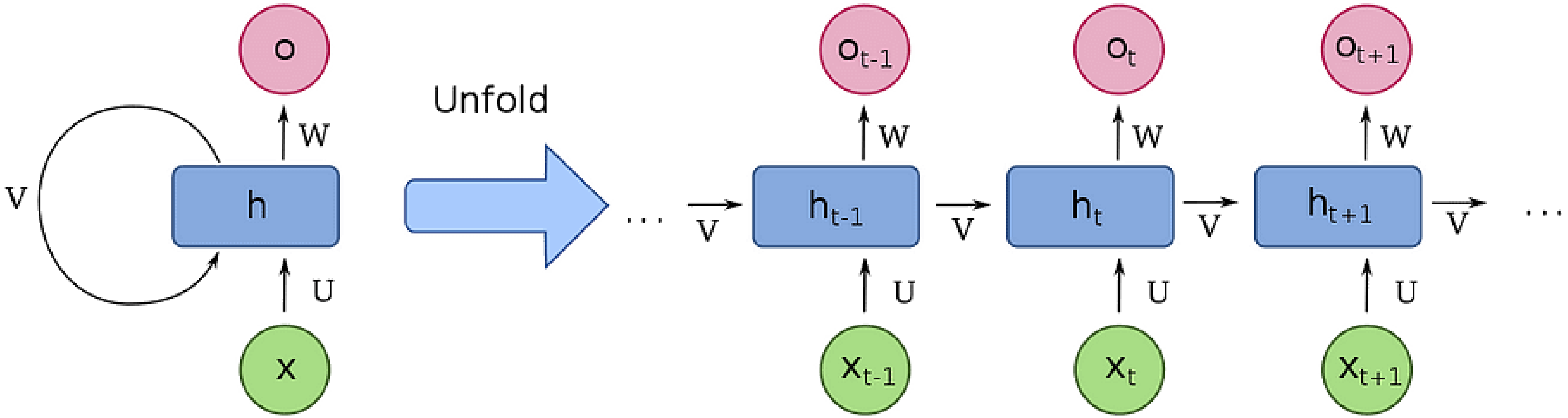
NLP Presentation

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

집현전 초급 오새찬

Overview

RNN Based NLP Model

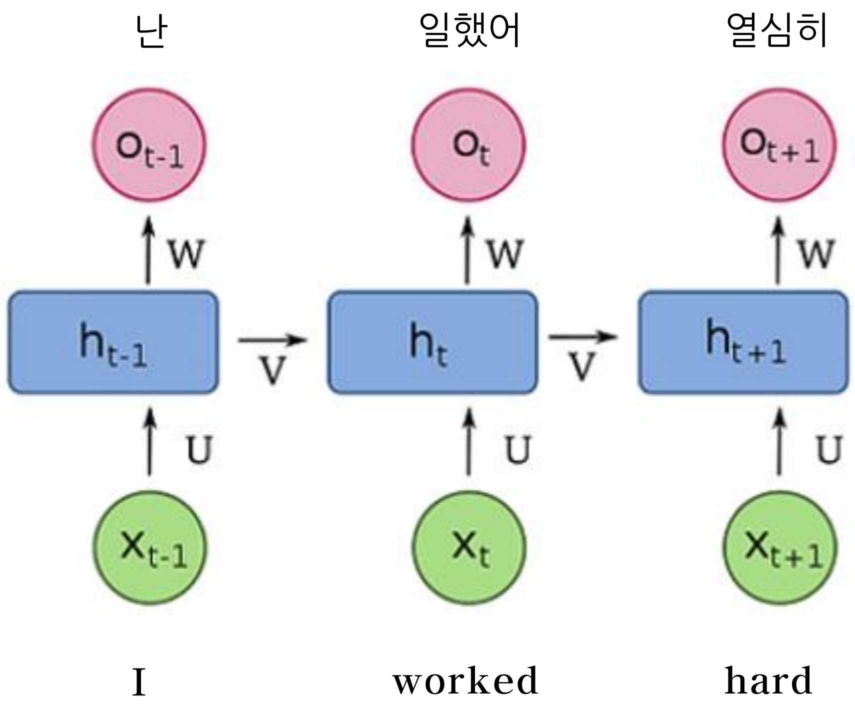


순환 신경망 이전까지의 개념은 알고있다고 가정할게요!

https://www.unite.ai/what-are-rnns-and-lstms-in-deep-learning/

Overview

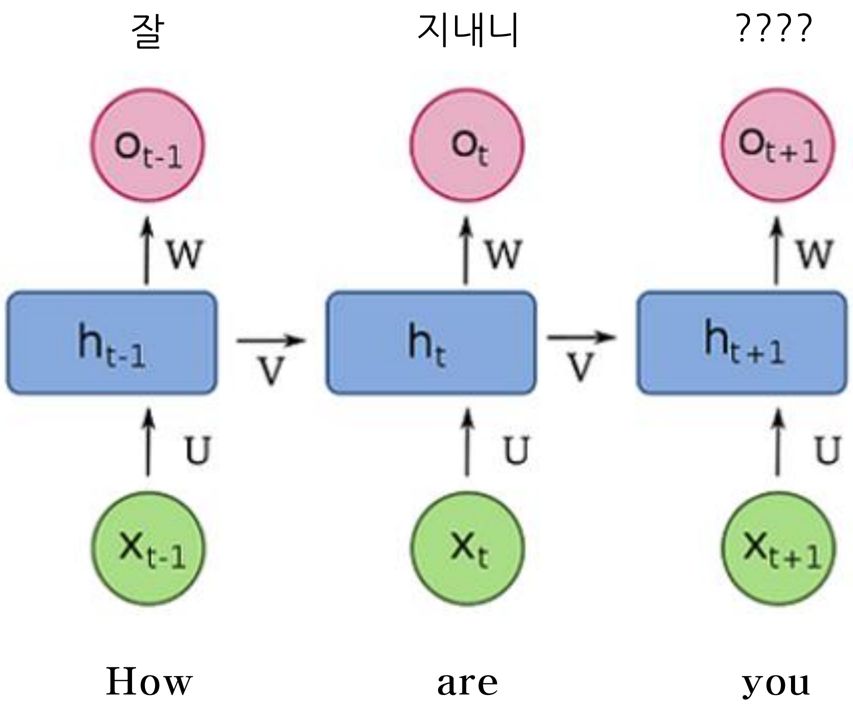
RNN Based NLP Model



https://www.unite.ai/what-are-rnns-and-lstms-in-deep-learning/

Overview

RNN Based NLP Model



How -> 잘 (?)

출력된 문장의 순서는 잘 지켜지는가?

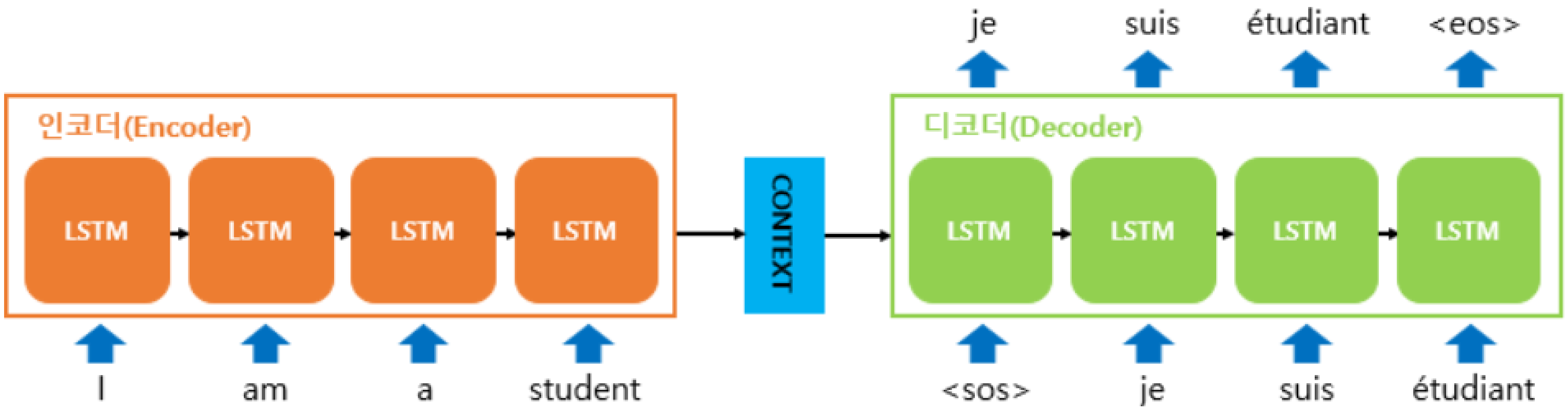
고질적인 기울기 소실 문제

3개의 입력에서 2개의 출력

https://www.unite.ai/what-are-rnns-and-lstms-in-deep-learning/

Overview

Seq2Seq Model

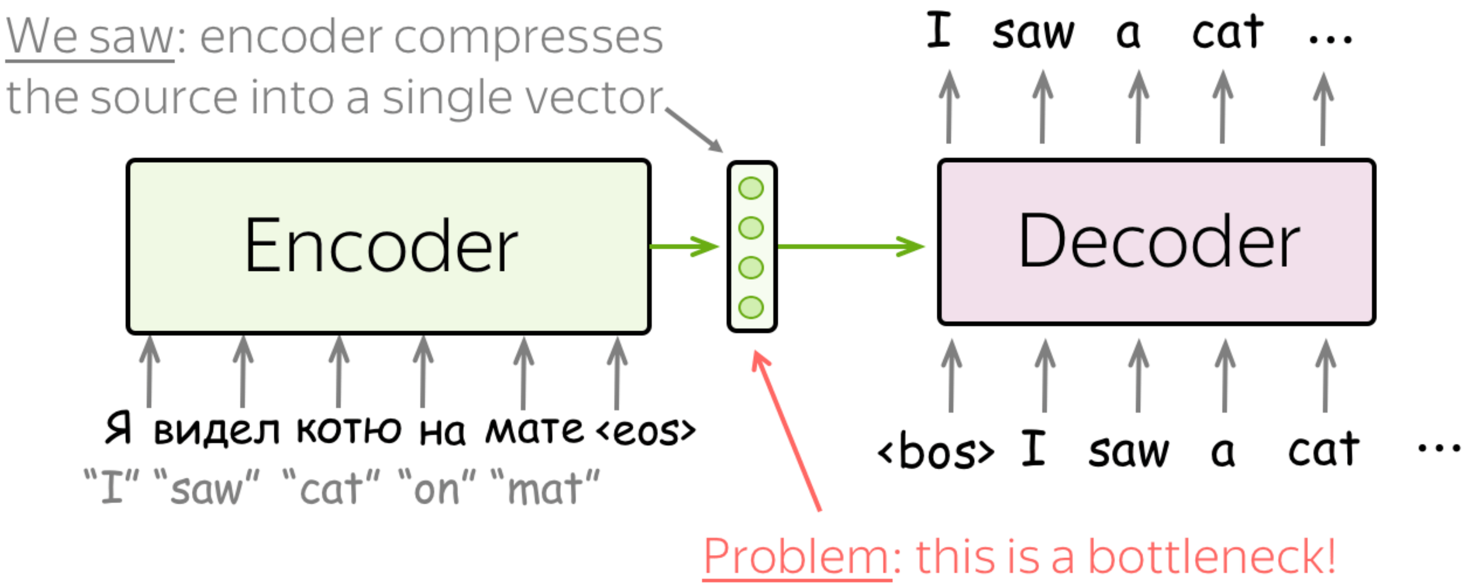


입력을 해석하는 부분(인코더)과 출력하는 부분을 구분(디코더)

고정된 크기의 문맥 벡터를 생성 후 전달

https://wikidocs.net/24996

Overview

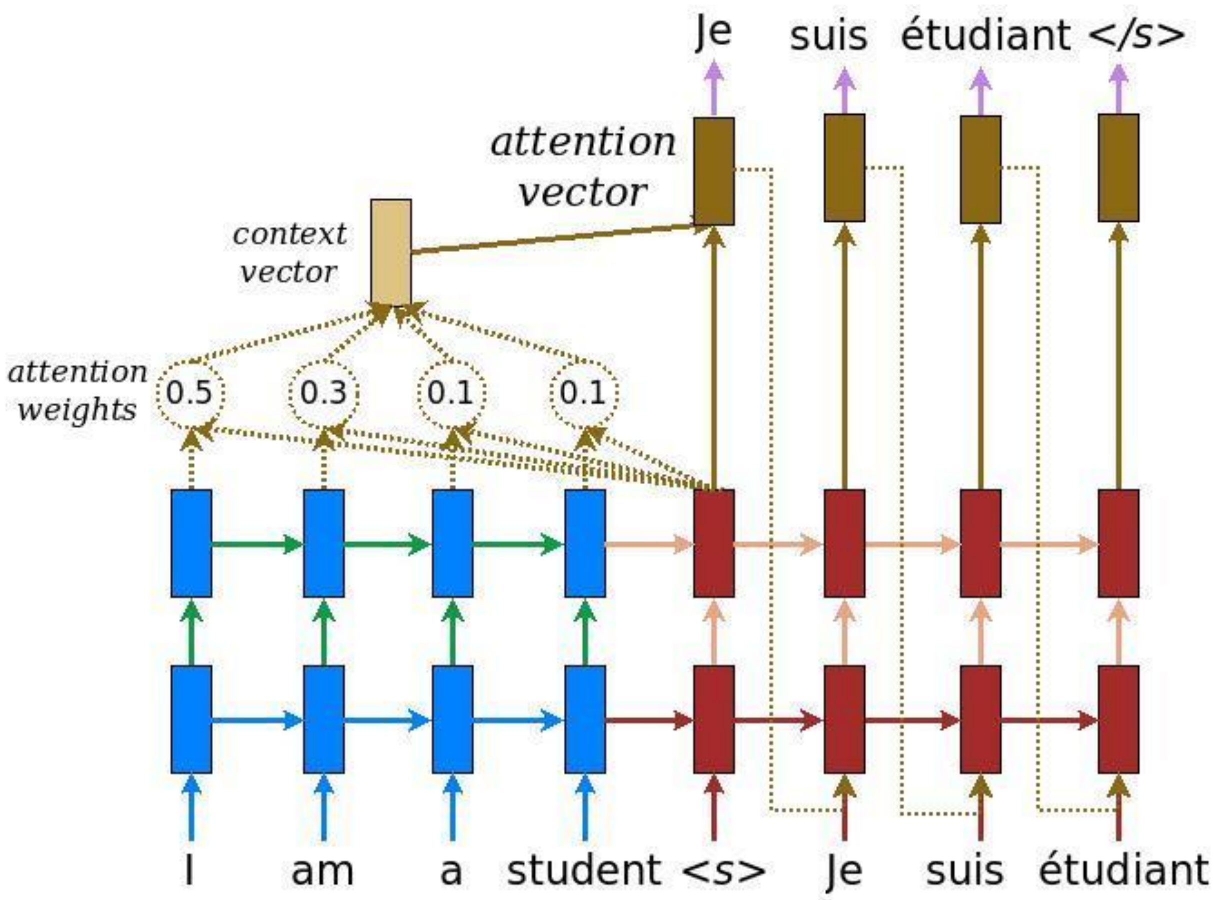


- Bottleneck problem
  - "We show that the neural machine translation performs relatively well on short sentences without unknown words, but **its performance degrades rapidly as the length of the sentence and the number of unknown words increase.**" (Cho et al. 2014)

https://www.slideshare.net/JaeHoJang7/attention-scores-and-mechanisms
https://lena-voita.github.io/nlp_course/seq2seq_and_attention.html#attention_intro

Overview

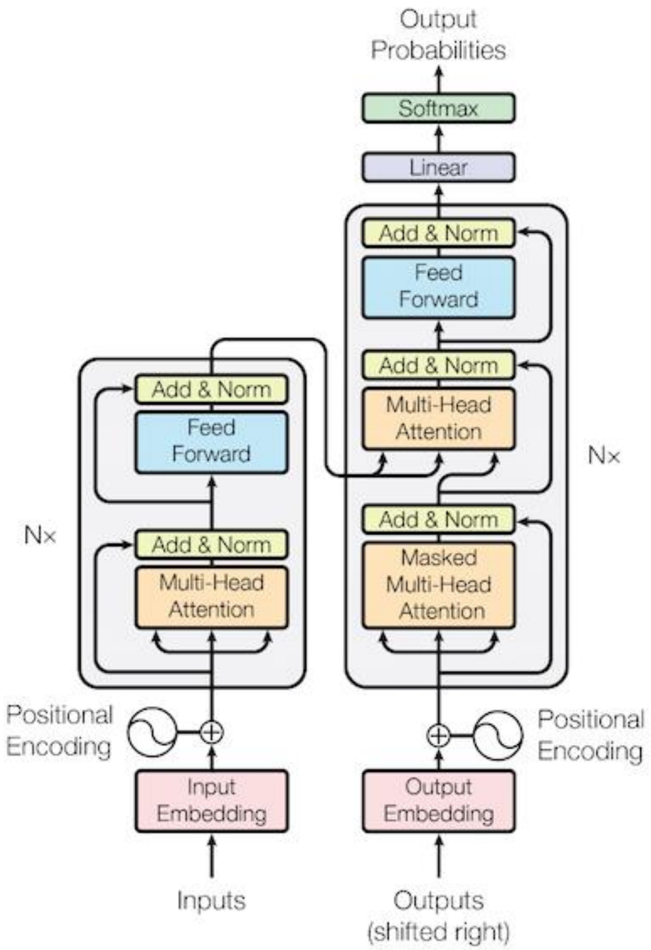
Attention Mechanism



https://medium.com/syncedreview/a-brief-overview-of-attention-mechanism-13c578ba9129

Overview

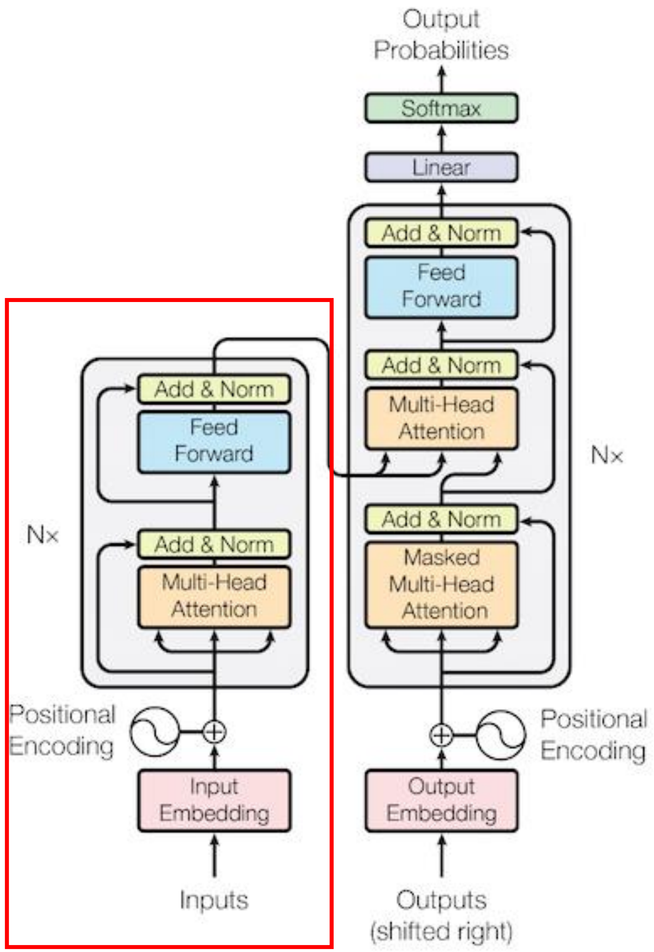
Transformer



https://arxiv.org/pdf/1706.03762.pdf

Overview

Transformer



https://arxiv.org/pdf/1706.03762.pdf

Mechanism

BERT input representation

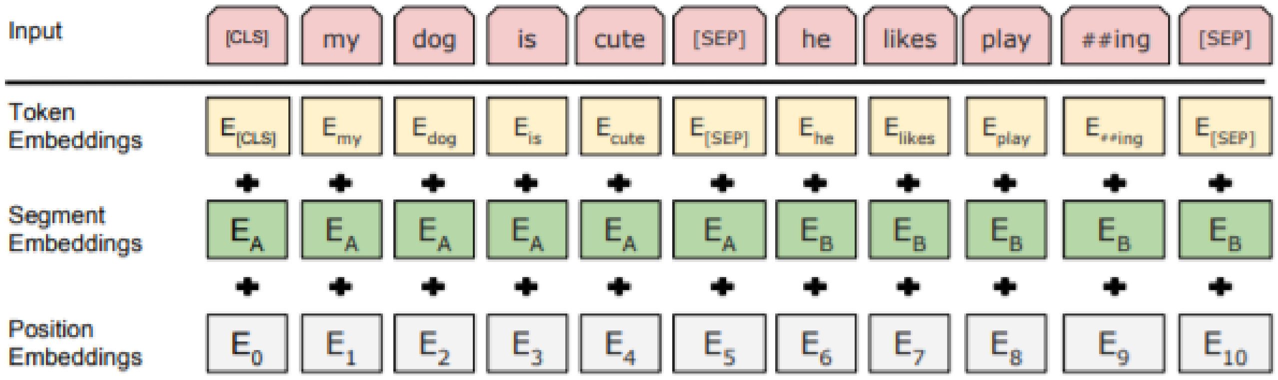


Figure 2: BERT input representation. The input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings.

our input representation is able to unambiguously represent both a single sentence and a pair of sentences

https://arxiv.org/pdf/1810.04805.pdf

Mechanism

BERT input representation

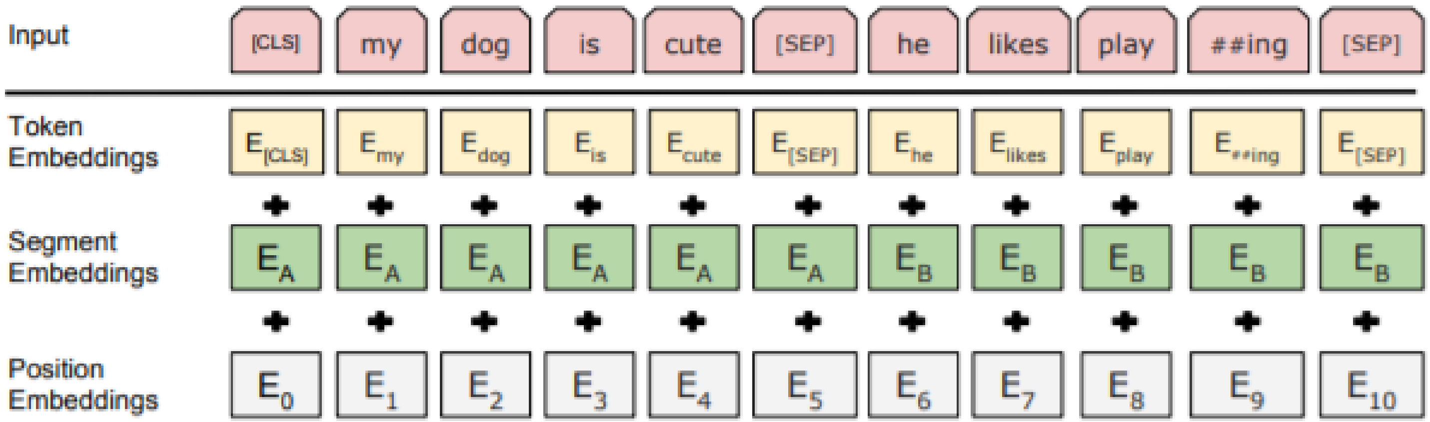


Figure 2: BERT input representation. The input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings.

https://arxiv.org/pdf/1810.04805.pdf

Mechanism

I → T.E [0.1, 0.3, 0.7]

S.E [0.8, 0.2, 0.5] → [2.0, 2.0, 1.3]

P.E [1.1, 1.5, 0.1]

am

hangry

[CLS] – [0.7, 0.8, 1.1]

I – [0.8, 0.1, 1.5]

am

hangry

[SEP]

Attention

is

all

you

need

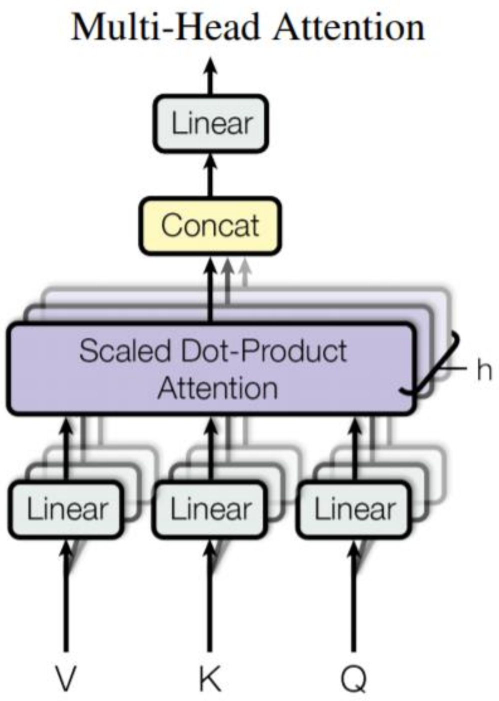
[SEP] – [1.5, 1.4, 2.4]

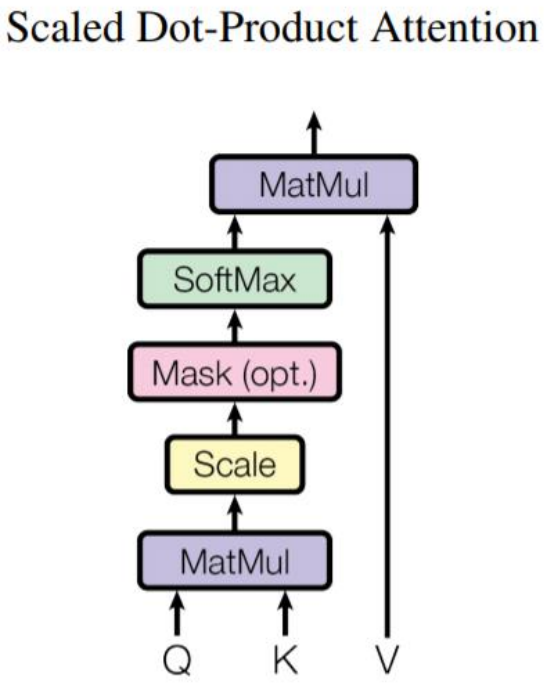
→ size(11, 3)

실제 BERT 모델에서는

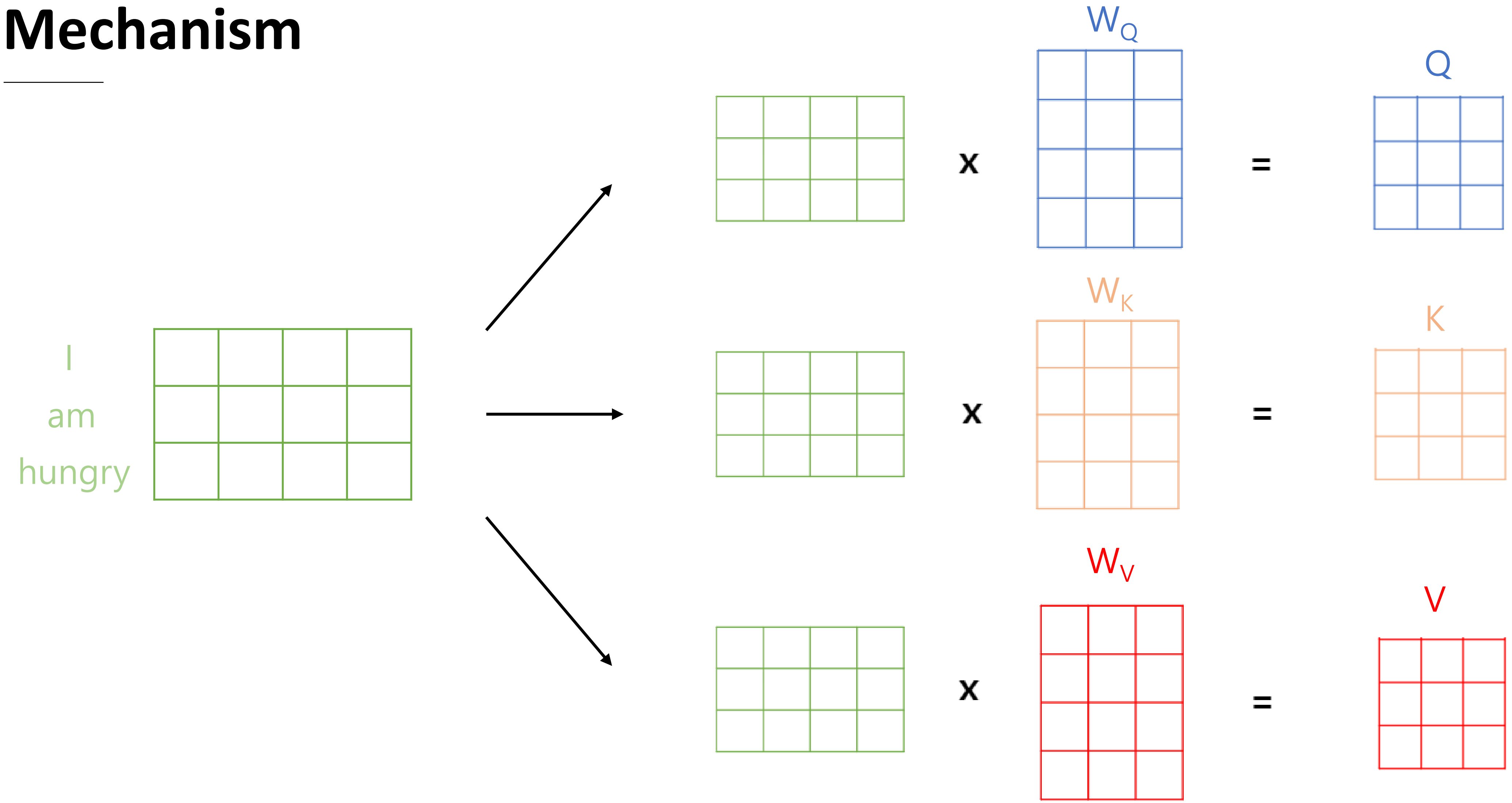
768차원, 1024차원

Mechanism

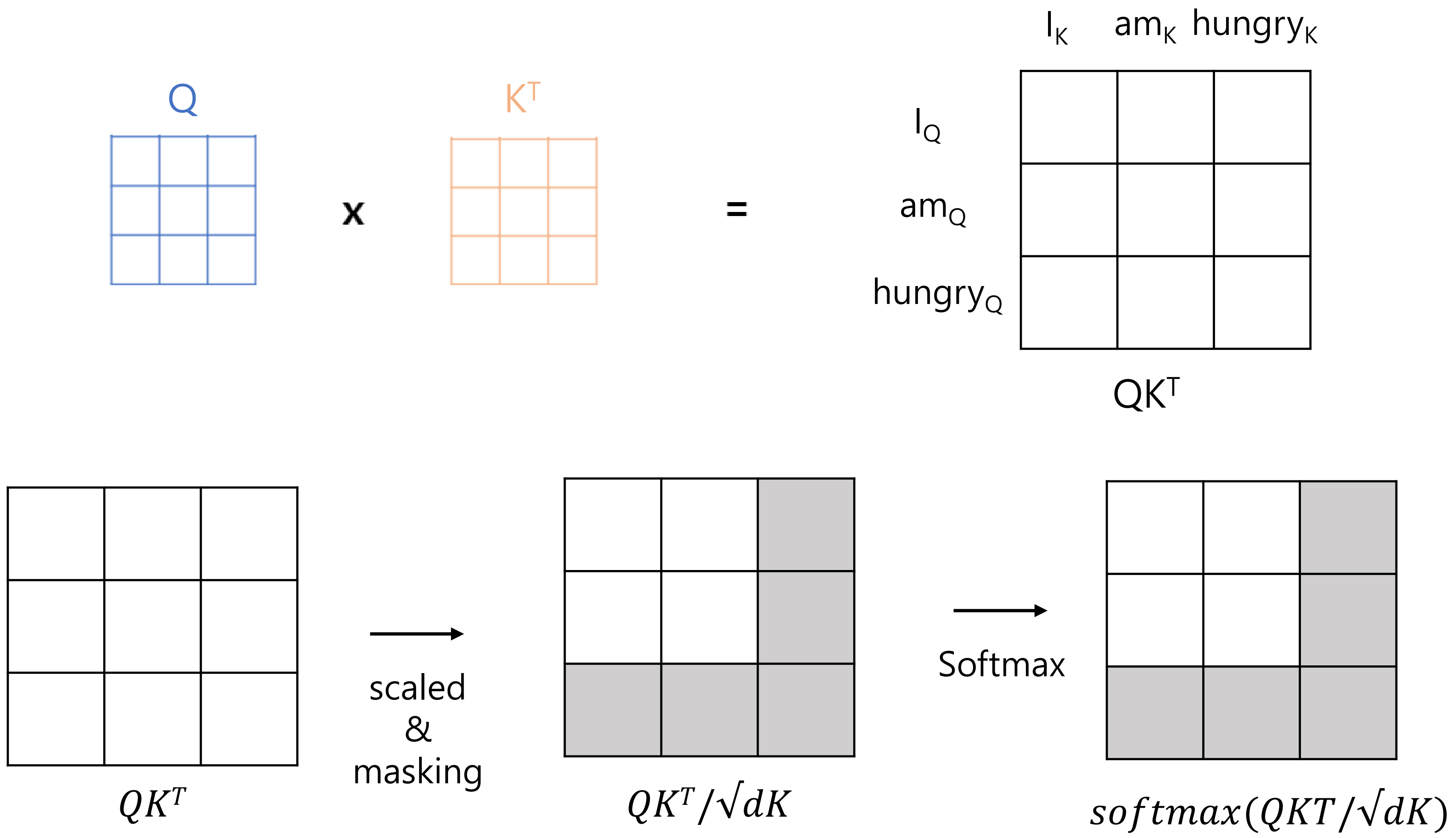

Multi-Head Attention
Linear
Concat
Scaled Dot-Product Attention
h
Linear
Linear
Linear
V
K
Q


Scaled Dot-Product Attention
MatMul
SoftMax
Mask (opt.)
Scale
MatMul
Q
K
V

Mechanism

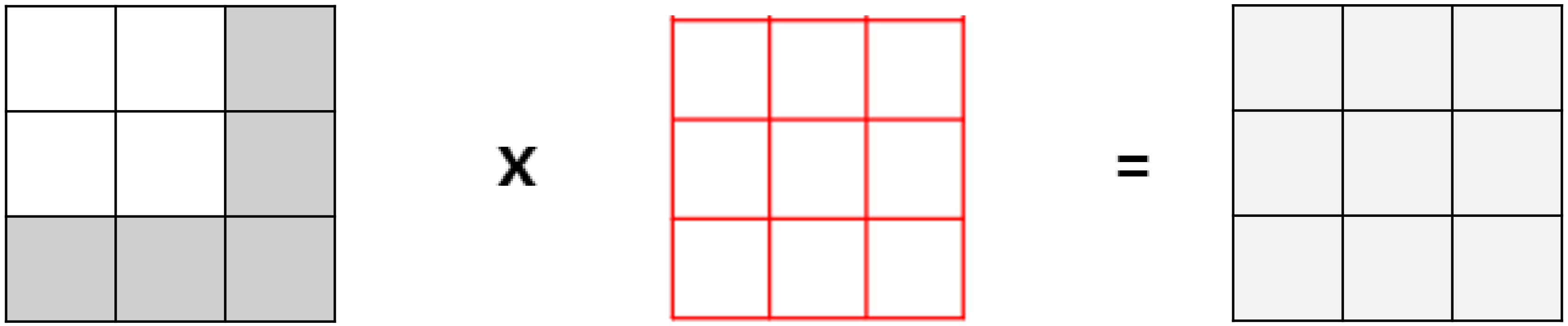


Mechanism



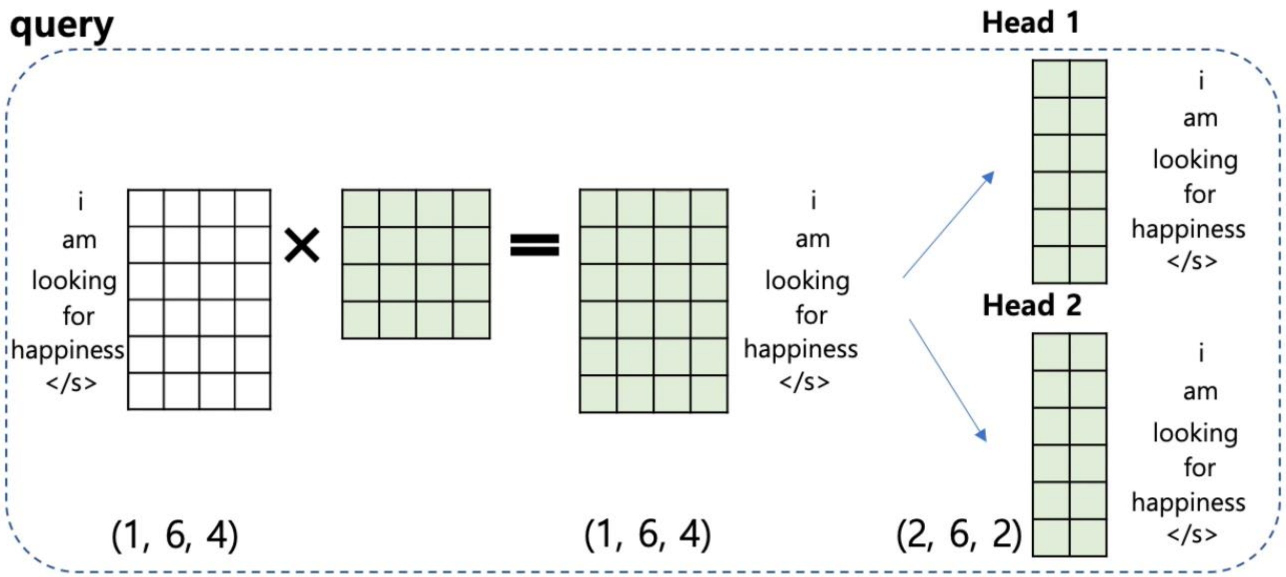
Q
K^T
x
=
I_K
am_K
$hungry_K$
I_Q
am_Q
$hungry_Q$
QK^T
QK^T
scaled
&
masking
$QK^T/\sqrt{d}K$
Softmax
$softmax(QKT/\sqrt{d}K)$

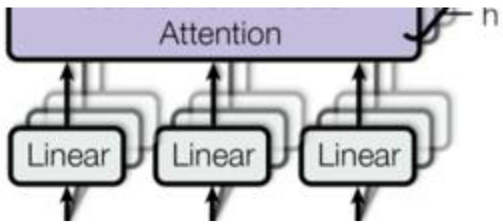
Mechanism



$softmax(QKT/\sqrt{dK})$ V $Attention\ Vector$

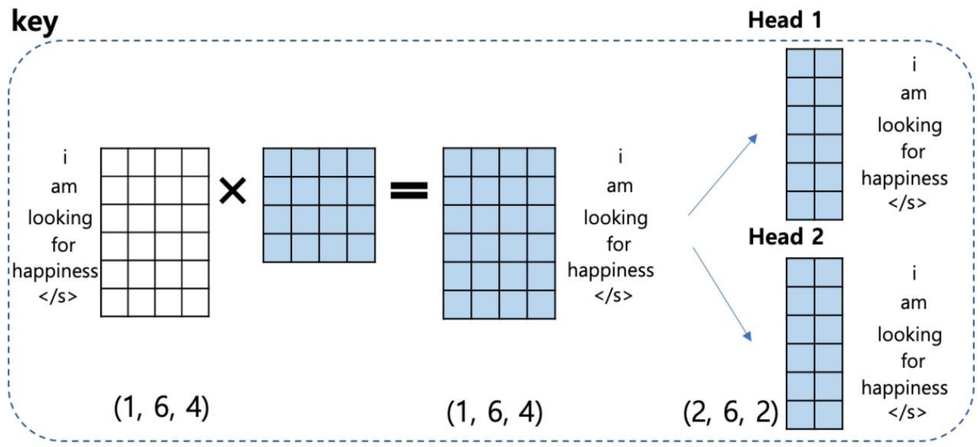
Mechanism

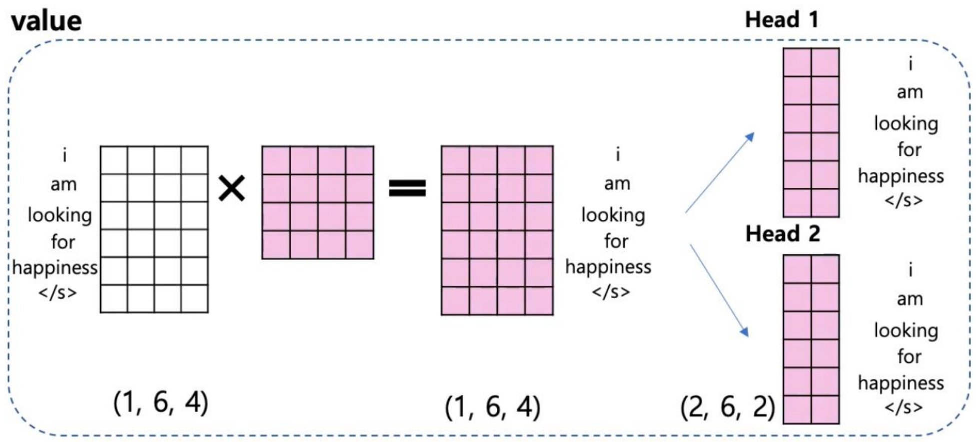




http://jalammar.github.io/illustrated-transformer/

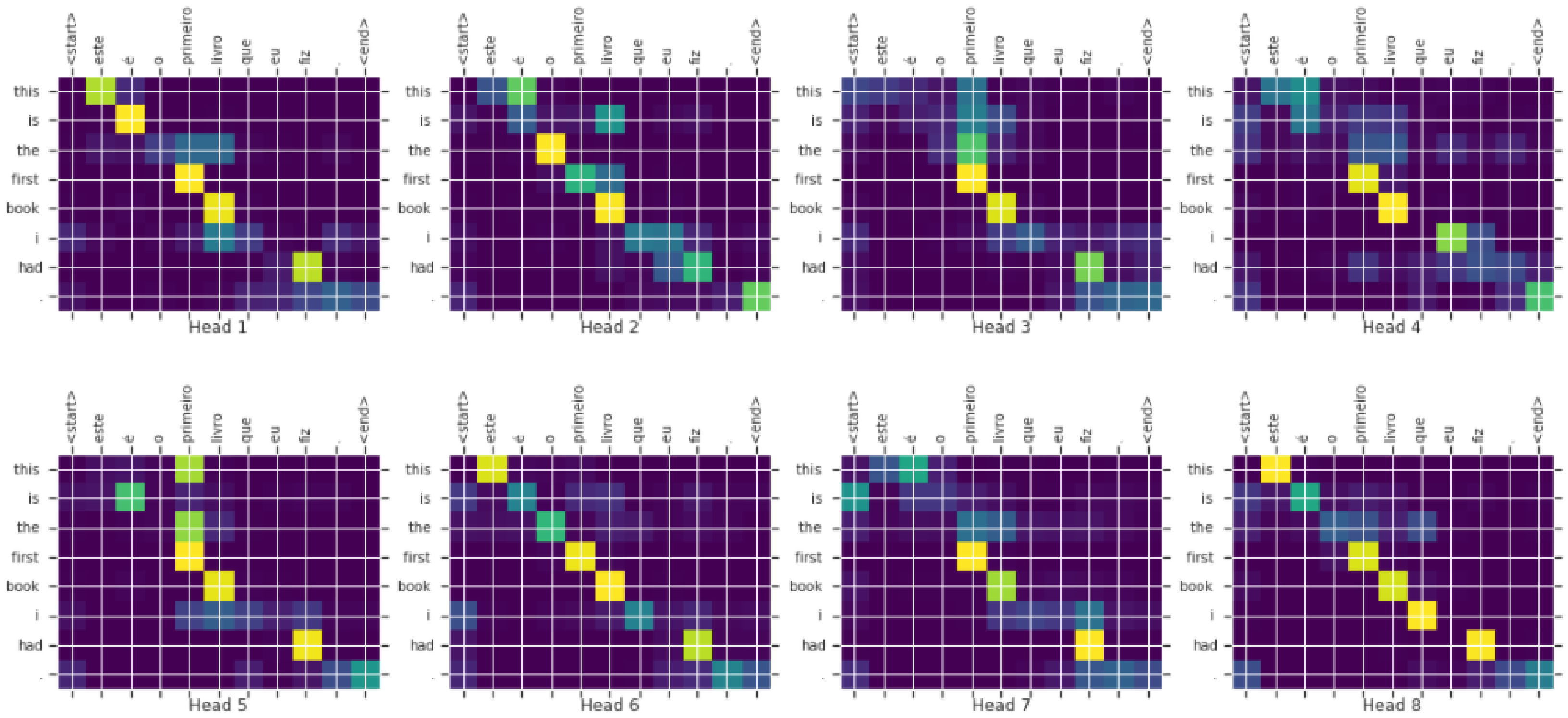
Mechanism





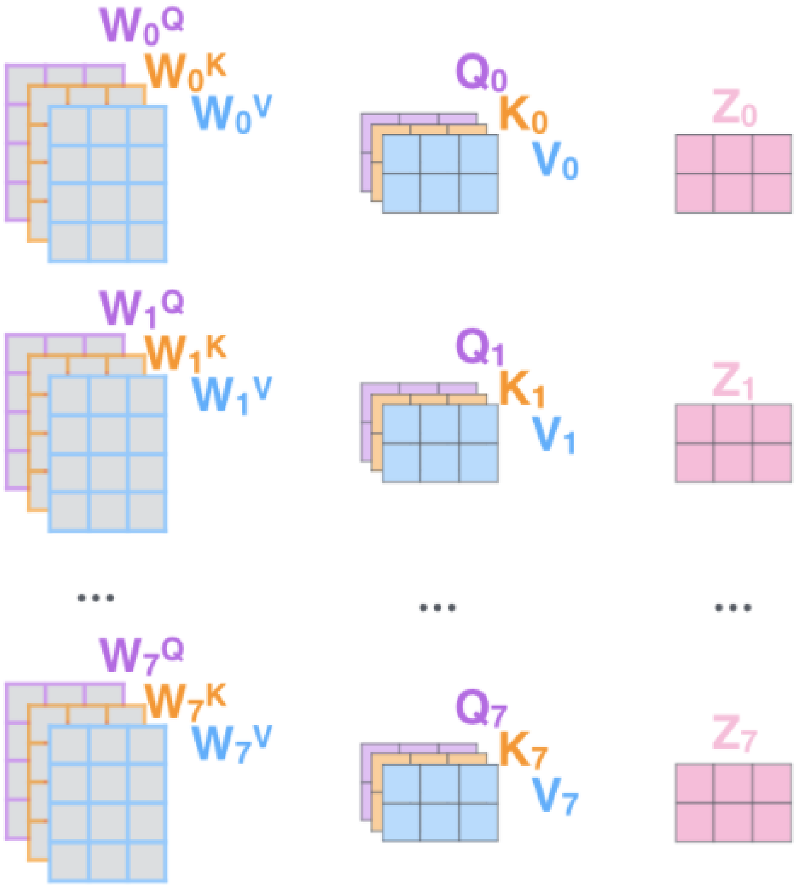
http://jalammar.github.io/illustrated-transformer/

Mechanism

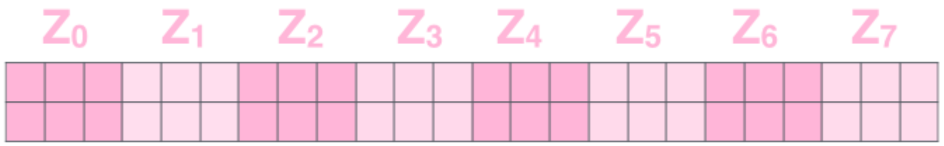


https://www.tensorflow.org/tutorials/text/transformer

Mechanism



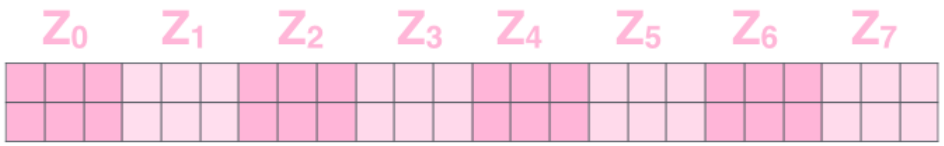
1) Concatenate all the attention heads

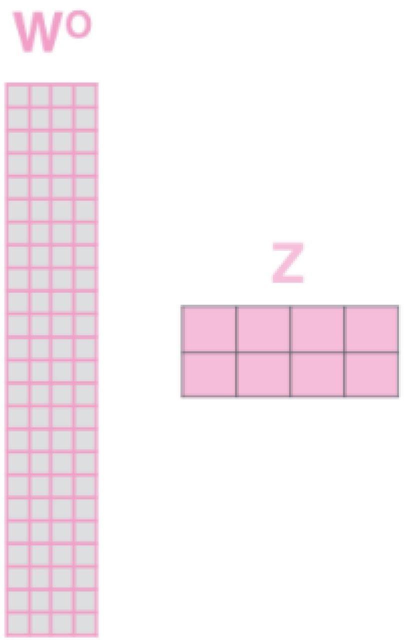


http://jalammar.github.io/illustrated-transformer/

Mechanism

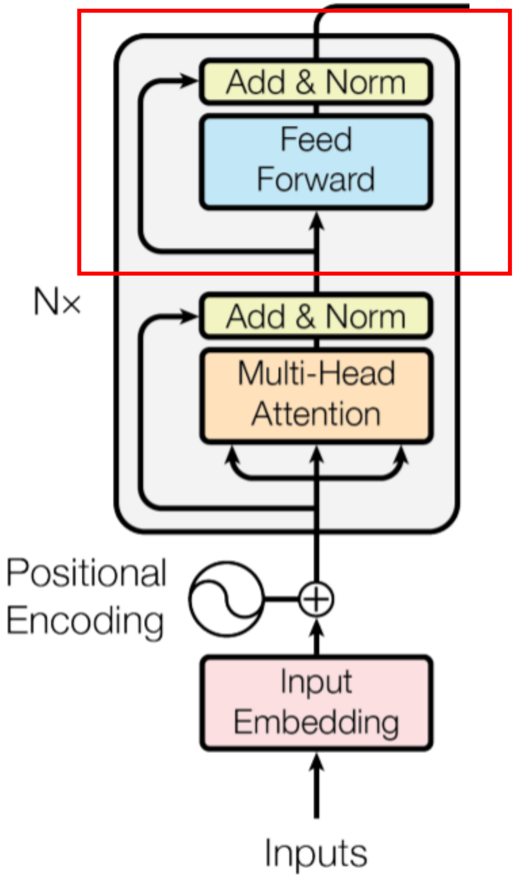
1) Concatenate all the attention heads

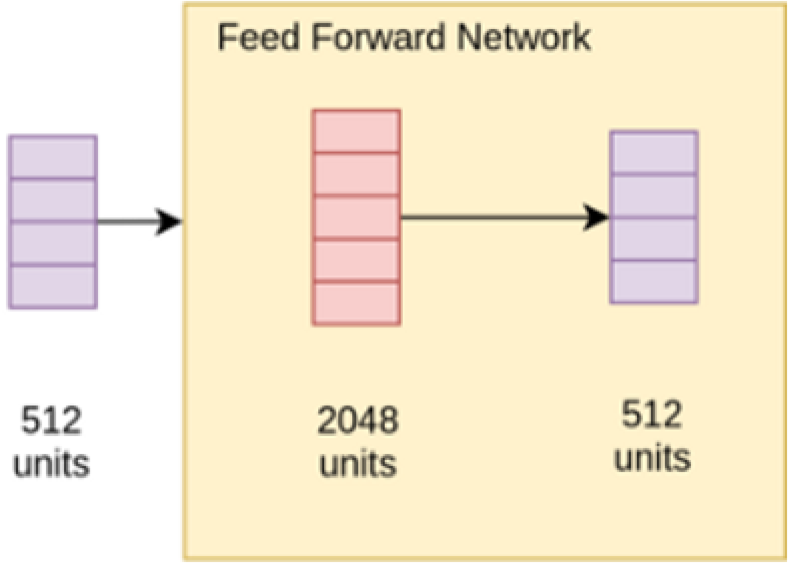




http://jalammar.github.io/illustrated-transformer/

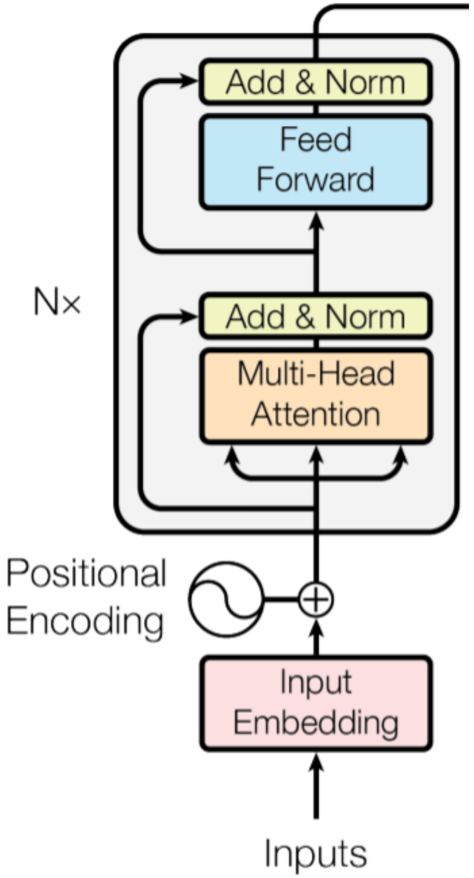
Mechanism

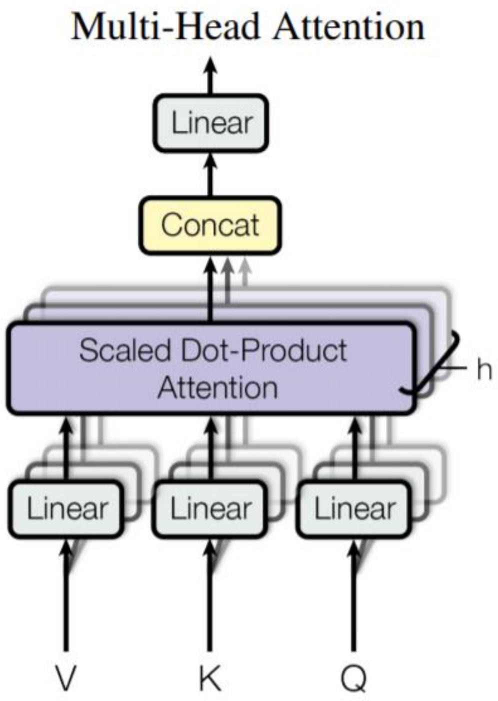


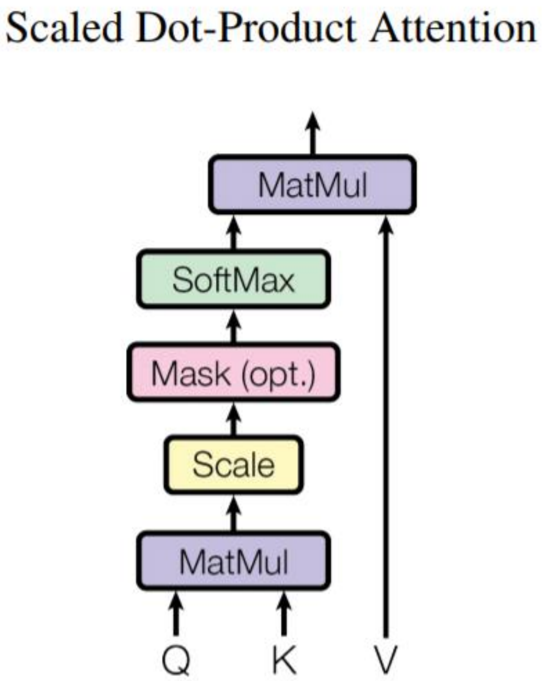


https://arxiv.org/pdf/1706.03762.pdf

Mechanism







https://arxiv.org/pdf/1706.03762.pdf

Config

Model Architecture

L : the number of layers

H : the hidden size

A : the number of self-attention heads

$\textbf{BERT}_{\textbf{BASE}}$

L = 12

H = 768

A = 12

Total Parameters = 110M

Chosen to have the same model size as
OpenAI GPT for comparison purposes

$\textbf{BERT}_{\textbf{LARGE}}$

L = 24

H = 1024

A = 16

Total Parameters = 340M

Pre-Training

Pre-training Data

For the pre-training corpus we use the BooksCorpus (800M words) and English Wikipedia (2,500M words)

Task1 : Masked Language Model (MLM)

- We mask 15% of all WordPiece tokens in each sequence at random
- the [MASK] token does not appear during fine-tuning

(80%) `my dog is hairy → my dog is [MASK]`

(10%) `my dog is hairy → my dog is apple`

(10%) `my dog is hairy → my dog is hairy`

Masking Rates			Dev Set Results		
MASK	SAME	RND	MNLI Fine-tune	NER Fine-tune	NER Feature-based
80%	10%	10%	84.2	95.4	94.9
100%	0%	0%	84.3	94.9	94.0
80%	0%	20%	84.1	95.2	94.6
80%	20%	0%	84.4	95.2	94.7
0%	20%	80%	83.7	94.8	94.6
0%	0%	100%	83.6	94.9	94.6

https://www.youtube.com/watch?v=IwtexRHoWG0&t=1181s

Pre-Training

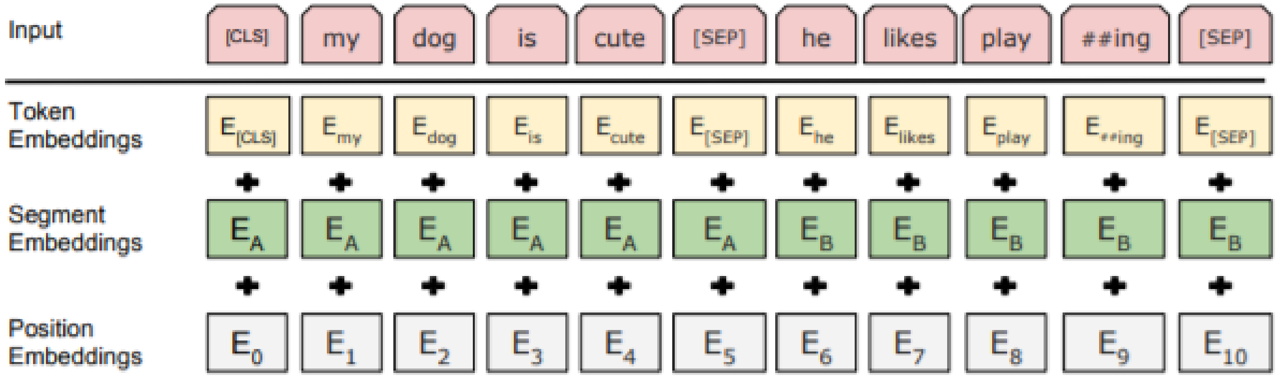


Figure 2: BERT input representation. The input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings.

Pre-Training

Task2 : Next Sentence Prediction (NSP)

- Many important downstream tasks such as Question Answering (QA) and Natural Language Inference (NLI) are based on understanding the relationship between two sentences

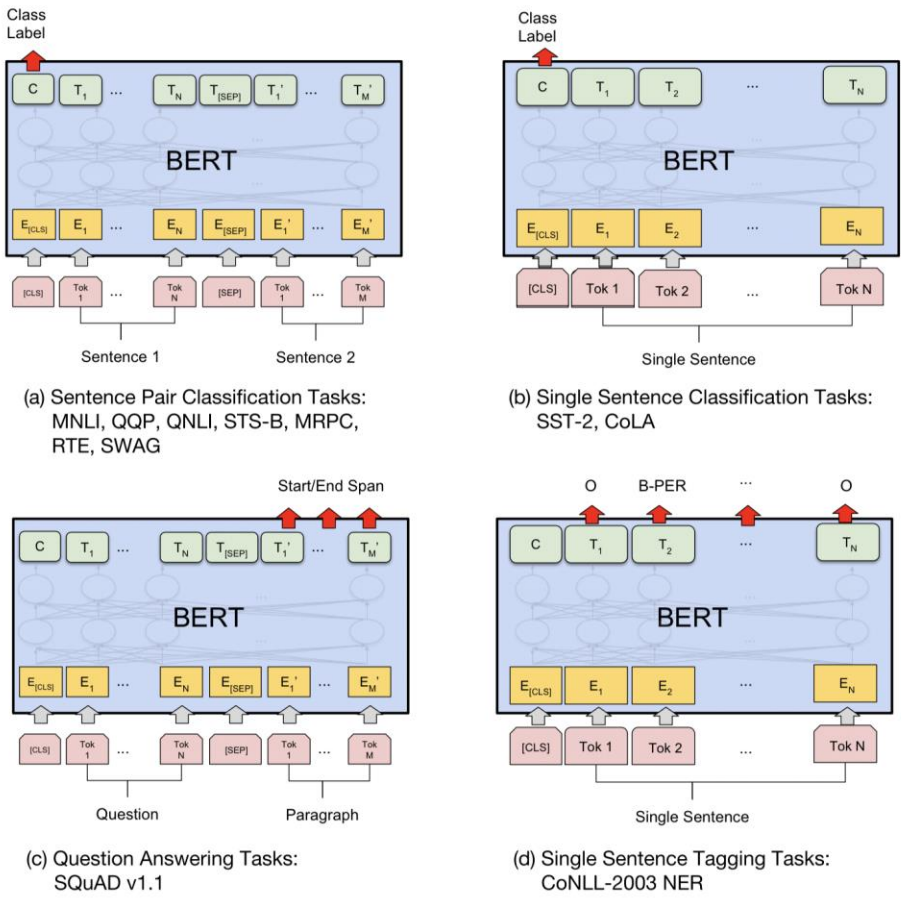
- 50% of the time B is the actual next sentence that follows A (labeled as IsNext), and 50% of the time it is a random sentence from the corpus (labeled as NotNext).

- this task is very beneficial to both QA and NLI.

Config

- **batch size** : 256 sequences
- **Adam optimizer**, learning rate : 1e-4, β1=0.9, β2=0.999,
- L2 weight decay of 0.01
- **Dropout prob**: 0.1 for all layers
- using **gelu** activation
- **BERT_base** - 4 TPUs, **BERT_large** - 16 TPUs For 4 days

Fine Tuning



(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

(b) Single Sentence Classification Tasks: SST-2, CoLA

(c) Question Answering Tasks: SQuAD v1.1

(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	**Average**
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

Experiment

Tasks	MNLI-m (Acc)	QNLI (Acc)	MRPC (Acc)	SST-2 (Acc)	SQuAD (F1)
	Dev Set				
$\text{BERT}_{\text{BASE}}$	84.4	88.4	86.7	92.7	88.5
No NSP	83.9	84.9	86.5	92.6	87.9
LTR & No NSP	82.1	84.3	77.5	92.1	77.8
+ BiLSTM	82.1	84.1	75.7	91.6	84.9

Table 5: Ablation over the pre-training tasks using the $\text{BERT}_{\text{BASE}}$ architecture. "No NSP" is trained without the next sentence prediction task. "LTR & No NSP" is trained as a left-to-right LM without the next sentence prediction, like OpenAI GPT. "+ BiLSTM" adds a randomly initialized BiLSTM on top of the "LTR + No NSP" model during fine-tuning.

NSP 없이 MLM만을 실험한 모델 LTR(Left-To-Right) 모델에 no NSP로 실험하는 경우.

LTR 인 경우 masking없이 모든 단어를 예측

Conclustion

- Recent empirical improvements due to transfer learning with language models have demonstrated that rich, unsupervised pre-training is an integral part of many language understanding systems.

- Our major contribution is further generalizing these findings to deep bidirectional architectures, allowing the same pre-trained model to successfully tackle a broad set of NLP tasks.

Thank You

감사합니다.