딥러닝 분반

임효진

토픽 모델링

- Topic Modeling: A type of statistical model for discovering the abstract "topics" that occur in a collection of documents
    - LSA
    - LDA

LSA (Latent Semantic Analysis)

	pizza	pizza hamburger cookie	hamburger	ramen	sushi	ramen sushi
	d1	d2	d3	d4	d5	d6
pizza	1	1	0	0	0	0
ham burger	0	1	1	0	0	0
cookie	0	1	0	0	0	0
ramen	0	0	0	1	0	1
sushi	0	0	0	0	1	1

= A

$$A \doteq U \times \Sigma \times V^T$$

U

	t1	t2	t3	t4	t5
w1	0.6	0	0	0.7	-0.3
w2	0.6	0	0	-0.7	-0.3
w3	0.5	0	0	0	0.9
w4	0	0.7	-0.7	0	0
w5	0	0.7	0.7	0	0

Word matrix for topic

Σ

	t1	t2	t3	t4	t5	t6
t1	1.9	0	0	0	0	0
t2	0	1.7	0	0	0	0
t3	0	0	1	0	0	0
t4	0	0	0	1	0	0
t5	0	0	0	0	0.5	0

Topic Strength

V^T

	d1	d2	d3	d4	d5	d6
t1	0.3	0.9	0.3	0	0	0
t2	0	0	0	0.4	0.4	0.8
t3	0	0	0	-0.7	0.7	0
t4	0.7	0	-0.7	0	0	0
t5	-0.6	0.5	-0.6	0	0	0
t6	0	0	0	-0.6	-0.6	0.6

Document matrix for topic

	d1	d2	d3	d4	d5	d6
t1	0.57	1.71	0.57	0	0	0
t2	0	0	0	0.68	0.68	1.36

=

Σ

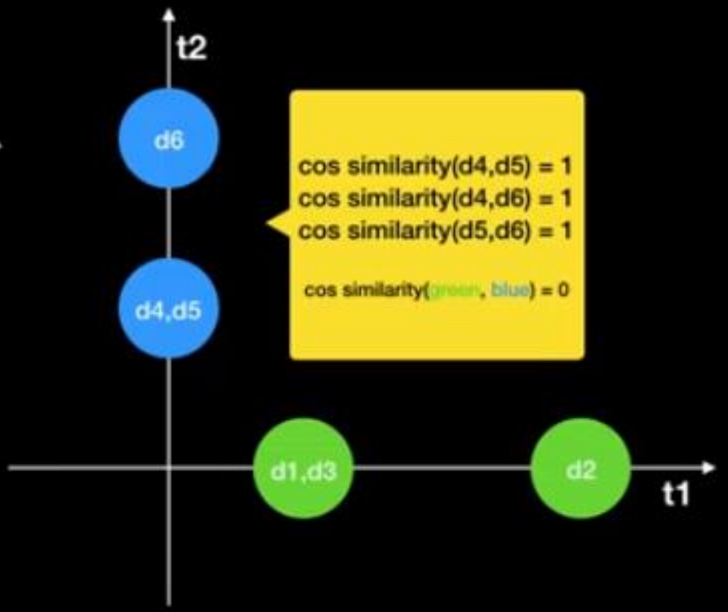
	t1	t2
t1	1.9	0
t2	0	1.7

X

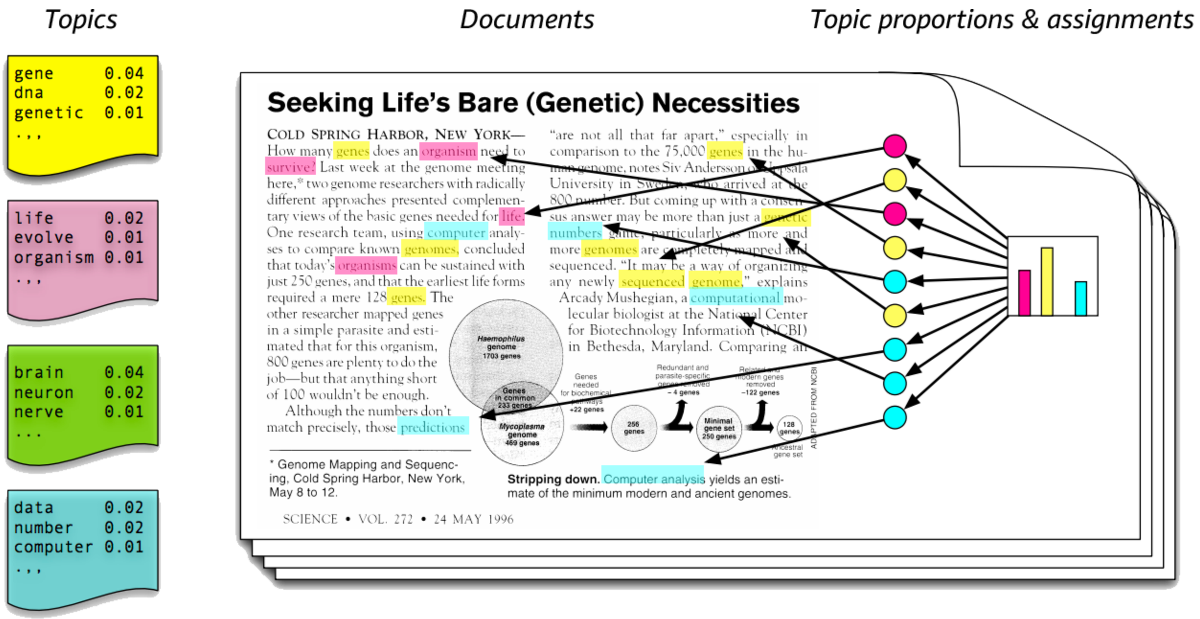
V^T

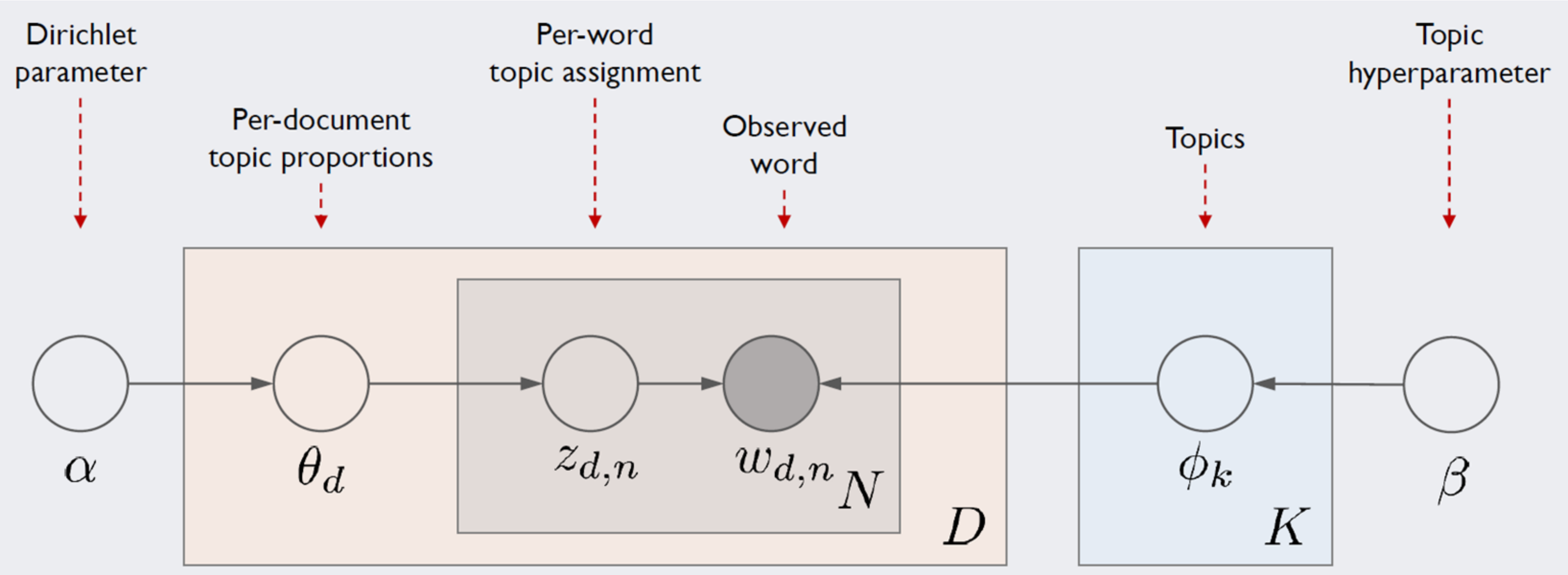
	d1	d2	d3	d4	d5	d6
t1	0.3	0.9	0.3	0	0	0
t2	0	0	0	0.4	0.4	0.8

	pizza	pizza hamburger cookie	hamburger	ramen	sushi	ramen sushi
	d1	d2	d3	d4	d5	d6
t1	0.57	1.71	0.57	0	0	0
t2	0	0	0	0.68	0.68	1.36



LDA (Latent Dirichlet Allocation)



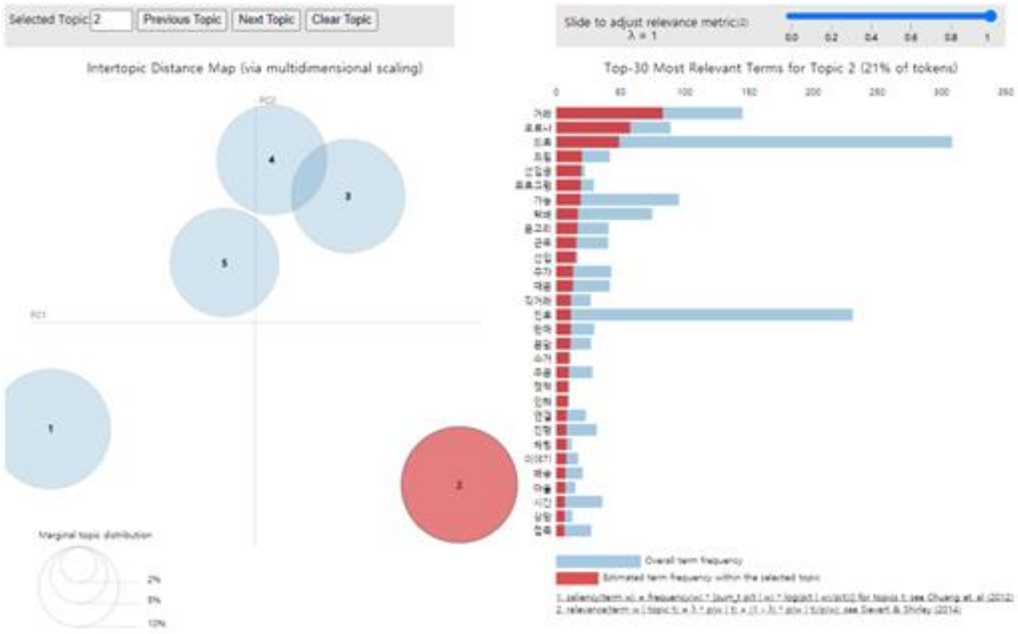

Dirichlet parameter
Per-document topic proportions
Per-word topic assignment
Observed word
Topics
Topic hyperparameter
α
θ_d
$z_{d,n}$
$w_{d,n}$
N
D
ϕ_k
K
β

	1번문서		2번문서				3번문서		
단어	문고리	거래	가방	나눔	문고리	드림	비대면	거래	택배
주제	topic1	topic2	topic1	topic1	topic2	topic2	topic3	topic2	topic3

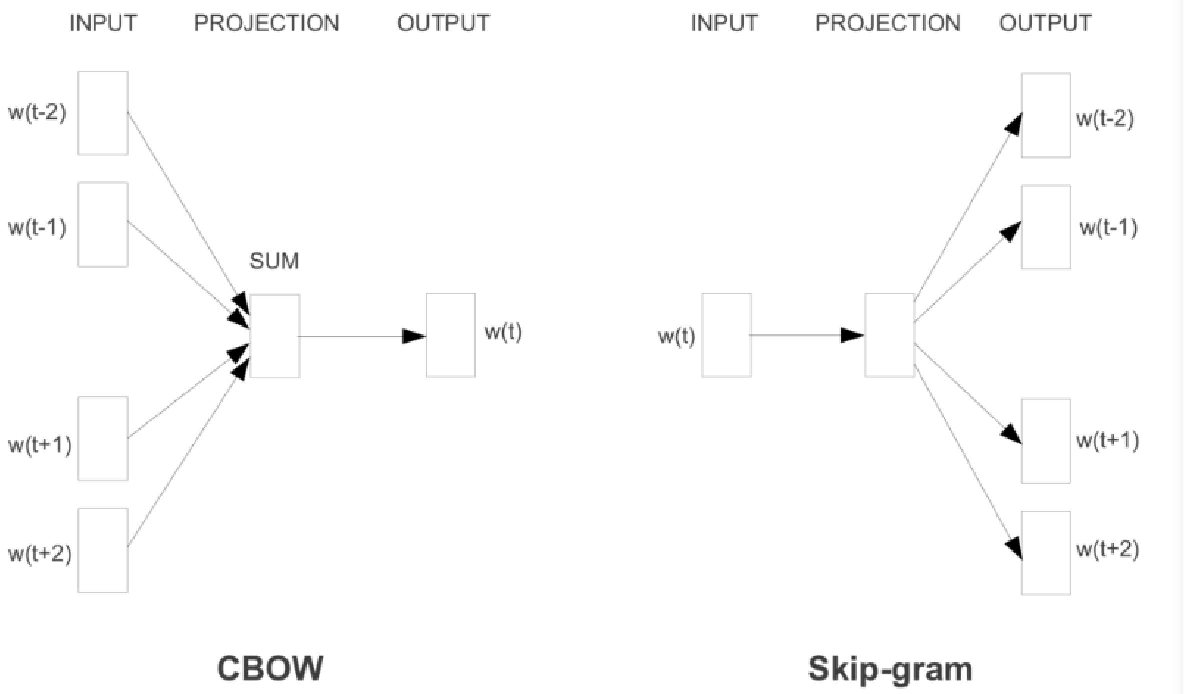
토픽-문서	1번문서	2번문서	3번문서
topic1	1.01	2.01	0.01
topic2	1.01	2.01	1.01
topic3	0.01	0.01	2.01

토픽-단어	문고리	거래	가방	나눔	드림	비대면	택배
topic1	1.001	0.001	1.001	1.001	0.001	0.001	0.001
topic2	1.001	2.001	0.001	0.001	1.001	0.001	0.001
topic3	0.001	0.001	0.001	0.001	0.001	1.001	1.001

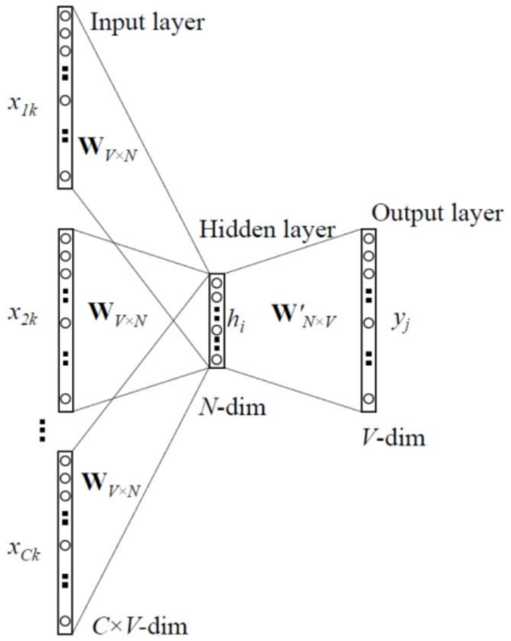
	1번문서		2번문서				3번문서		
단어	문고리	거래	가방	나눔	문고리	드림	비대면	거래	택배
주제	미분류	topic2	topic1	topic1	topic2	topic2	topic3	topic2	topic3


Selected Topic 2
Previous Topic
Next Topic
Clear Topic
Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
4
3
5
1
2
Marginal topic distribution
2%
5%
10%
Slide to adjust relevance metric
λ = 1
0.0
0.2
0.4
0.6
0.8
1
Top-30 Most Relevant Terms for Topic 2 (21% of tokens)
0
50
100
150
200
250
300
350
Overall term frequency
Estimated term frequency within the selected topic

Word2vec


INPUT
PROJECTION
OUTPUT
w(t-2)
w(t-1)
SUM
w(t)
w(t+1)
w(t+2)
CBOW
INPUT
PROJECTION
OUTPUT
w(t)
w(t-2)
w(t-1)
w(t+1)
w(t+2)
Skip-gram

CBOW (Continuous Bag of Words)



$$x_k = [0, \ldots, 0, 1, 0, \ldots, 0]$$

$$(x^{c-m}, x^{c-m+1}, \ldots, x^{c-1}, x^{c+1}, \ldots x^{c+m-1}, x^{c+m}) \in \mathbb{R}^{|V|}$$

$$\mathbf{W} \in \mathbb{R}^{V\times N},\ \mathbf{W}' \in \mathbb{R}^{N\times V}$$

$$P(x_c | x_{c-m}, \ldots x_{c-1}, x_{c+1}, \ldots, x_{c+m})$$

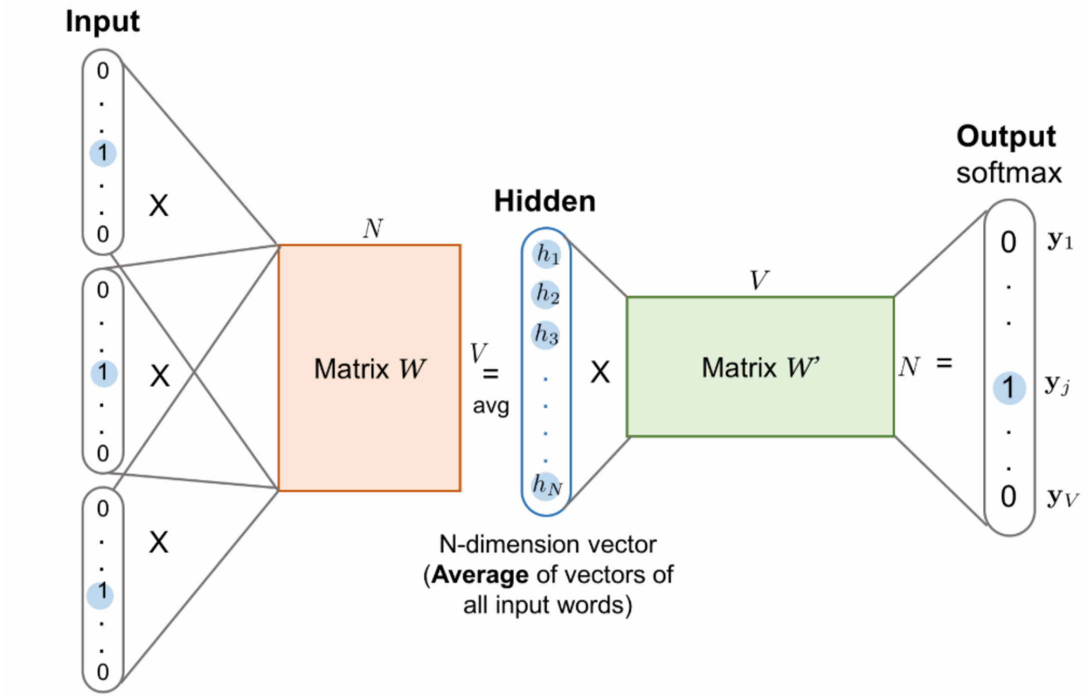
$$\begin{bmatrix}0 & 0 & 0 & \boxed{1} & 0\end{bmatrix} \times \begin{bmatrix}17 & 24 & 1 \\ 23 & 5 & 7 \\ 4 & 6 & 13 \\ \boxed{10 & 12 & 19} \\ 11 & 18 & 25\end{bmatrix} = \begin{bmatrix}10 & 12 & 19\end{bmatrix}$$

$$(v_{c-m} = \mathbf{W}x^{c-m}, \ldots, v_{c+m} = \mathbf{W}x^{c+m}) \in \mathbb{R}^n$$

$$\hat{v} = \frac{v_{c-m} + v_{c-m+1} + \cdots + v_{c+m}}{2m} \in \mathbb{R}^n$$

$$z = \mathbf{U}\hat{v} \in \mathbb{R}^{|V|}$$

$$\hat{y} = softmax(z) \in \mathbb{R}^{|V|}$$



$$H(\hat{y}, y) = -\sum_{j=1}^{|V|} y_j \log(\hat{y}_j)$$

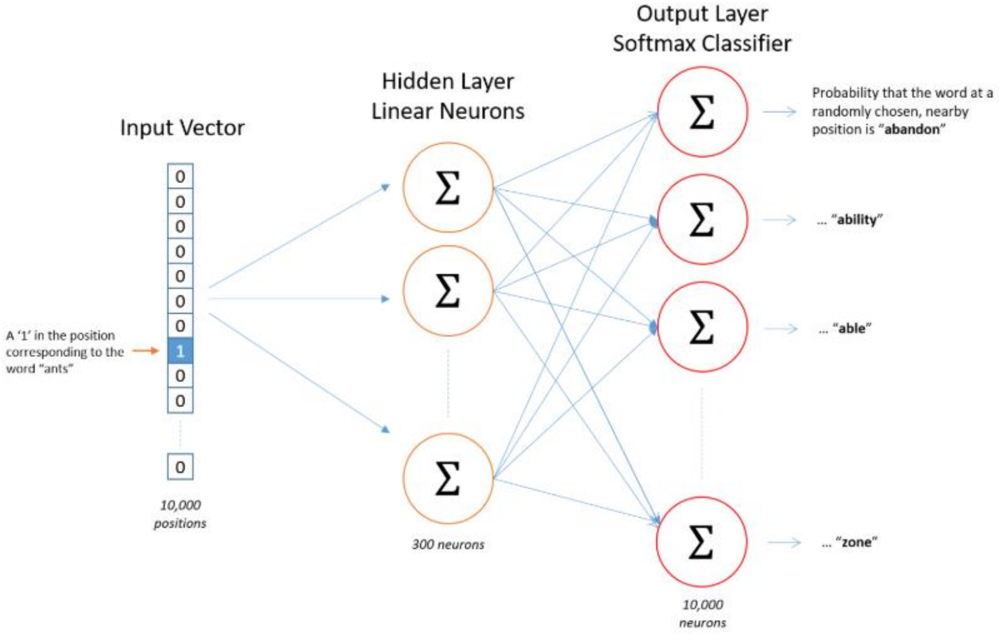
$$H(\hat{y}, y) = -y_i \log(\hat{y}_i)$$

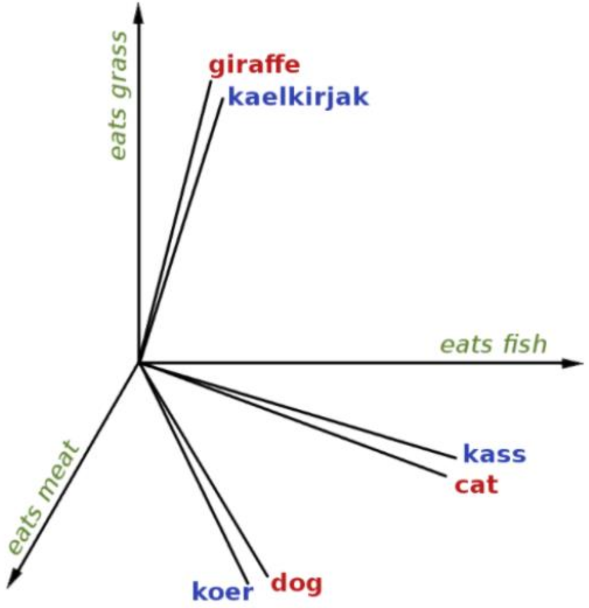
$$\begin{aligned} minimizeJ &= -\log P(w_c | w_{c-m}, \ldots, w_{c+m}) \\ &= -\log P(u_c | v^{)} \\ &= -\log \frac{exp(u_c^{\intercal} \hat{v})}{\sum_{j=1}^{|V|} exp(u_j^{\intercal} \hat{v})} \\ &= -u_c^{intercal} \hat{v} + \log \sum_{j=1}^{|V|} exp(u_j^{\intercal} \hat{v}) \end{aligned}$$

- C개의 단어를 Hidden Layer로 보내는 $C \times N$
- Hidden Layer에서 Output Layer로 가는 $N \times V$

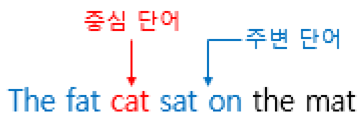
Skip-gram

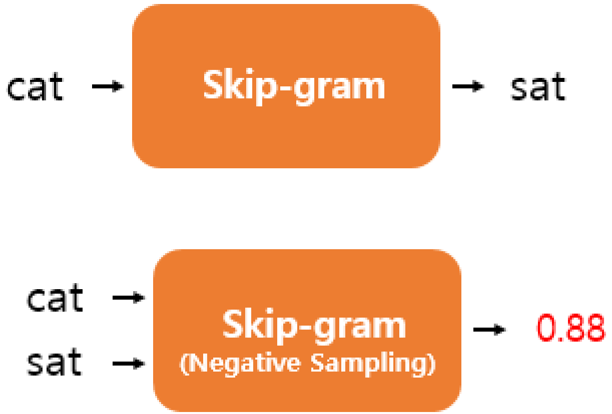
Model Architecture	Semantic-Syntactic Word Relationship test set		MSR Word Relatedness Test Set [20]
	Semantic Accuracy [%]	Syntactic Accuracy [%]	
RNNLM	9	36	35
NNLM	23	53	47
CBOW	24	64	61
Skip-gram	55	59	56





Negative Sampling





중심 단어

주변 단어

The fat cat sat on the mat

The fat cat sat on the mat

입력	레이블
cat	The
cat	fat
cat	sat
cat	on
sat	fat
sat	cat
sat	on
sat	the
...	...

입력과 레이블의 변화

입력1	입력2	레이블
cat	The	1
cat	fat	1
cat	sat	1
cat	on	1
sat	fat	1
sat	cat	1
sat	on	1
sat	the	1
...	...	...

입력과 레이블의 변화

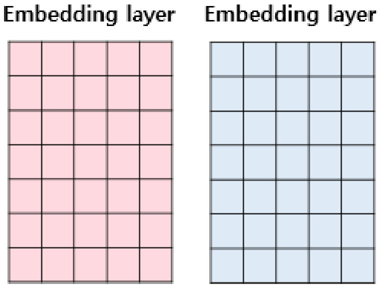
입력1	입력2	레이블
cat	The	1
cat	fat	1
cat	sat	1
cat	on	1

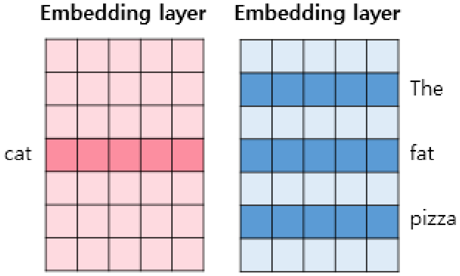
Negative Sampling

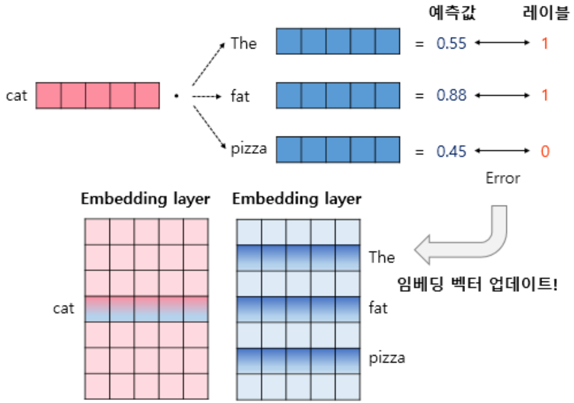
입력1	입력2	레이블
cat	The	1
cat	fat	1
cat	**pizza**	**0**
cat	**computer**	**0**
cat	sat	1
cat	on	1

단어 집합에서 랜덤으로
선택된 단어들을
레이블 0의 샘플로 추가.

입력1	입력2	레이블
cat	The	1
cat	fat	1
cat	pizza	0
cat	computer	0
cat	sat	1
cat	on	1
cat	cute	1
cat	mighty	0
...	...	...







Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "**Sequence to sequence learning with neural networks**." Advances in neural information processing systems. 2014.

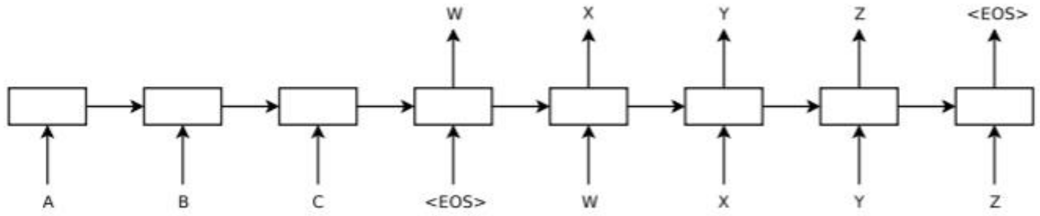
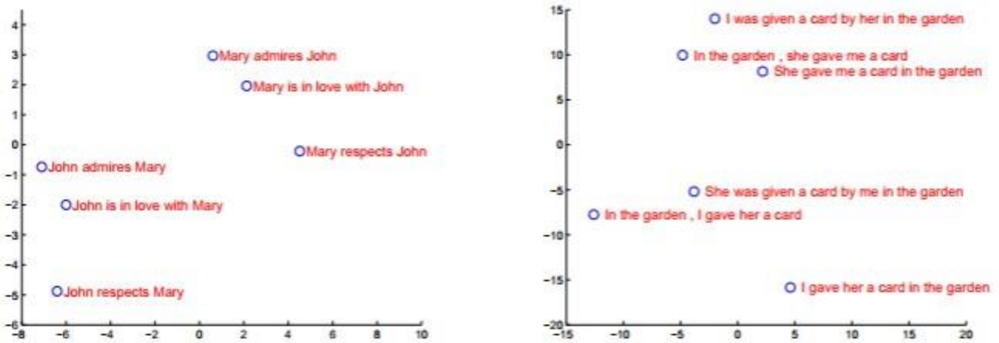


Figure 1: Our model reads an input sentence "ABC" and produces "WXYZ" as the output sentence. The model stops making predictions after outputting the end-of-sentence token. Note that the LSTM reads the input sentence in reverse, because doing so introduces many short term dependencies in the data that make the optimization problem much easier.



Vaswani, Ashish, et al. "**Attention is all you need.**" Advances in neural information processing systems. 2017.

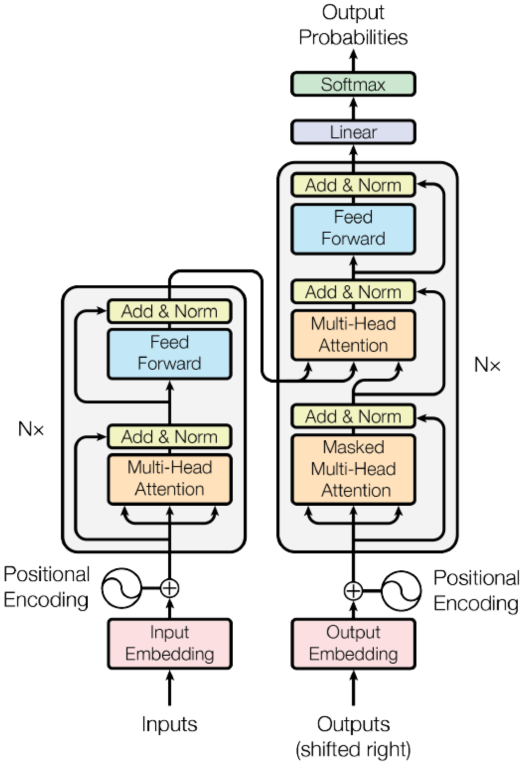


Figure 1: The Transformer - model architecture.

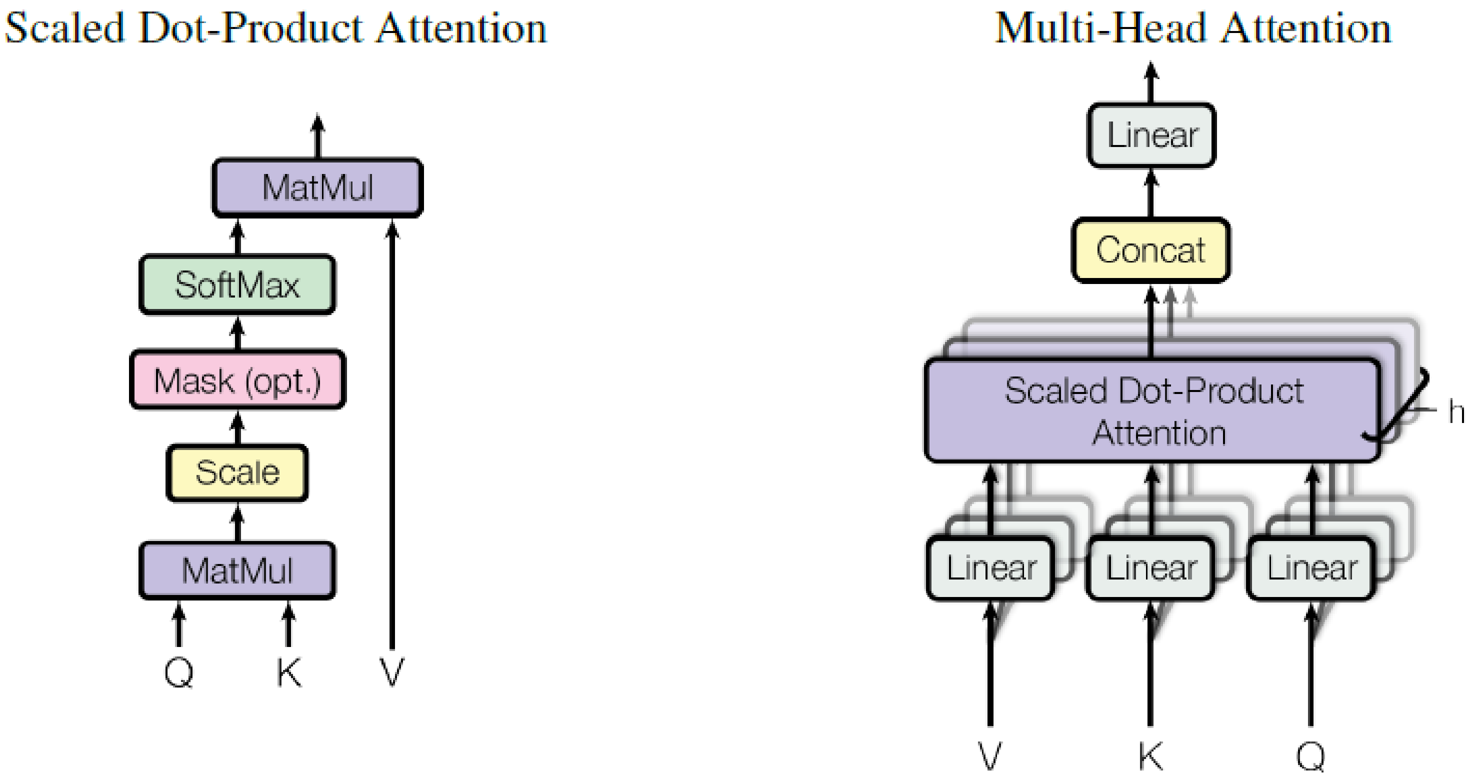
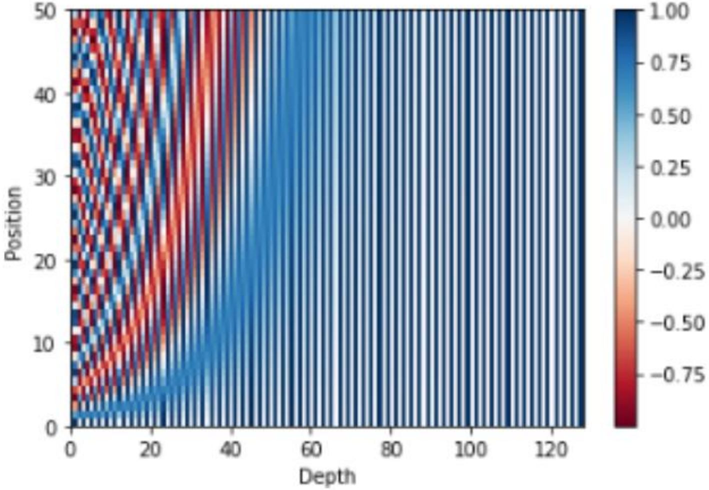


Figure 2: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel.

$$\mathrm{Attention}(Q,K,V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

$$\mathrm{MultiHead}(Q,K,V) = \mathrm{Concat}(\mathrm{head_1},...,\mathrm{head_h})W^O$$
$$\text{where } \mathrm{head_i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

Where the projections are parameter matrices $W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$ and $W^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$.



Attention Visualizations

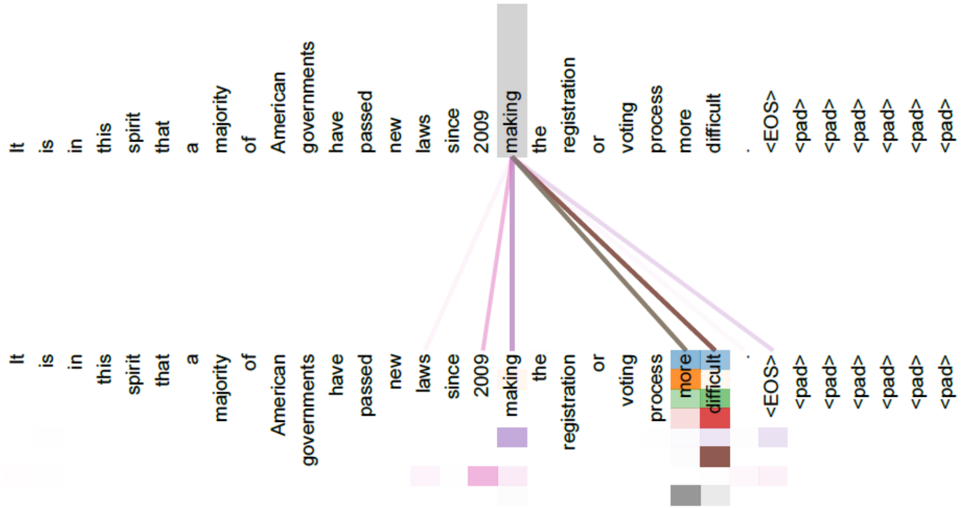
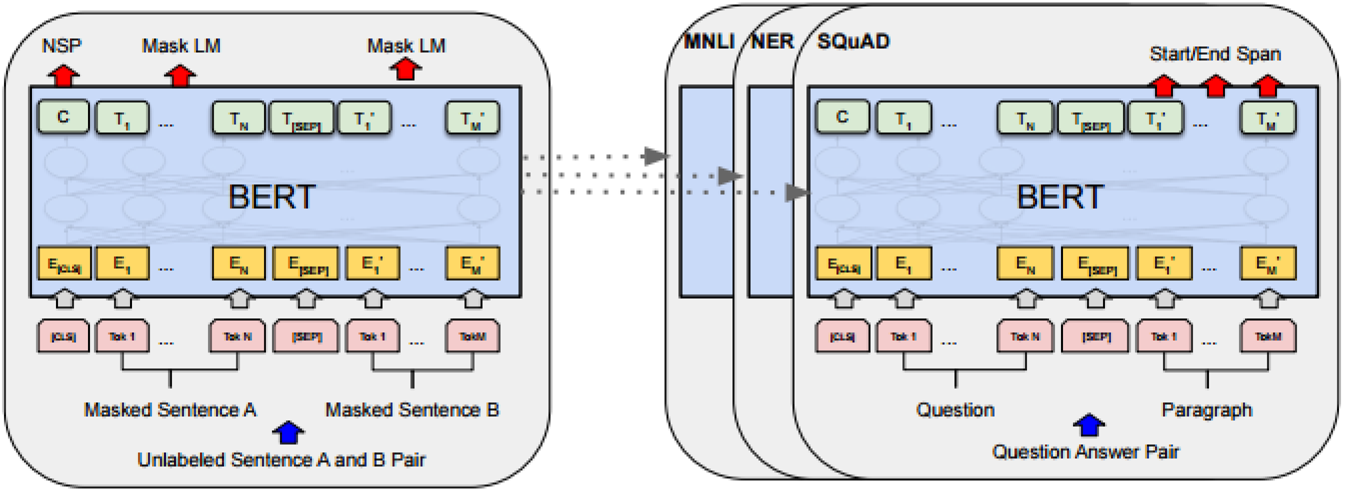


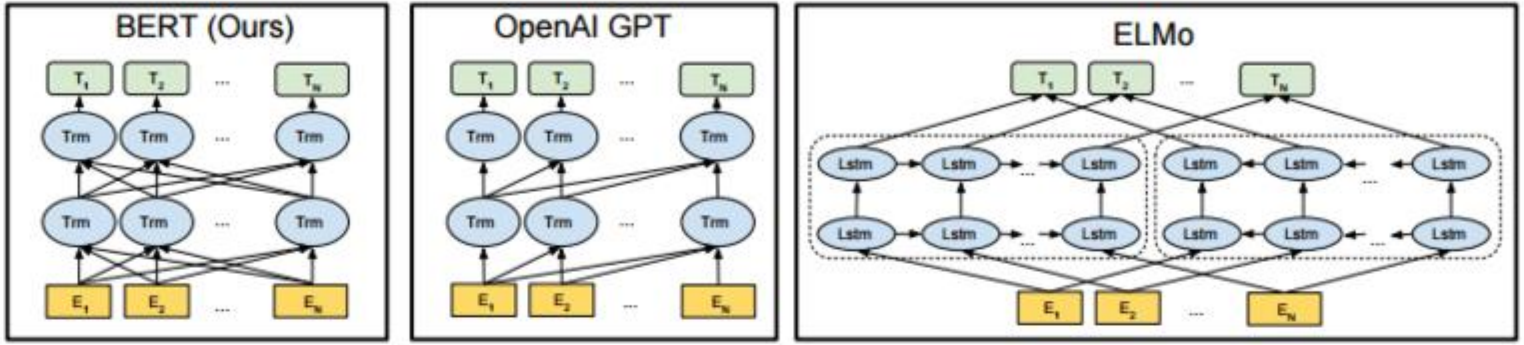
Figure 3: An example of the attention mechanism following long-distance dependencies in the encoder self-attention in layer 5 of 6. Many of the attention heads attend to a distant dependency of the verb 'making', completing the phrase 'making...more difficult'. Attentions here shown only for the word 'making'. Different colors represent different heads. Best viewed in color.

Devlin, Jacob, et al. "**Bert: Pre-training of deep bidirectional transformers for language understanding**." arXiv preprint arXiv:1810.04805 (2018).

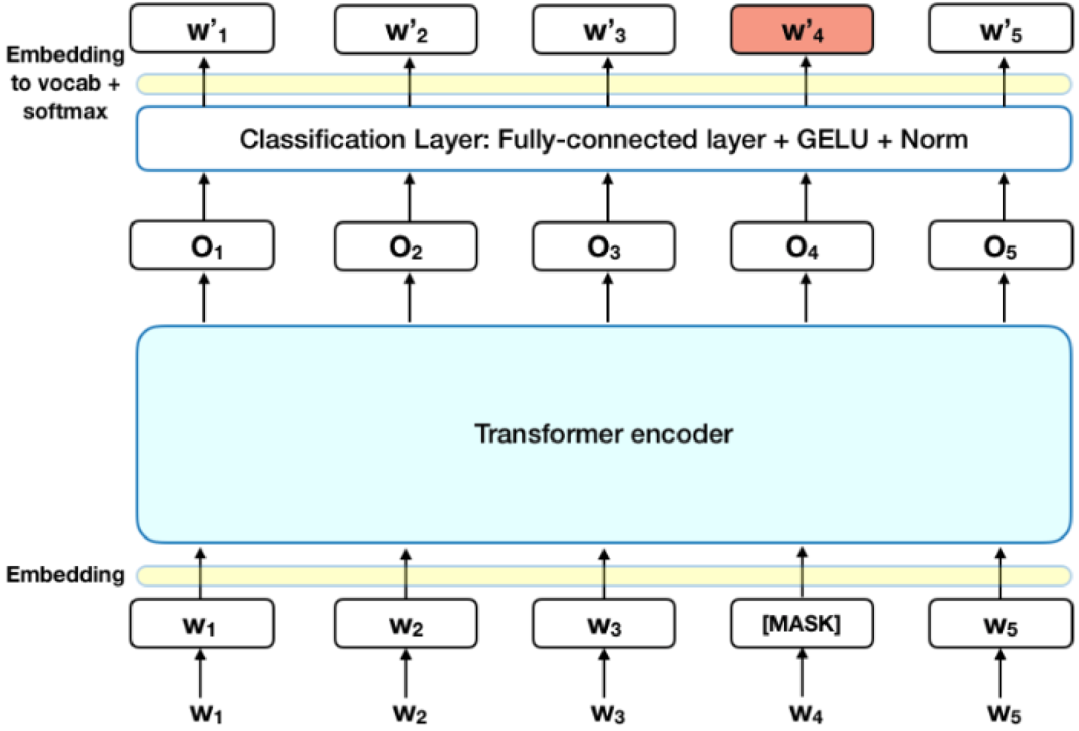


Pre-training

Fine-Tuning



BERT (Ours)
OpenAI GPT
ELMo
Trm
Lstm



- 80% of the time: Replace the word with the `[MASK]` token, e.g., `my dog is hairy` → `my dog is [MASK]`
- 10% of the time: Replace the word with a random word, e.g., `my dog is hairy` → `my dog is apple`
- 10% of the time: Keep the word unchanged, e.g., `my dog is hairy` → `my dog is hairy`. The purpose of this is to bias the representation towards the actual observed word.

Input = `[CLS] the man went to [MASK] store [SEP]`

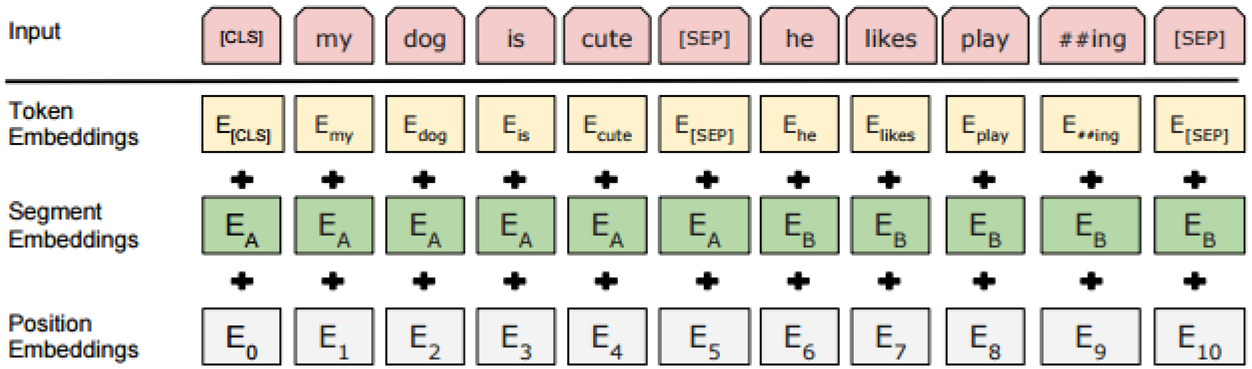
`he bought a gallon [MASK] milk [SEP]`

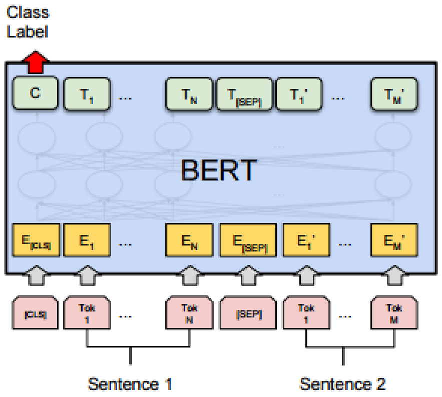
Label = `IsNext`

Input = `[CLS] the man [MASK] to the store [SEP]`

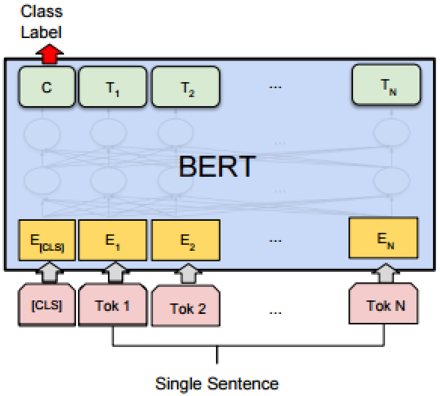
`penguin [MASK] are flight ##less birds [SEP]`

Label = `NotNext`

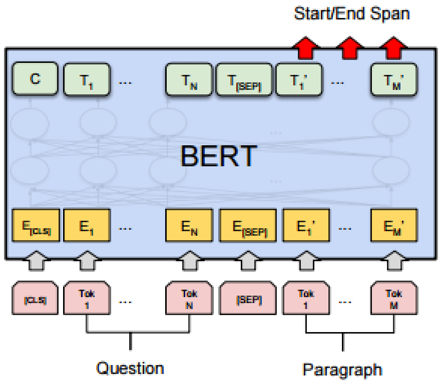

Input
[CLS]
my
dog
is
cute
[SEP]
he
likes
play
##ing
[SEP]
Token Embeddings
E[CLS]
Emy
Edog
Eis
Ecute
E[SEP]
Ehe
Elikes
Eplay
E##ing
E[SEP]
Segment Embeddings
EA
EA
EA
EA
EA
EA
EB
EB
EB
EB
EB
Position Embeddings
E0
E1
E2
E3
E4
E5
E6
E7
E8
E9
E10



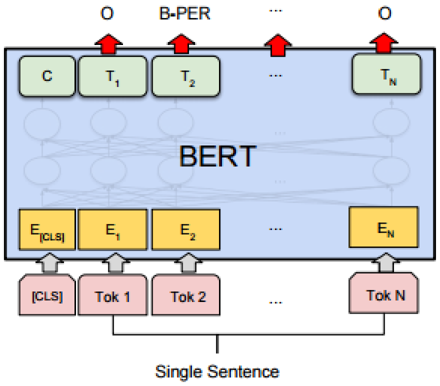
(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG



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



(c) Question Answering Tasks: SQuAD v1.1



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

Figure 4: Illustrations of Fine-tuning BERT on Different Tasks.