
- Self attention과 비슷한데 complexity를 줄이면 되겠다 ! – (2)

04 Low Rank Self-Attention

Low Rank Attention (IEEE/ACM Trans. Audio, Speech and Lang 2019 , 3회 인용)

- **Low-Rank Parameterization**
- Low-Rank and Locality Constrained Self-Attention for Sequence Modeling (IEEE/ACM Trans. Audio, Speech and Lang 2019 , 3회 인용)

  : Sequence Modeling에 맞는 inductive bias를 추가해서 더 좋은 self Attention mechanism을 제안하자
  : Attention matrix가 Linguistic prior를 가질 수 있도록 Band Attention / Low-Rank Attention으로 decompose

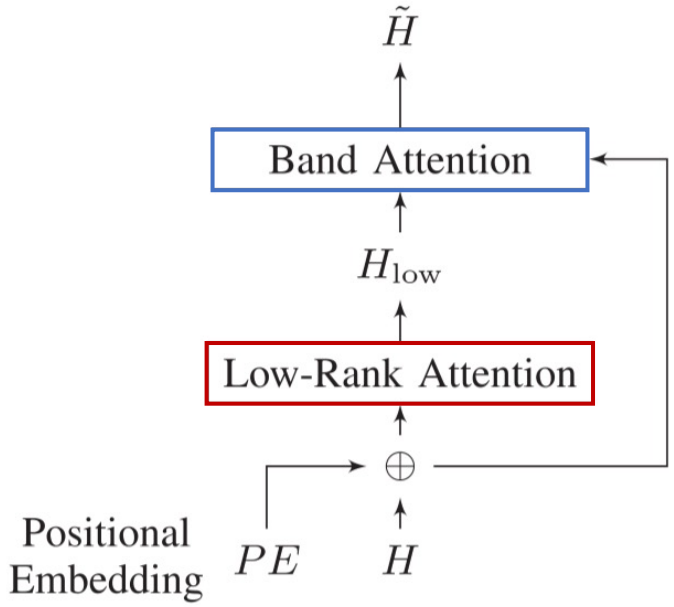
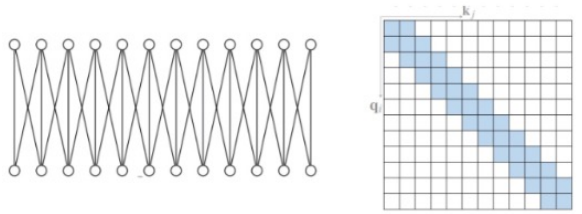


Fig. 2. Pipeline of our proposed self-attention layer.

- Band Attention (Sliding Window Attention / Local Attention)

  언어의 Locality를 반영하기 위하여, 근접한 이웃 노드들에 대해서만 Attend



#2 Attention – 윤훈상

- Low Rank Attention (non local Attention)

$$\mathbf{A}_l = \mathbf{H}\mathbf{W}_l^Q(\mathbf{W}_l^K)^T\mathbf{H}^T \quad \mathbf{A} = \text{softmax}(\mathbf{H}\mathbf{W}^Q(\mathbf{W}^K)^T\mathbf{H}^T),$$

$$= \text{softmax}(\mathbf{H}\mathbf{W}\mathbf{H}^T)$$

where $\mathbf{W}_l^Q \in \mathbb{R}^{d\times r}$, $\mathbf{W}_l^K \in \mathbb{R}^{d\times r}$ and $r \ll l \ll d$.

04 Low Rank Self-Attention

Low Rank Attention (IEEE/ACM Trans. Audio, Speech and Lang 2019 , 3회 인용)

- **Low-Rank Parameterization**
- Low-Rank and Locality Constrained Self-Attention for Sequence Modeling (IEEE/ACM Trans. Audio, Speech and Lang 2019 , 3회 인용)

  : Sequence Modeling에 맞는 inductive bias를 추가해서 더 좋은 self Attention mechanism을 제안하자
  : Attention matrix가 Linguistic prior를 가질 수 있도록 Band Attention / Low-Rank Attention으로 decompose

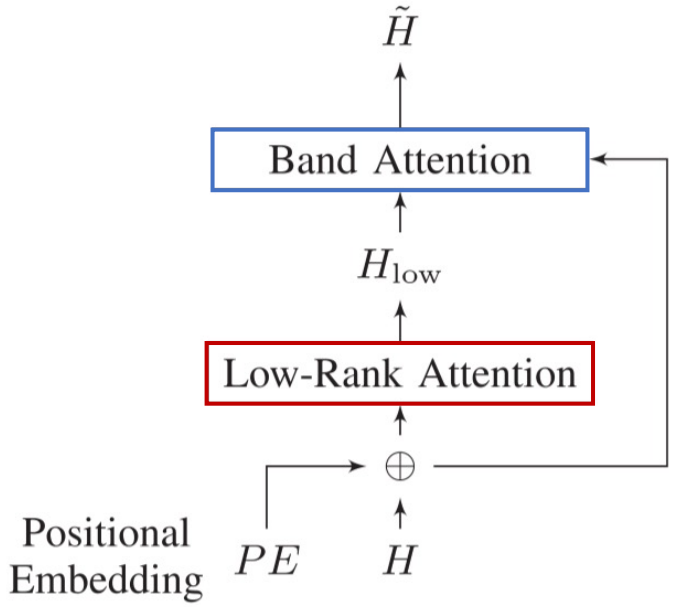


Fig. 2. Pipeline of our proposed self-attention layer.

- Low Rank Attention (non local Attention)

$$\mathbf{A}_l = \mathbf{H}\mathbf{W}_l^Q(\mathbf{W}_l^K)^T\mathbf{H}^T \quad \mathbf{A} = \text{softmax}(\mathbf{H}\mathbf{W}^Q(\mathbf{W}^K)^T\mathbf{H}^T),$$
$$= \text{softmax}(\mathbf{H}\mathbf{W}\mathbf{H}^T)$$

where $\mathbf{W}_l^Q \in \mathbb{R}^{d\times r}$, $\mathbf{W}_l^K \in \mathbb{R}^{d\times r}$ and $r \ll l \ll d$.

✓ Low rank property를 inductive bias로 사용함

✓ Self attention matrix를 low rank attention module로 변환
(기존보다 더 작은 D_k주면서도 non local 정보를 학습할 수 있도록)

https://blog.tensorflow.org/2020/02/matrix-compression-operator-tensorflow.html

04 Low Rank Self-Attention

Low Rank Attention (IEEE/ACM Trans. Audio, Speech and Lang 2019 , 3회 인용)

- **Low-Rank Parameterization**
- Low-Rank and Locality Constrained Self-Attention for Sequence Modeling (IEEE/ACM Trans. Audio, Speech and Lang 2019 , 3회 인용)

  : Sequence Modeling에 맞는 inductive bias를 추가해서 더 좋은 self Attention mechanism을 제안하자
  : Attention matrix가 Linguistic prior를 가질 수 있도록 Band Attention / Low-Rank Attention으로 decompose

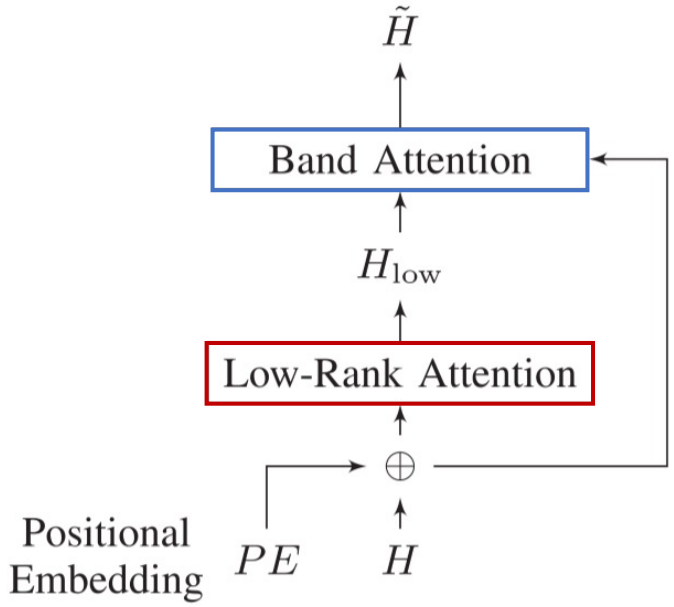
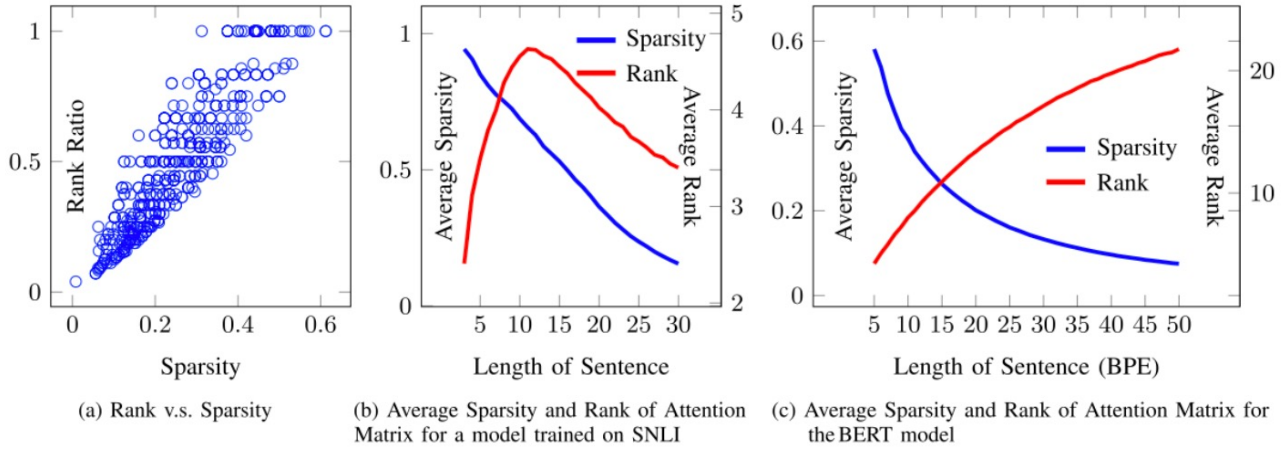


Fig. 2. Pipeline of our proposed self-attention layer.



(a) Rank v.s. Sparsity (b) Average Sparsity and Rank of Attention Matrix for a model trained on SNLI (c) Average Sparsity and Rank of Attention Matrix for the BERT model

✓ Low rank property를 inductive bias로 사용함

✓ Self attention matrix를 low rank attention module로 변환 (기존보다 더 작은 D_k주면서도 non local 정보를 학습할 수 있도록)

✓ 일종의 트릭으로 이해했음

https://blog.tensorflow.org/2020/02/matrix-compression-operator-tensorflow.html

04 Low Rank Self-Attention

Nyströmformer (AAAI 2021, 14회 인용)

- **Low-Rank Approximation** (Perfomer, Nyströmformer 등)
- Nyströmformer: A Nyström-Based Algorithm for Approximating Self-Attention (AAAI 2021, 14회 인용)

: Low Rank approximation을 통해 기존 Self Attention 연산의 계산 복잡도를 줄이자
: 이때 사용하는 방법은 matrix decompose에 많이 활용되는 Nyström method(2001)임

[Self attention]

$$S = \underset{\in \mathbf{R}^{n\times n}}{\text{softmax}}\left(\frac{QK^T}{\sqrt{d_q}}\right) = \begin{bmatrix} A_S & B_S \\ F_S & C_S \end{bmatrix}$$

where $A_S \in \mathbf{R}^{m\times m}, B_S \in \mathbf{R}^{m\times(n-m)}$,

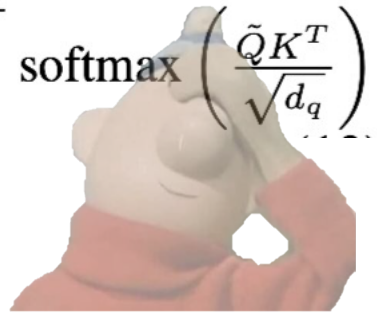
$F_S \in \mathbf{R}^{(n-m)\times m}\, C_S \in \mathbf{R}^{(n-m)\times(n-m)}$

m은 n 개의 column,row에서 샘플링 한 개수를 의미함

[Low rank Approximation]

$$\hat{S} = \begin{bmatrix} A_S & B_S \\ F_S & F_S A_S^+ B_S \end{bmatrix} = \begin{bmatrix} A_S \\ F_S \end{bmatrix} A_S^+ \begin{bmatrix} A_S & B_S \end{bmatrix}$$

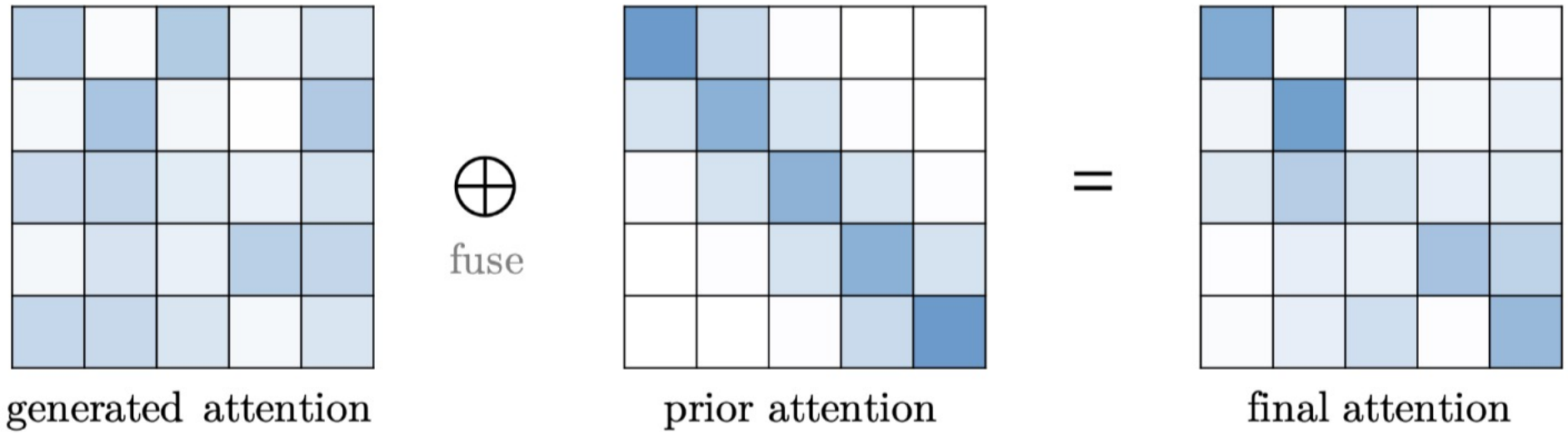
$$\hat{S} = \text{softmax}\left(\frac{Q\tilde{K}^T}{\sqrt{d_q}}\right)\left(\text{softmax}\left(\frac{\tilde{Q}\tilde{K}^T}{\sqrt{d_q}}\right)\right)^+ \text{softmax}\left(\frac{\tilde{Q}K^T}{\sqrt{d_q}}\right)$$



"....."

04 Attention with Prior

Introduction

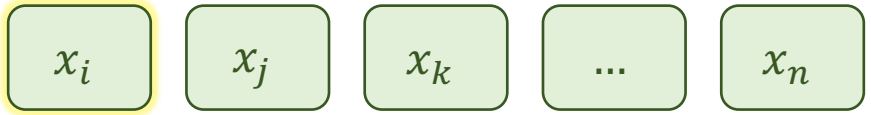


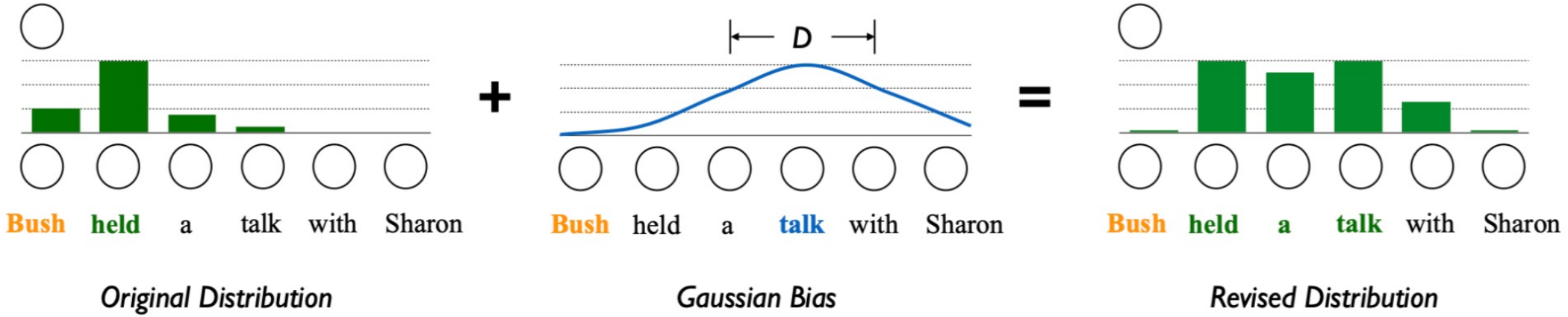
- 우리가 생성해낸 Attention을 더 풍부하게 만들어 줄 수 있는 방법 ?
  - Other source로 부터 만들어진 attention distribution을 활용하자 == Prior !
  - generated attention과 prior attention을 weighted sum으로 fuse 하는것이 일반적임

04 Attention with Prior

1) Prior that Models locality

- **Modeling Localness for Self-Attention Networks** (EMNLP 2018 , 96회 인용)
  - Self Attention은 global dependencies를 찾아낼 수 있음
  - 본 논문은 self attention network를 위한 model localness를 제안함
    → 이를 통해 useful local context를 찾아낼 수 있는 능력이 강화됨
  - Learnable Gaussian bias

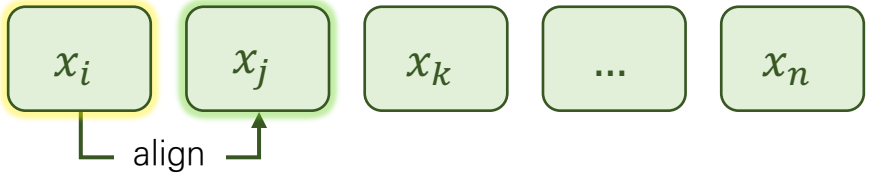


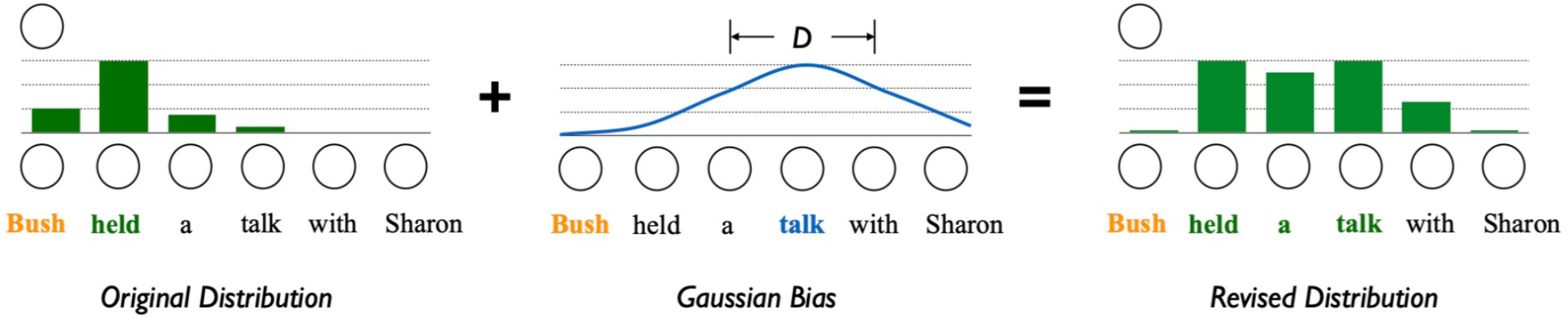


04 Attention with Prior

1) Prior that Models locality

- **Modeling Localness for Self-Attention Networks** (EMNLP 2018 , 96회 인용)
  - Self Attention은 global dependencies를 찾아낼 수 있음
  - 본 논문은 self attention network를 위한 model localness를 제안함
    → 이를 통해 useful local context를 찾아낼 수 있는 능력이 강화됨
  - Learnable Gaussian bias

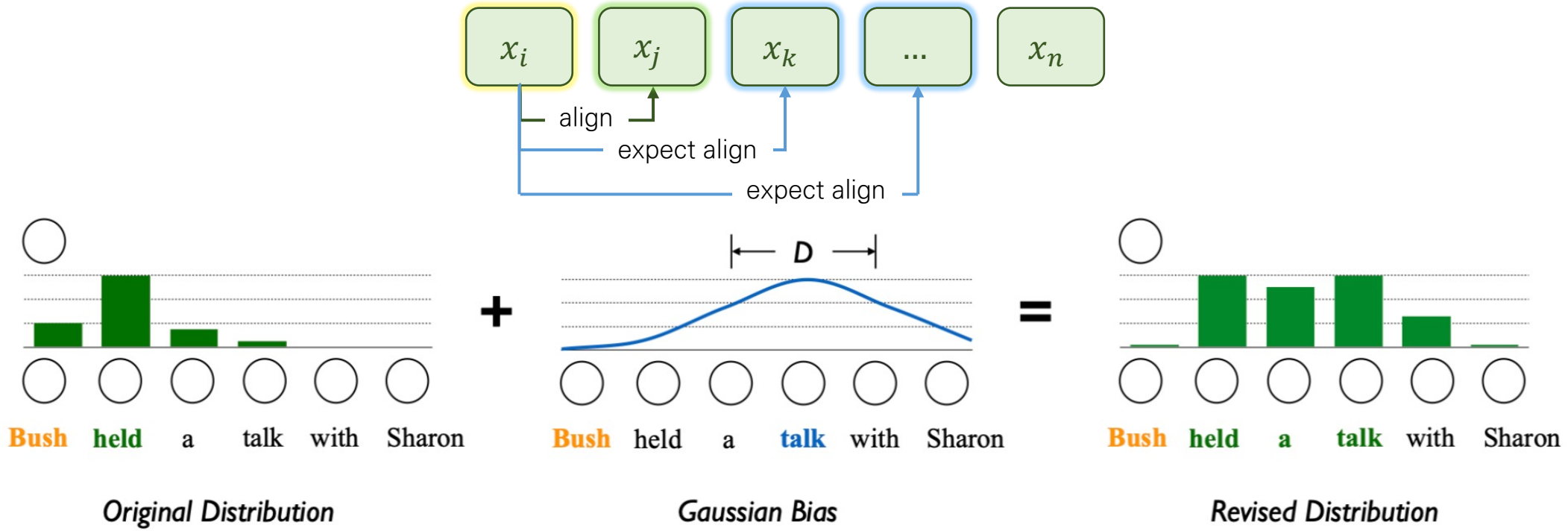




04 Attention with Prior

1) Prior that Models locality

- **Modeling Localness for Self-Attention Networks** (EMNLP 2018 , 96회 인용)
  - Self Attention은 global dependencies를 찾아낼 수 있음
  - 본 논문은 self attention network를 위한 model localness를 제안함
    → 이를 통해 useful local context를 찾아낼 수 있는 능력이 강화됨
  - Learnable Gaussian bias



https://blog.tensorflow.org/2020/02/matrix-compression-operator-tensorflow.html

04 Attention with Prior

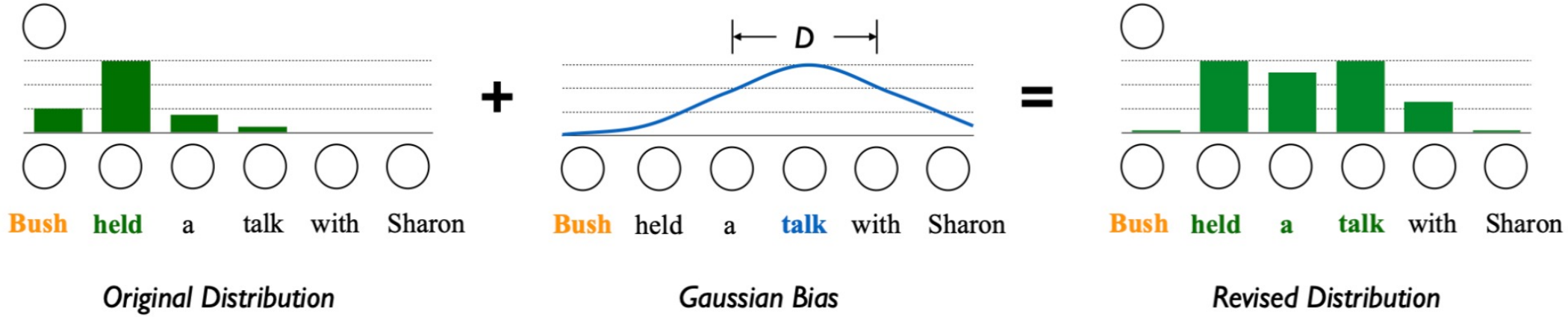
1) Prior that Models locality

- **Modeling Localness for Self-Attention Networks** (EMNLP 2018 , 96회 인용)
  - Self Attention은 global dependencies를 찾아낼 수 있음
  - 본 논문은 self attention network를 위한 model localness를 제안함
    → 이를 통해 useful local context를 찾아낼 수 있는 능력이 강화됨
  - Learnable Gaussian bias

Central position

$$G_{i,j} = -\frac{(j - P_i)^2}{2\sigma_i^2},$$

단어 x_j와
central position 사이의 tightness

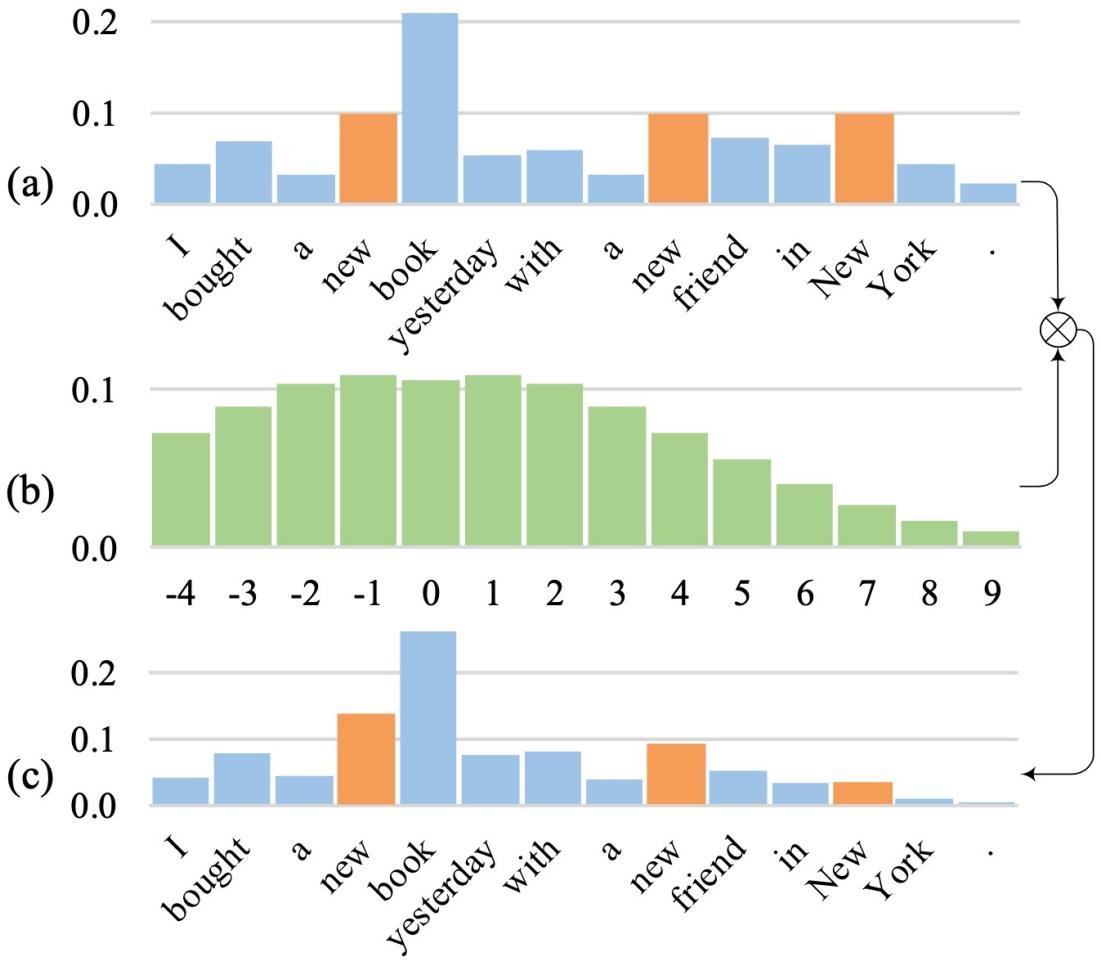


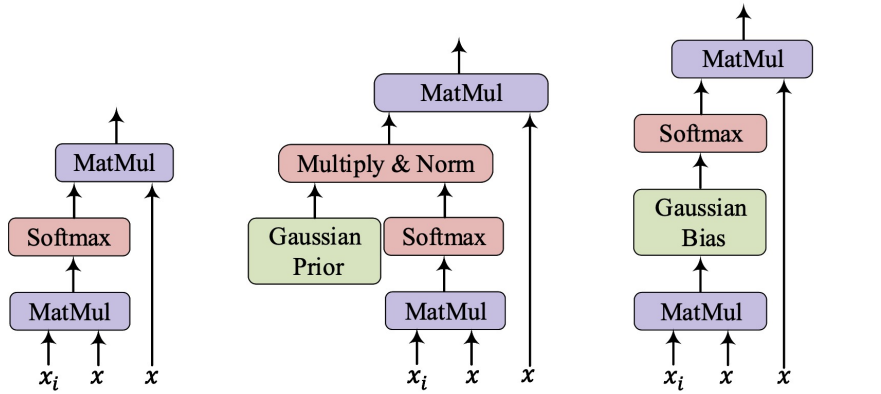
https://blog.tensorflow.org/2020/02/matrix-compression-operator-tensorflow.html

04 Attention with Prior

1) Prior that Models locality

- Gaussian Transformer: A Lightweight Approach for Natural Language Inference (AAAI 2019 , 29회 인용)





Original Gaussian prior extended self-attention

$$G_{ij} = -|w(i-j)^2 + b|,$$

i-th row in Q is the query $\mathbf{q}_i$

& 각 q_i의 중심을 i라 가정함

https://blog.tensorflow.org/2020/02/matrix-compression-operator-tensorflow.html

04 Attention with Prior

2) Prior from lower modules

- Transformer 구조에서 attention distribution은 adjacent layer와 비슷함
  - [Discuss] 레이어별 representation의 기저가 달라진다 ? 그렇다면 attention distribution은 ?!
- 이전 레이어에서 나오는 attention distribution을 prior로 사용하는 사례도 존재함

$$\hat{\mathbf{A}}^{(l)} = w_1 \cdot \mathbf{A}^{(l)} + w_2 \cdot g(\mathbf{A}^{(l-1)})$$

l 번째 레이어의 attention score

Previous score를 prior로 바꿔주는 함수

$g : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{n \times n}$

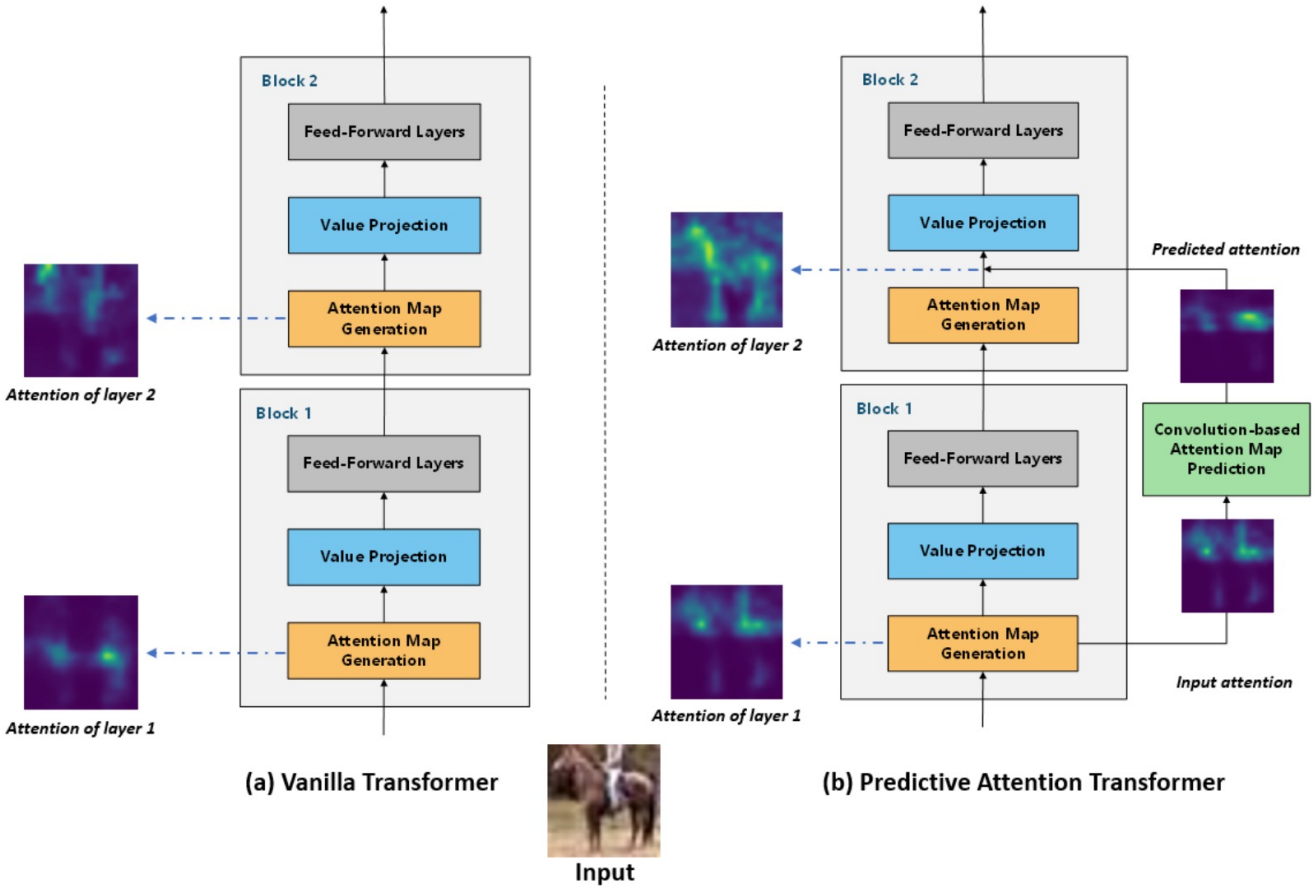
04 Attention with Prior

2) Prior from lower modules

- PREDICTIVE ATTENTION TRANSFORMER: IMPROVING TRANSFORMER WITH ATTENTION MAP PREDICTION (~~ICLR 2021 Reject~~, 1회 인용)

$$\hat{\mathbf{A}}^{(l)} = w_1 \cdot \mathbf{A}^{(l)} + w_2 \cdot g(\mathbf{A}^{(l-1)})$$

convolution



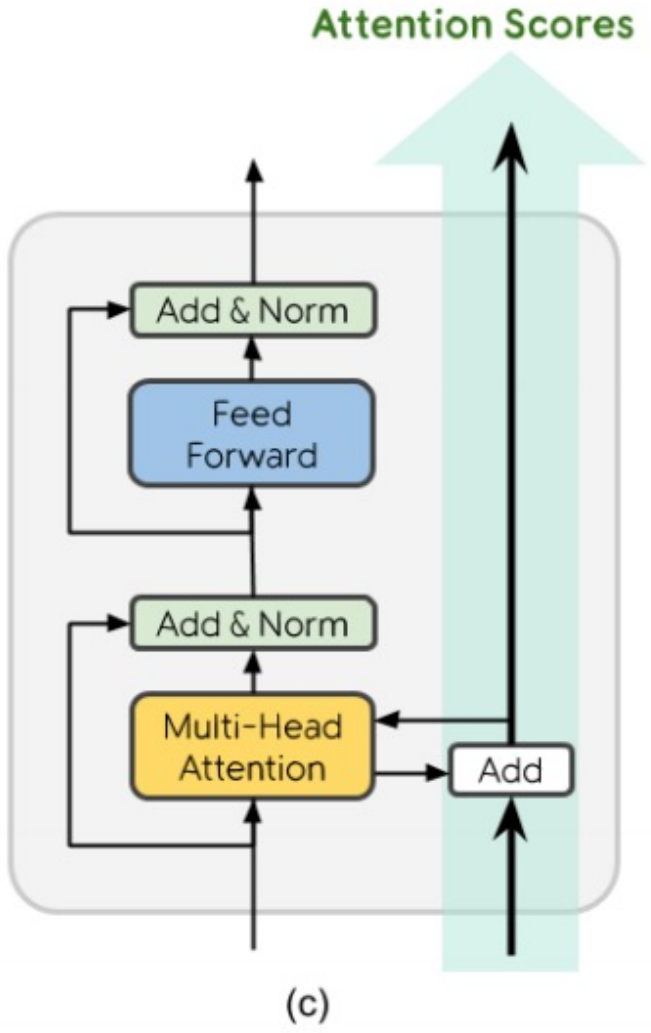
Decision: Reject

Comment: Multiple reviewers point out the interesting improvement to mix attention maps at different layers via convolution based prediction modules. This module is sufficient to show improvements only on encoder side while comparing to concurrent work Synthesizer. However, the novelty of the work is limited as compared to other papers and the results though improved did not convince the reviewers fully to gain a strong accept.

04 Attention with Prior

2) Prior from lower modules

- RealFormer: Transformer Likes Residual Attention (arXiv preprint, 5회 인용)



$$\hat{\mathbf{A}}^{(l)} = w_1 \cdot \mathbf{A}^{(l)} + w_2 \cdot g(\mathbf{A}^{(l-1)})$$

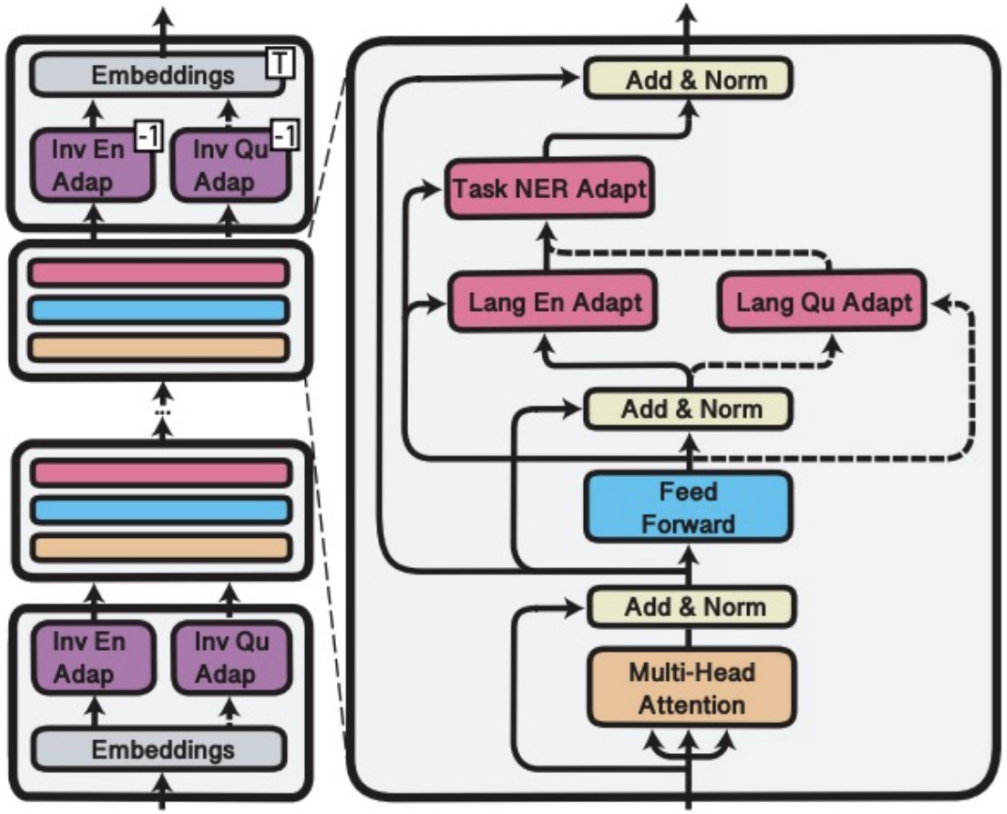
$w_1, w_2 = 1$ Identity map

- 현재 레이어에서 생성된 Generated attention score에 previous attention score를 directly 하게 더함
- 이 모습이 Residual skip connection과 유사함
- 동일한 과정을 attention map에서 진행했다고 이해할 수 있음

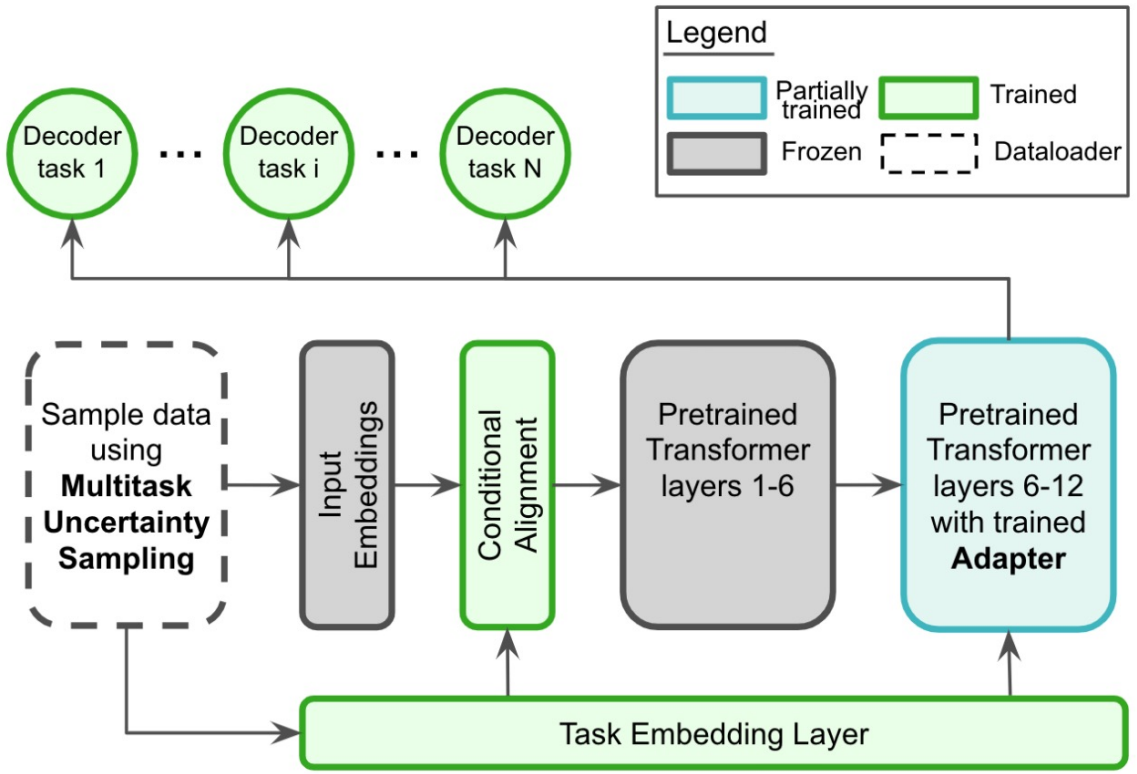
04 Attention with Prior

3) Prior as Multi-task Adapter

- CONDITIONALLY ADAPTIVE MULTI-TASK LEARNING: IMPROVING TRANSFER LEARNING IN NLP USING FEWER PARAMETERS & LESS DATA (ICLR 2021, 7회 인용)



내가 아는 Adapter / Pfeiffer et al. (2021)
모델에 약간의 capacity를 주어 성능 향상을 기대함

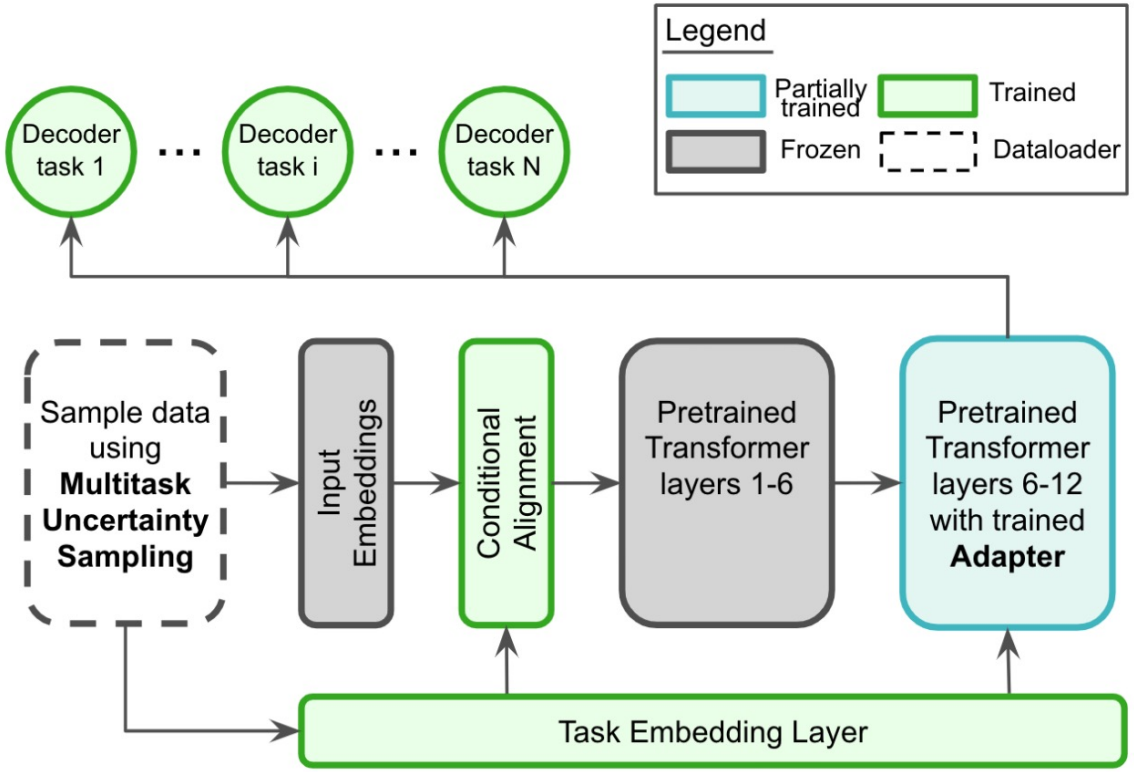


CAMTL의 Adapter

04 Attention with Prior

3) Prior as Multi-task Adapter

- CONDITIONALLY ADAPTIVE MULTI-TASK LEARNING: IMPROVING TRANSFER LEARNING IN NLP USING FEWER PARAMETERS & LESS DATA (ICLR 2021, 7회 인용)



CAMTL의 Adapter

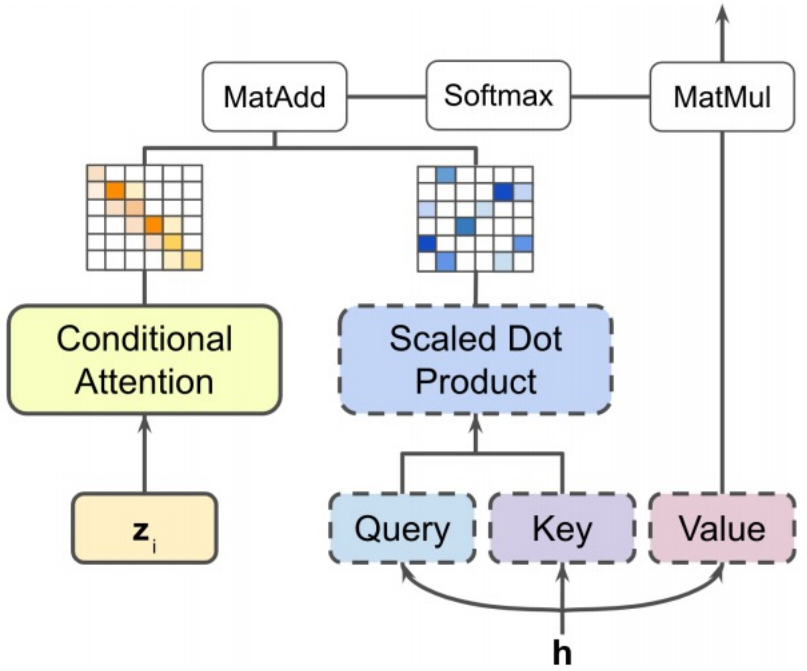


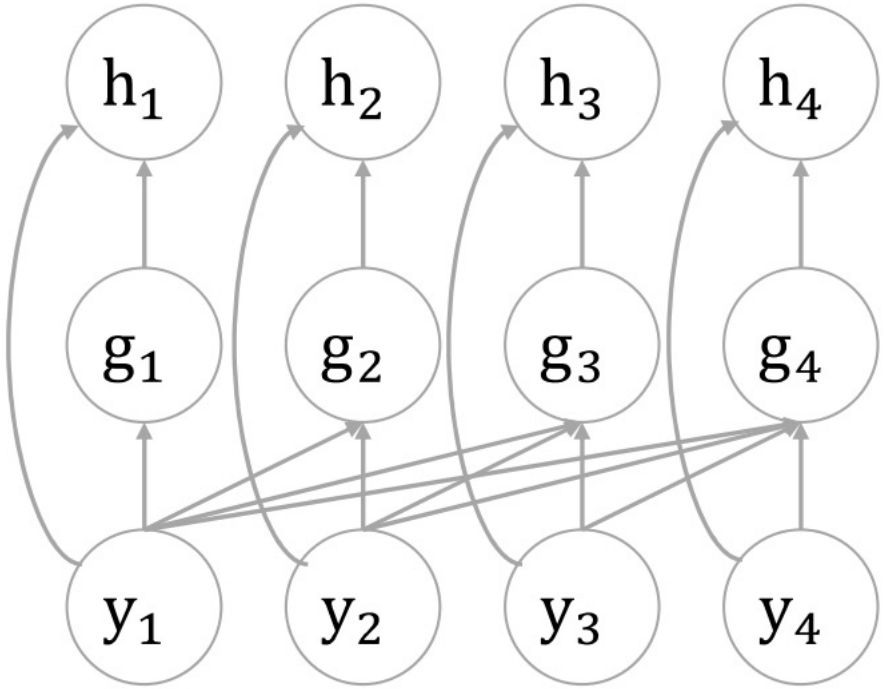
Figure 2: Conditional Attention Module

$$M(\mathbf{z}_i) = \bigoplus_{j=1}^{m} A'_j(\mathbf{z}_i), \quad A'_j(\mathbf{z}_i) = A_j\gamma_i(\mathbf{z}_i) + \beta_i(\mathbf{z}_i),$$

04 Attention with Prior

4) Attention with Only Prior – fuse하지 않고 prior attention distribution만 사용함

- Accelerating Neural Transformer via an Average Attention Network (ACL 2018, 79회 인용)



Gating Layer

모델 표현력 향상

Average Layer

History info. 요약

Input Layer

$$\mathbf{h}_j = \text{LayerNorm}\left(\mathbf{y}_j + \tilde{\mathbf{h}}_j\right)$$

$$\mathbf{i}_j, \mathbf{f}_j = \sigma\left(\mathbf{W}\left[\mathbf{y}_j; \mathbf{g}_j\right]\right)$$

$$\tilde{\mathbf{h}}_j = \mathbf{i}_j \odot \mathbf{y}_j + \mathbf{f}_j \odot \mathbf{g}_j$$

$$\mathbf{g}_j = \text{FFN}\left(\frac{1}{j}\sum_{k=1}^{j}\mathbf{y}_k\right)$$

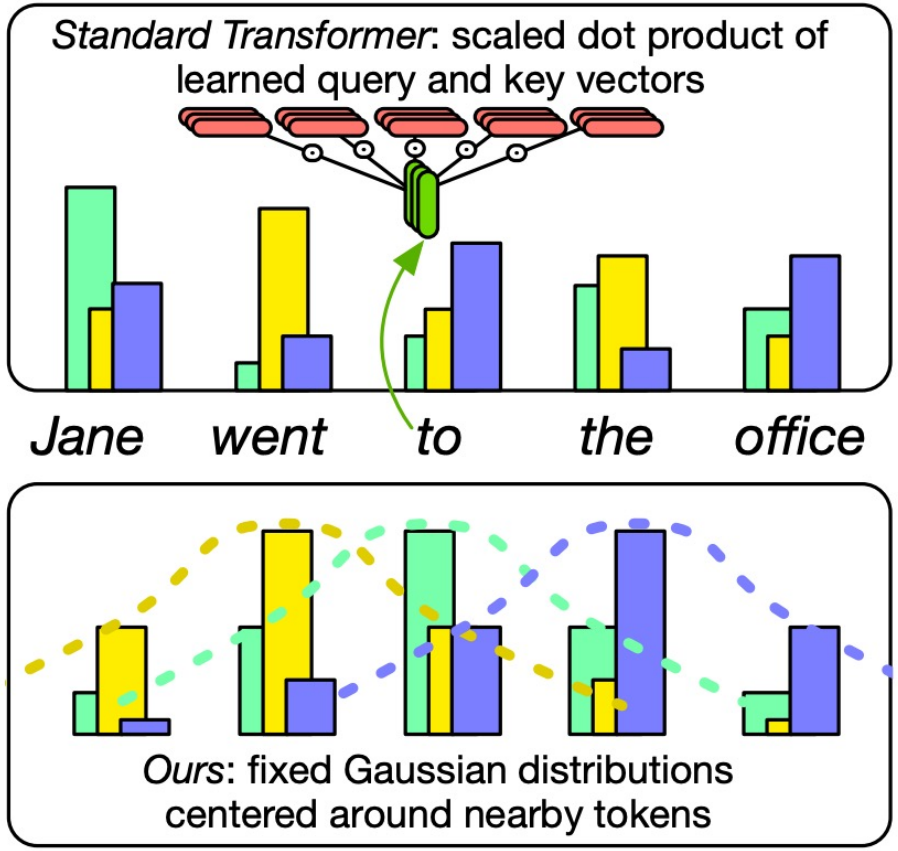
cumulative-average

: It builds up dependencies with previous input embeddings so that the generated representations are not independent of each other

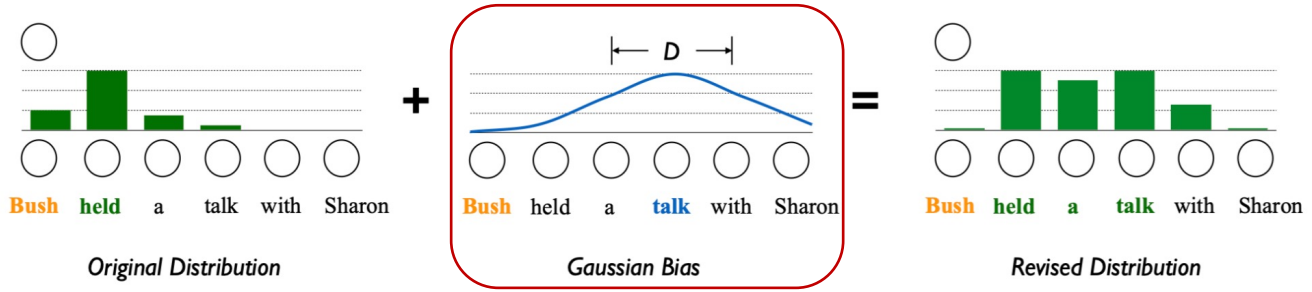
04 Attention with Prior

4) Attention with Only Prior

- Hard-Coded Gaussian Attention for Neural Machine Translation (ACL 2020, 16회 인용)



✓ Modeling Localness for Self-Attention Networks (EMNLP 2018 , 96회 인용)

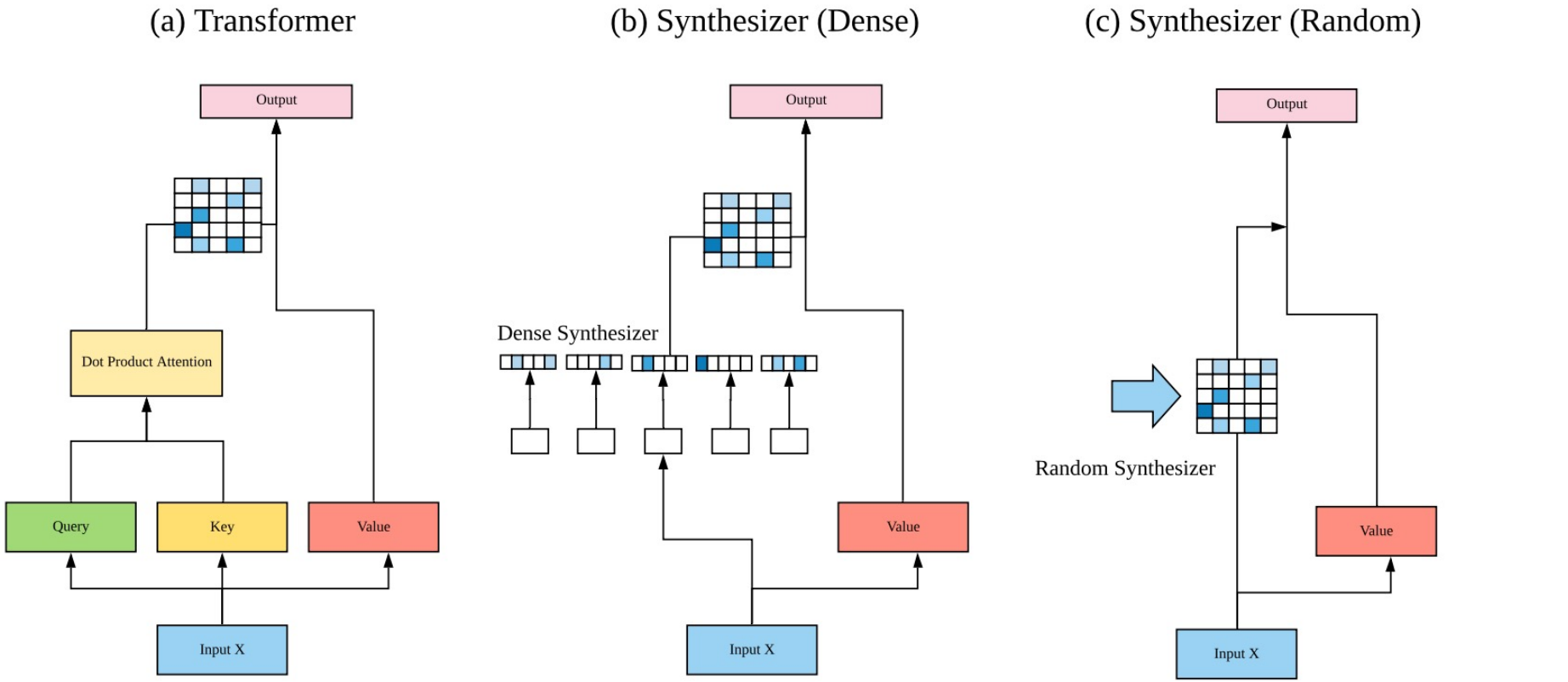


이 부분만 사용해서 Attention dist 계산함

04 Attention with Prior

4) Attention with Only Prior

- Synthesizer: Rethinking Self-Attention in Transformer Models (ICML 2021, 56회 인용)

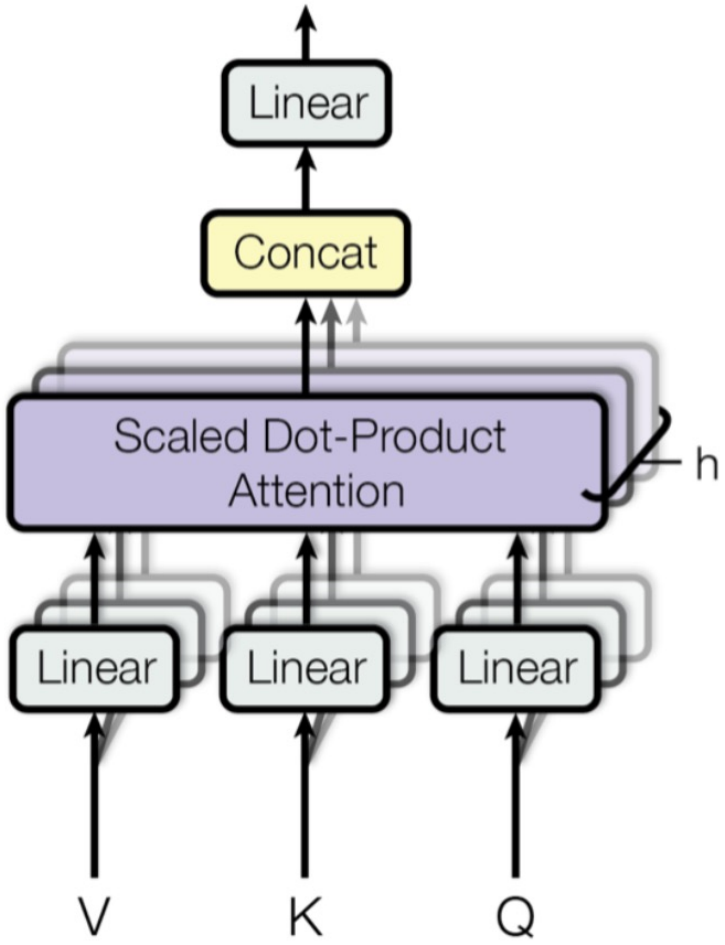


Attention scores from FFNN (only conditioned input)

Randomly initialized attention score

04 Improved Multi-Head Mechanism

introduction



Multi-head attention： different representation subspace 〉 different positions 〉 Joint

~~MHA 이대로 괜찮은가 ?~~

04 Improved Multi-Head Mechanism

1) Head Behavior Modeling

- Multi-Head Attention with Disagreement Regularization (EMNLP 2018, 79회 인용)

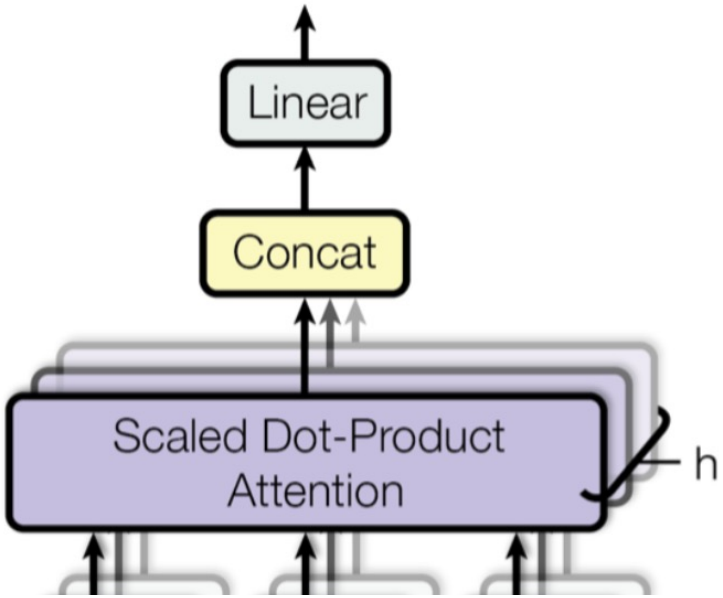
$$J(\theta) = \arg\max_{\theta} \{ \underbrace{L(\mathbf{y}|\mathbf{x};\theta)}_{likelihood} + \lambda * \underbrace{D(\mathbf{a}|\mathbf{x},\mathbf{y};\theta)}_{disagreement} \},$$

$$D_{subpace} = -\frac{1}{H^2}\sum_{i=1}^{H}\sum_{j=1}^{H}\frac{V^i \cdot V^j}{\|V^i\|\|V^j\|}. \quad D_{position} = -\frac{1}{H^2}\sum_{i=1}^{H}\sum_{j=1}^{H}|A^i \odot A^j|.$$

$$D_{output} = -\frac{1}{H^2}\sum_{i=1}^{H}\sum_{j=1}^{H}\frac{O^i \cdot O^j}{\|O^i\|\|O^j\|}.$$

04 Improved Multi-Head Mechanism

2) Multi-head with Restricted span



우리는 모든 헤드의 값을 동일하게 사용하지 않음 !

특정 헤드에 focus하는 경우가 있음

그렇다면,, 애초에 제한을 해볼까 ?!

- Locality

  : local constraint를 줌으로서

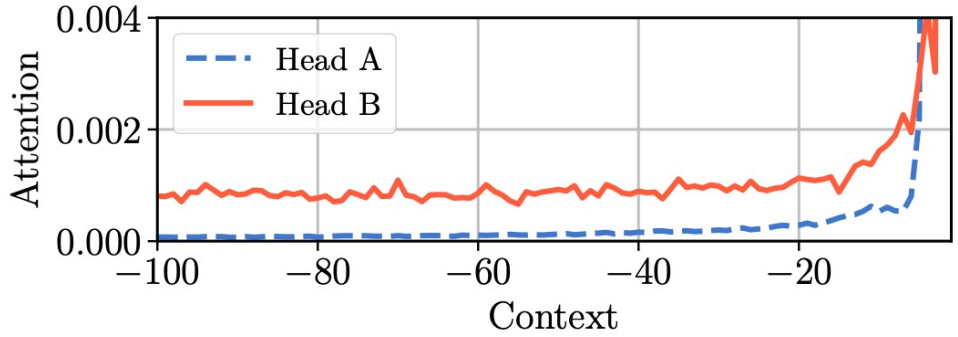
  이를 important prior로 사용 할 수 있음

- Efficiency
  : 긴 시퀀스에서는 메모리 효율이 올라감

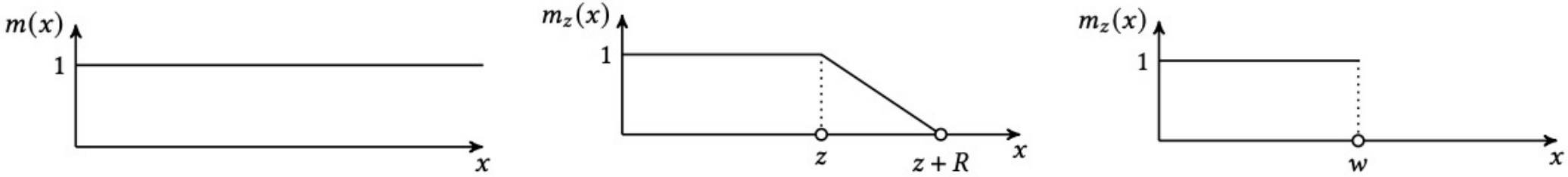
04 Improved Multi-Head Mechanism

2) Multi-head with Restricted span

- Adaptive Attention Span in Transformers. (ACL 2019, 109회 인용)
- Multi-Scale Self-Attention for Text Classification (AAAI 2020, 11회 인용)
  - Restricted span을 만들기 위해 mask function에 변화를 줌



서로 다른 Head는 상이한 Attention pattern을 가짐
애초에 head 별 self-attention에 mask를 다양하게 주자

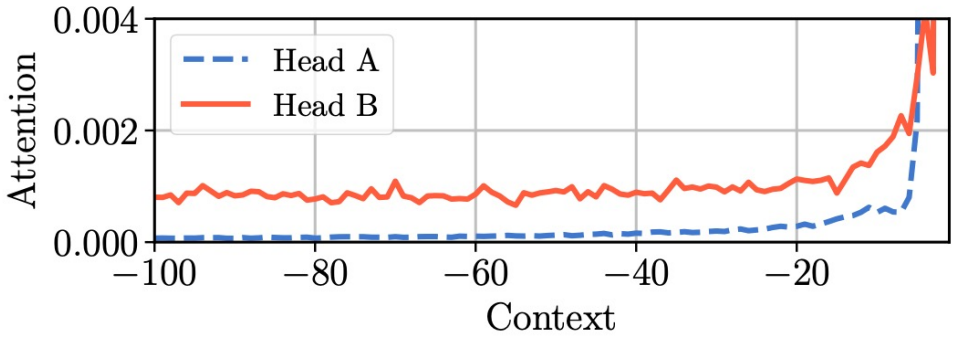


(a) mask function for vanilla attention (b) mask function for adaptive span (c) mask function for fixed span

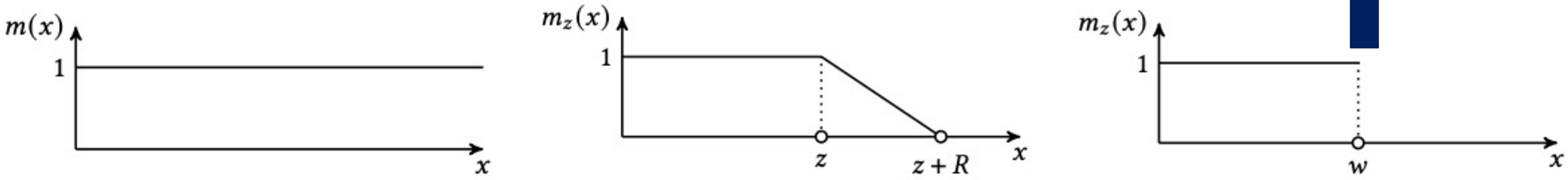
04 Improved Multi-Head Mechanism

2) Multi-head with Restricted span

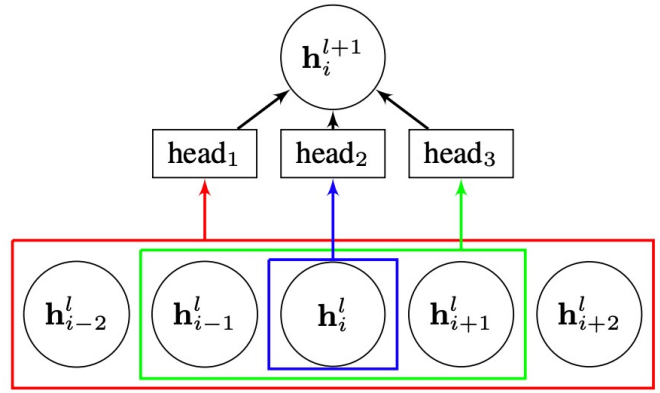
- Adaptive Attention Span in Transformers. (ACL 2019, 109회 인용)
- Multi-Scale Self-Attention for Text Classification (AAAI 2020, 11회 인용)
  - Restricted span을 만들기 위해 mask function에 변화를 줌



서로 다른 Head는 상이한 Attention pattern을 가짐
애초에 head 별 self-attention에 mask를 다양하게 주자

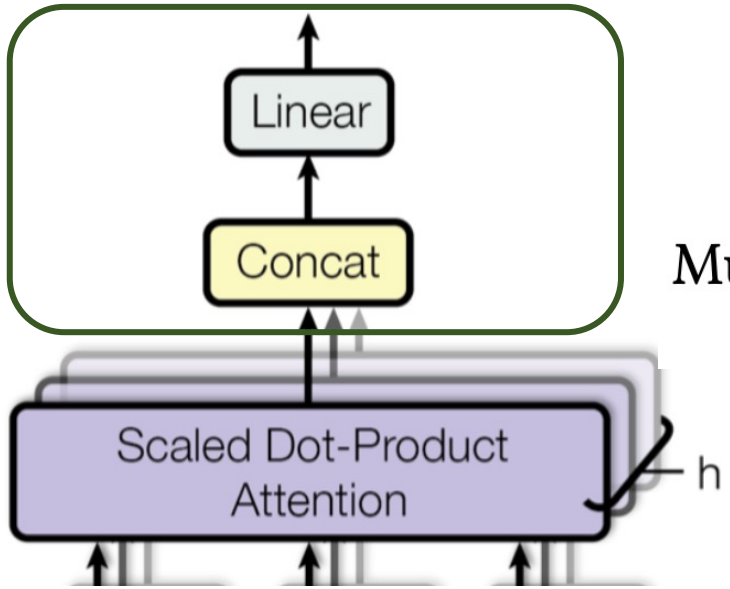


(a) mask function for vanilla attention (b) mask function for adaptive span (c) mask function for fixed span



04 Improved Multi-Head Mechanism

3) Multi-head with Refined Aggregation



$$\mathbf{W}^O = [\mathbf{W}_1^O; \mathbf{W}_2^O; \cdots; \mathbf{W}_H^O],$$

$$\text{MultiHeadAttn}(Q, K, V) = \sum_{i=1}^{H} \text{Attention}(Q\mathbf{W}_i^Q, K\mathbf{W}_i^K, V\mathbf{W}_i^V\mathbf{W}_i^O).$$

Value 묶을 때 한번 더 파라미터 넣어서
aggregate-by-summation paradigm으로 바꿔주자

~~별거 아님 주의~~

Concat과 linear projection은
너무나 간단한 aggregation일지도 . . ?

04 Improved Multi-Head Mechanism

4) Other modifications

- Fast Transformer Decoding: One Write-Head is All You Need (arXiv preprint 2019, 9회인용)
- Low-Rank Bottleneck in Multi-head Attention Models (ICML 2020, 6회 인용)
  - Head 개수가 증가하면 더 많은 자원이 필요하니 multihead에 LowRank constraint를 주어 사용하겠다는 연구

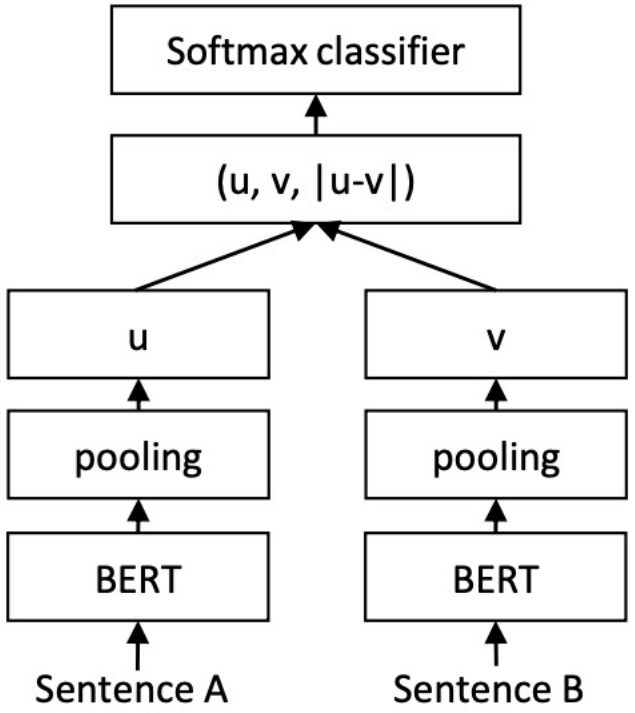
$$\mathrm{fixedhead}(\mathbf{X})_i = \mathbf{V}_v^i\mathbf{X} \cdot \mathrm{Softmax}\left[(\mathbf{V}_k^i\mathbf{X})^T(\mathbf{V}_q^i\mathbf{X})/\sqrt{d_p}\right] \in \mathbb{R}^{d_p \times n}$$

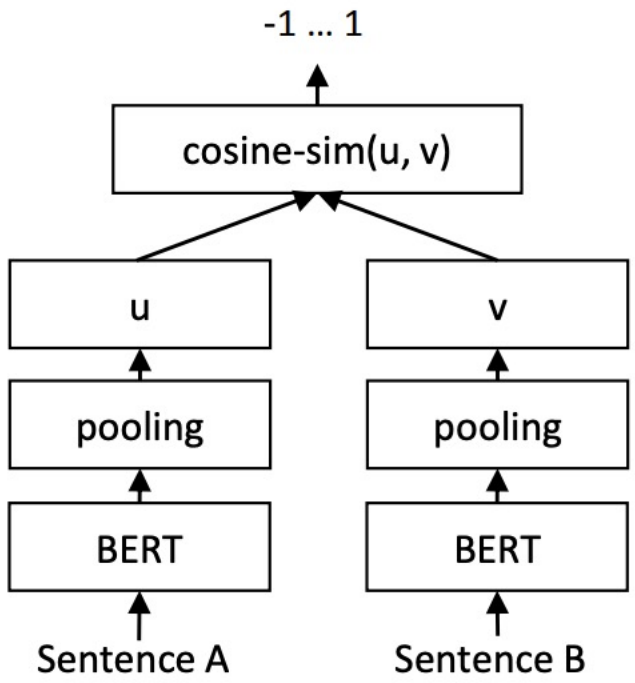
$$\mathrm{FixedMultiHead}(\mathbf{X}) = \mathrm{Concat}[\mathrm{fixedhead}(\mathbf{X})_1, \cdots, \mathrm{fixedhead}(\mathbf{X})_h] \in \mathbb{R}^{d_p \cdot h \times n}.$$

04 Discuss : [CLS]

4) Other modifications

- Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks (EMNLP 2019, 976회 인용)





04 Discuss : [CLS]

4) Other modifications

- Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks (EMNLP 2019, 976회 인용)

Model	STS12	STS13	STS14	STS15	STS16	STSb	SICK-R	Avg.
Avg. GloVe embeddings	55.14	70.66	59.73	68.25	63.66	58.02	53.76	61.32
Avg. BERT embeddings	38.78	57.98	57.98	63.15	61.06	46.35	58.40	54.81
BERT CLS-vector	20.16	30.01	20.09	36.88	38.08	16.50	42.63	29.19
InferSent - Glove	52.86	66.75	62.15	72.77	66.87	68.03	65.65	65.01
Universal Sentence Encoder	64.49	67.80	64.61	76.83	73.18	74.92	**76.69**	71.22
SBERT-NLI-base	70.97	76.53	73.19	79.09	74.30	77.03	72.91	74.89
SBERT-NLI-large	72.27	**78.46**	**74.90**	80.99	76.25	**79.23**	73.75	76.55
SRoBERTa-NLI-base	71.54	72.49	70.80	78.74	73.69	77.77	74.46	74.21
SRoBERTa-NLI-large	**74.53**	77.00	73.18	**81.85**	**76.82**	79.10	74.29	**76.68**