

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

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

	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

	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

	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

Word matrix
for topic

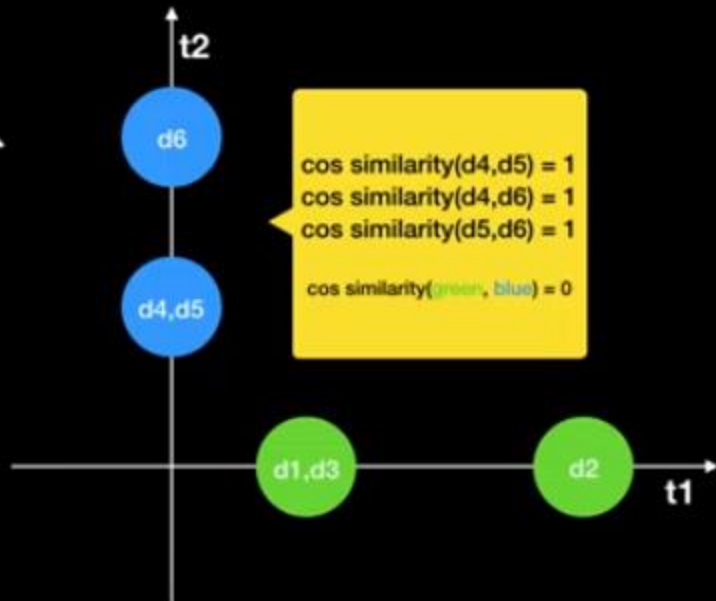
Topic Strength

Document matrix
for topic

$$\begin{array}{|c|c|c|c|c|c|c|} \hline & d1 & d2 & d3 & d4 & d5 & d6 \\ \hline t1 & 0.57 & 1.71 & 0.57 & 0 & 0 & 0 \\ \hline t2 & 0 & 0 & 0 & 0.68 & 0.68 & 1.36 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline & t1 & t2 \\ \hline t1 & 1.9 & 0 \\ \hline t2 & 0 & 1.7 \\ \hline \end{array} \times \begin{array}{|c|c|c|c|c|c|c|} \hline & d1 & d2 & d3 & d4 & d5 & d6 \\ \hline t1 & 0.3 & 0.9 & 0.3 & 0 & 0 & 0 \\ \hline t2 & 0 & 0 & 0 & 0.4 & 0.4 & 0.8 \\ \hline \end{array}$$

The image illustrates a matrix multiplication operation. On the left is a 2x7 matrix with columns labeled d1 through d6 and rows labeled t1 and t2. This is followed by an equals sign, then a 2x2 matrix labeled Σ with rows and columns labeled t1 and t2. This is followed by a multiplication symbol \times , then a 2x7 matrix labeled V^T with columns labeled d1 through d6 and rows labeled t1 and t2. The matrices are displayed with a color scheme: the first matrix has a blue header and first column, and black data cells; the Σ matrix has a blue header and first column, and green data cells; the V^T matrix has a blue header and first column, and green data cells.

		pizza	pizza	hamburger	hamburger cookie		
		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)

Topics

gene	0.04
dna	0.02
genetic	0.01
...	

life	0.02
evolve	0.01
organism	0.01
...	

brain	0.04
neuron	0.02
nerve	0.01
...	

data	0.02
number	0.02
computer	0.01
...	

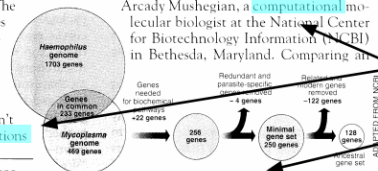
Documents

Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK—How many **genes** does an **organism** need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for **life**. One research team, using **computer** analyses to compare known **genomes**, concluded that today's **organisms** can be sustained with just 250 genes, and that the earliest life forms required a mere 128 **genes**. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

Although the numbers don't match precisely, those **predictions**

"are not all that far apart," especially in comparison to the 75,000 **genes** in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a **guess**. **Numbers** alone, particularly as more and more **genomes** are completely mapped and sequenced, "It may be a way of organizing any newly **sequenced genome**," explains Arcady Mushegian, a **computational** molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an

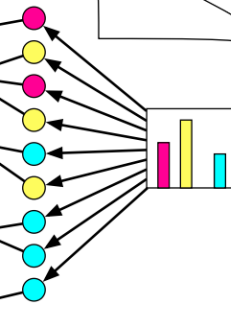


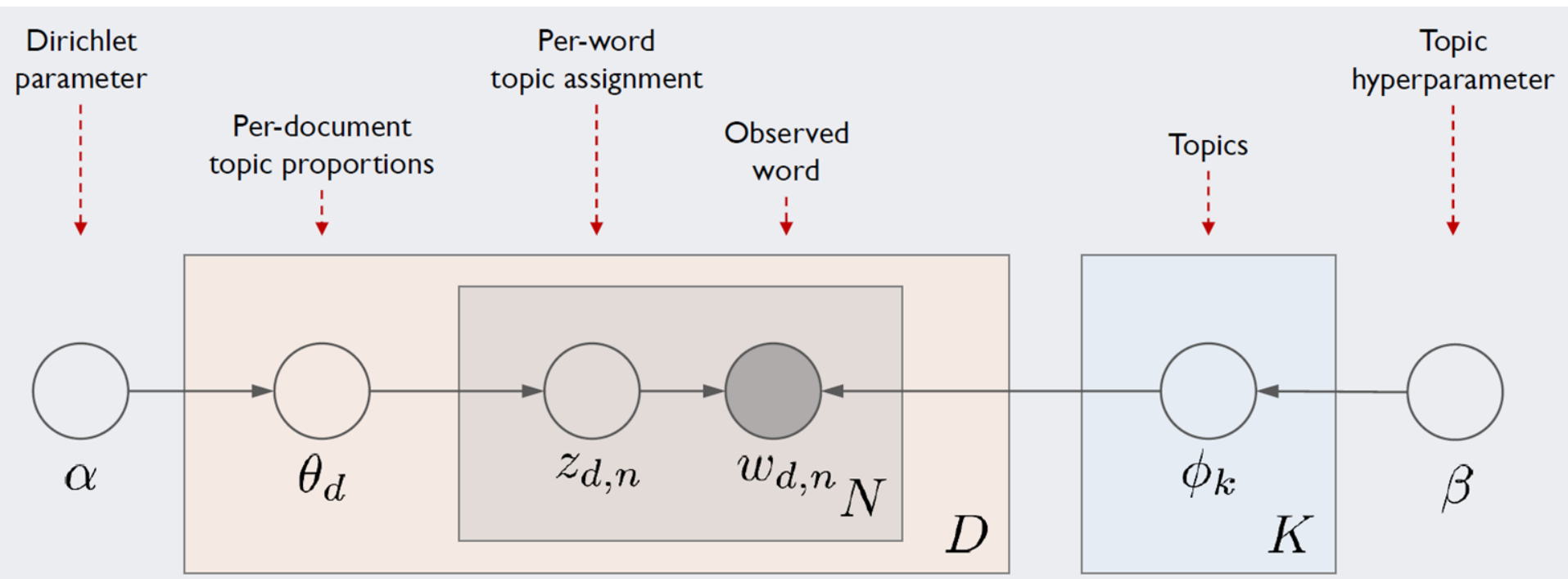
* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

Stripping down. **Computer analysis** yields an estimate of the minimum modern and ancient genomes.

SCIENCE • VOL. 272 • 24 MAY 1996

Topic proportions & assignments





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

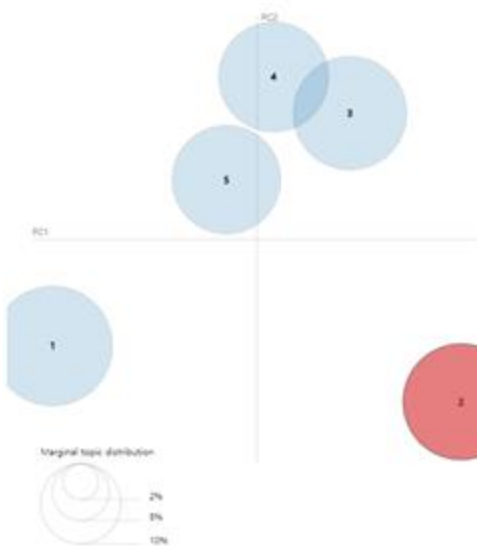
토픽-문서	1번문서	2번문서	3번문서
topic1	1.01	2.01	0.01
topic2	1.01	2.01	1.01
topic3	0.01	0.01	2.01

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

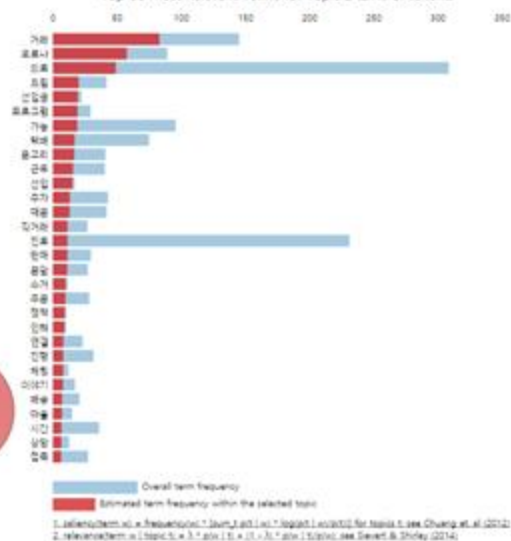
Selected Topic: 2 Previous Topic Next Topic Clear Topic

Intertopic Distance Map (via multidimensional scaling)

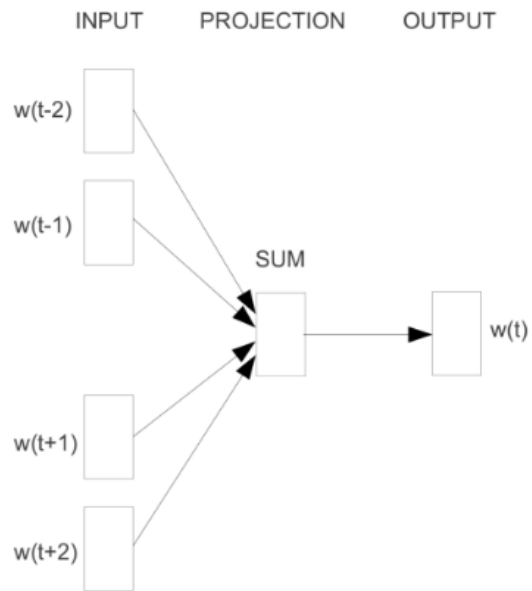


Slide to adjust relevance metric: $\lambda = 1$

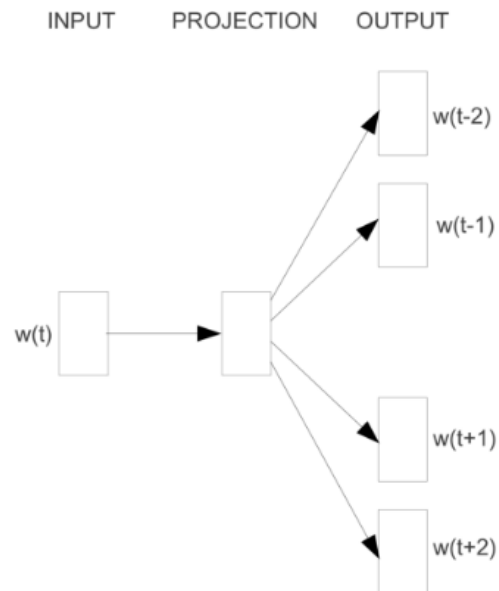
Top-30 Most Relevant Terms for Topic 2 (21% of tokens)



Word2vec

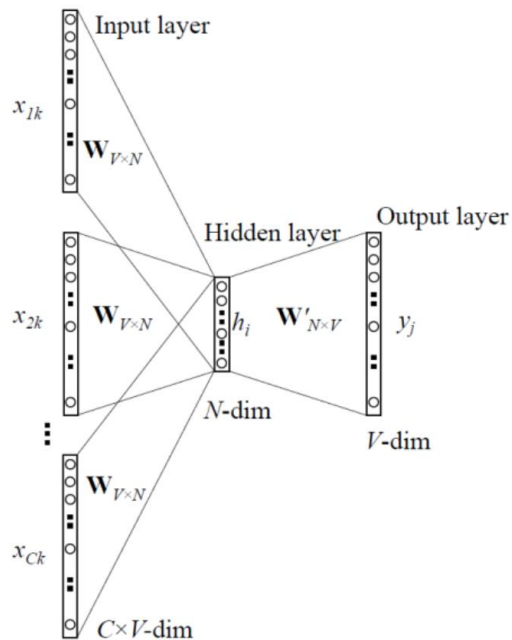


CBOW



Skip-gram

CBOW (Continuous Bag of Words)



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

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

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

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

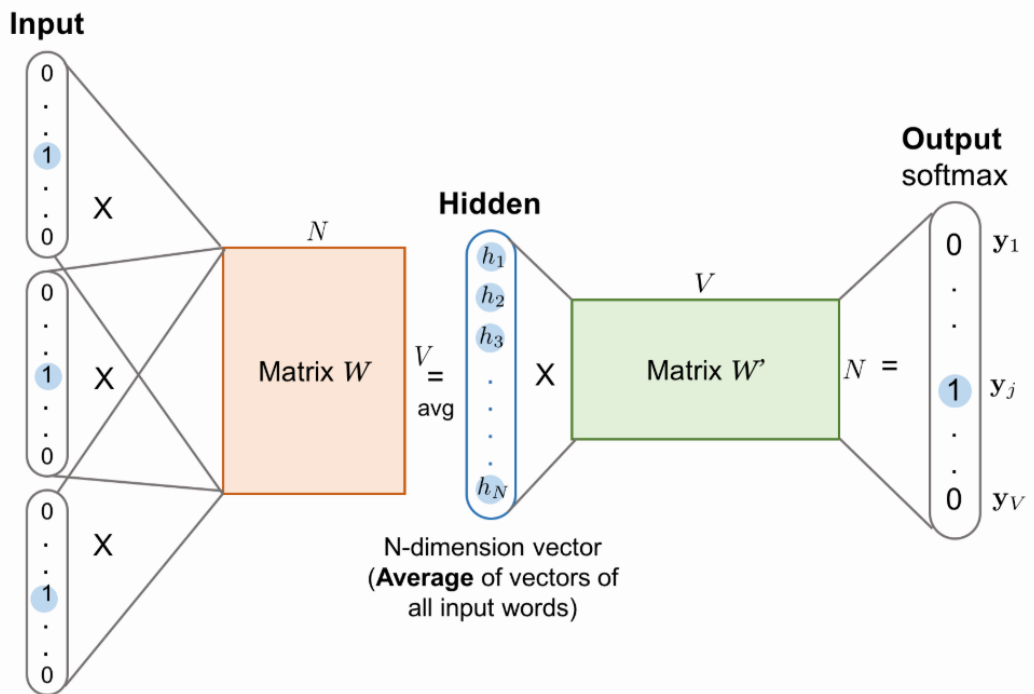
$$[0 \quad 0 \quad 0 \quad \mathbf{1} \quad 0] \times \begin{bmatrix} 17 & 24 & 1 \\ 23 & 5 & 7 \\ 4 & 6 & 13 \\ \mathbf{10} & \mathbf{12} & \mathbf{19} \\ 11 & 18 & 25 \end{bmatrix} = [10 \quad 12 \quad 19]$$

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

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

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

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



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

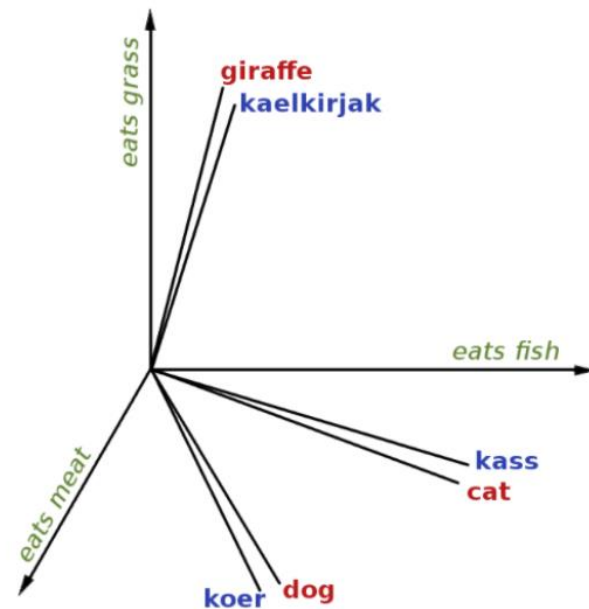
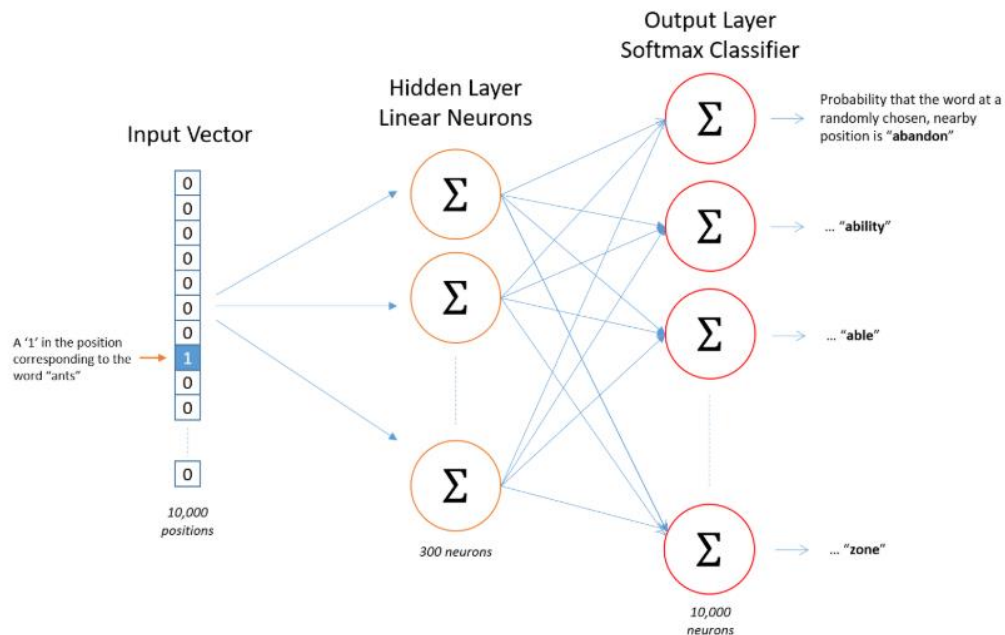
$$H(\hat{y}, y) = -y_i \log(\hat{y}_i)$$

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

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

Skip-gram

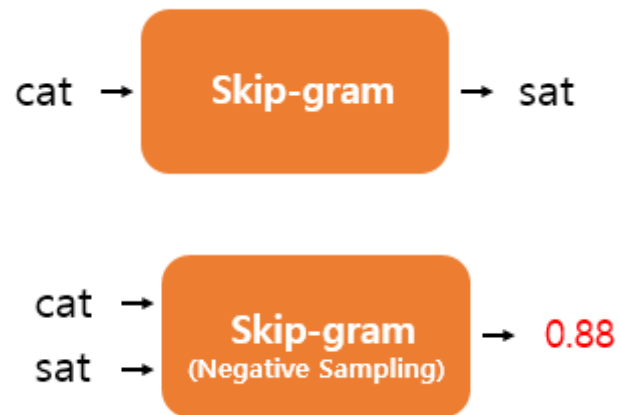
Model Architecture	Semantic-Syntactic Word Relationship test set		MSR Word Relatedness Test Set [20]
	Semantic Accuracy [%]	Syntactic Accuracy [%]	
RNNLM	9	36	35
NNLM	23	53	47
CBOV	24	64	61
Skip-gram	55	59	56

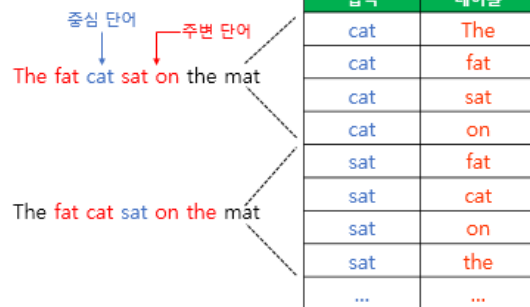


Negative Sampling

중심 단어
주변 단어

The fat **cat** sat on the mat





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



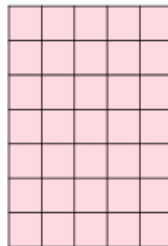
Negative Sampling

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

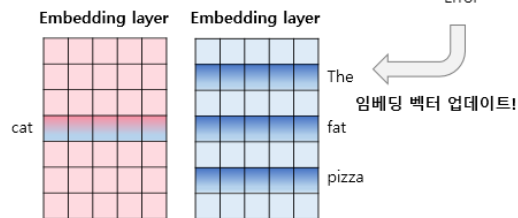
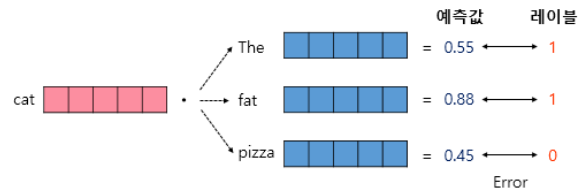
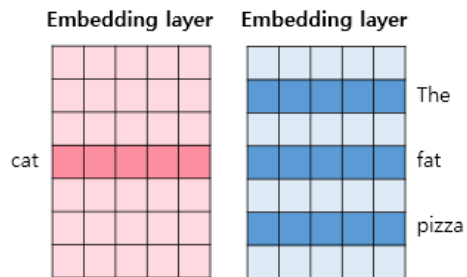
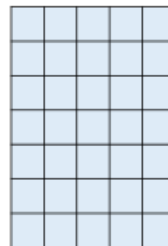
단어 집합에서 랜덤으로
선택된 단어들을
레이블 0의 샘플로 추가.

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
...

Embedding layer



Embedding layer



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

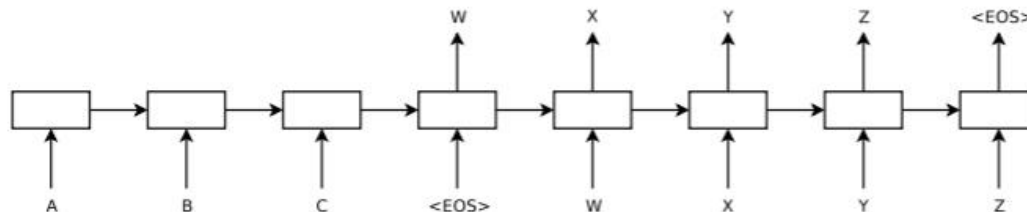
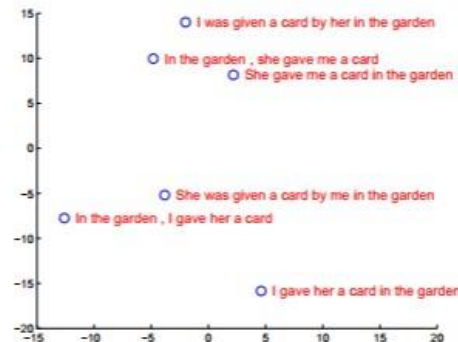
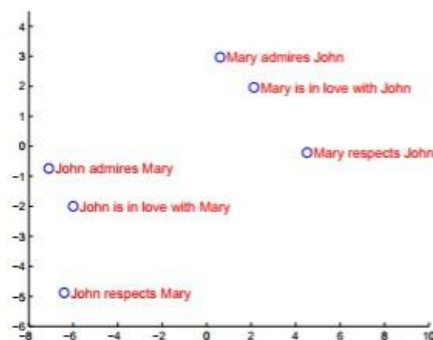


Figure 1: Our model reads an input sentence “ABC” and produces “WXYZ” as the output sentence. The model stops making predictions after outputting the end-of-sentence token. Note that the LSTM reads the input sentence in reverse, because doing so introduces many short term dependencies in the data that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



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

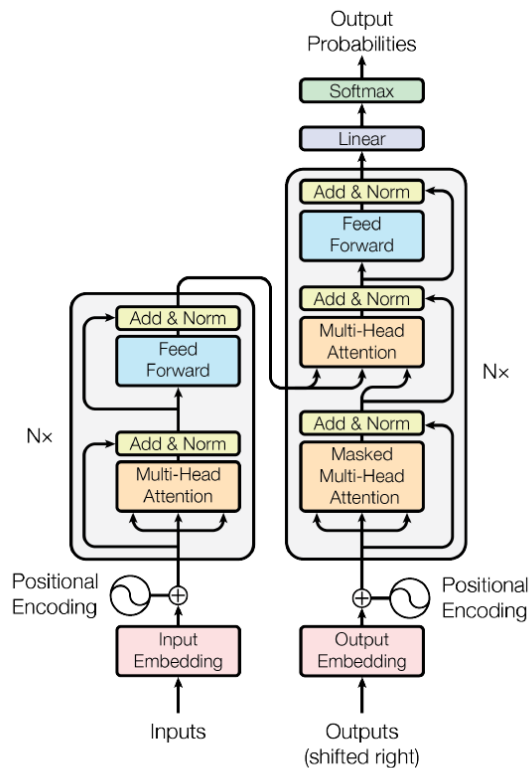
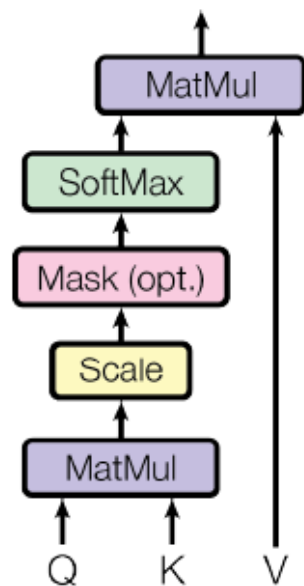


Figure 1: The Transformer - model architecture.

Scaled Dot-Product Attention



Multi-Head Attention

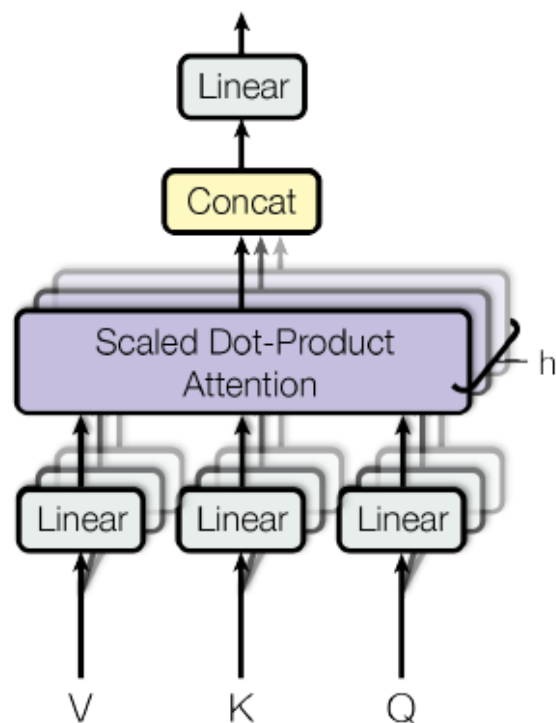


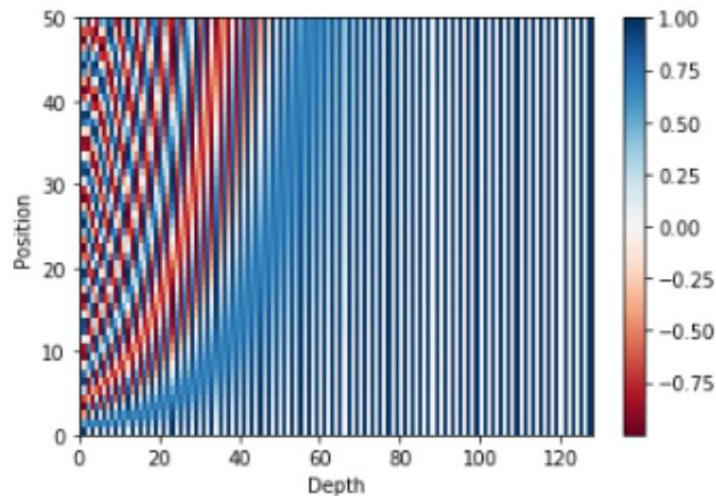
Figure 2: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel.

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

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O$$

$$\text{where } \text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

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

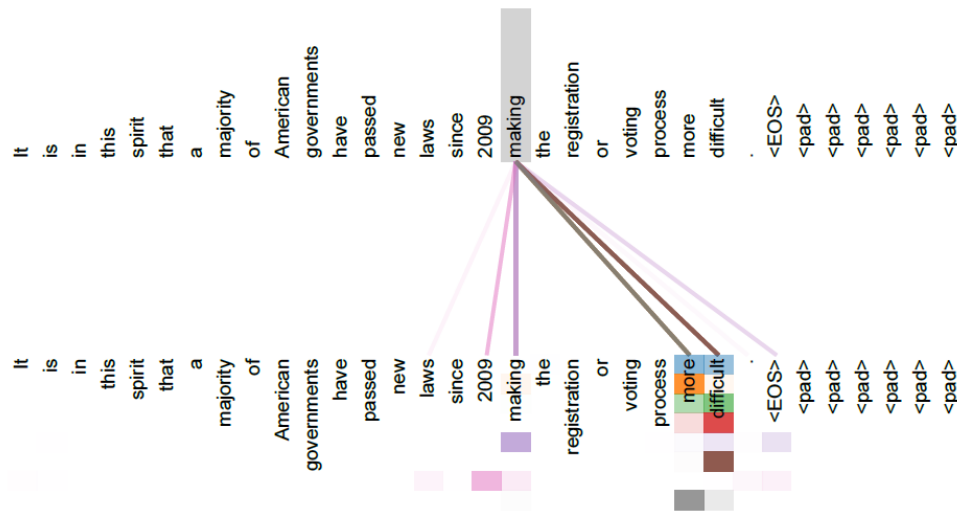
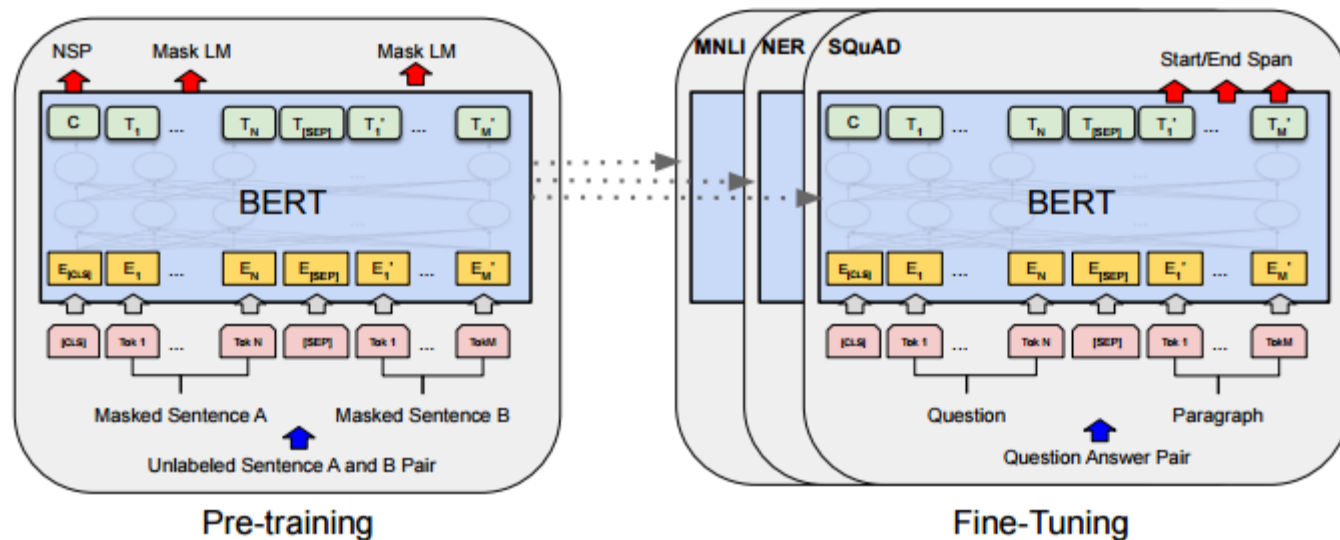
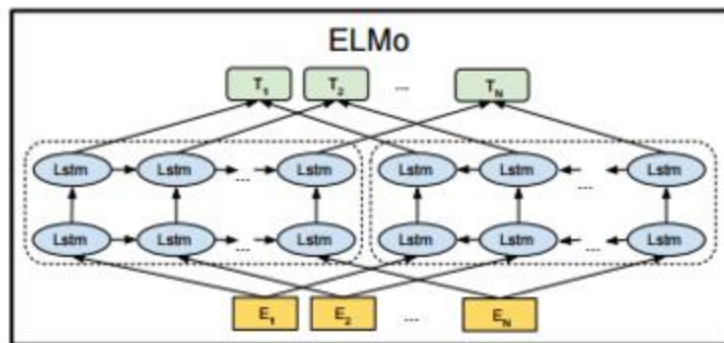
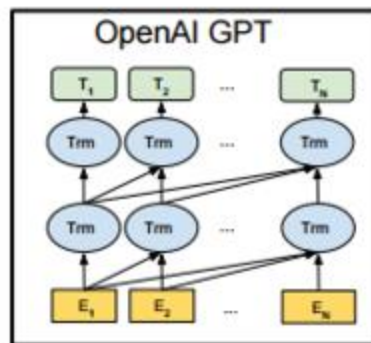
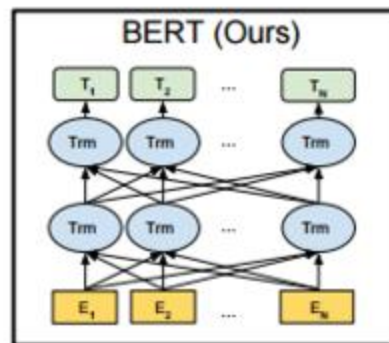
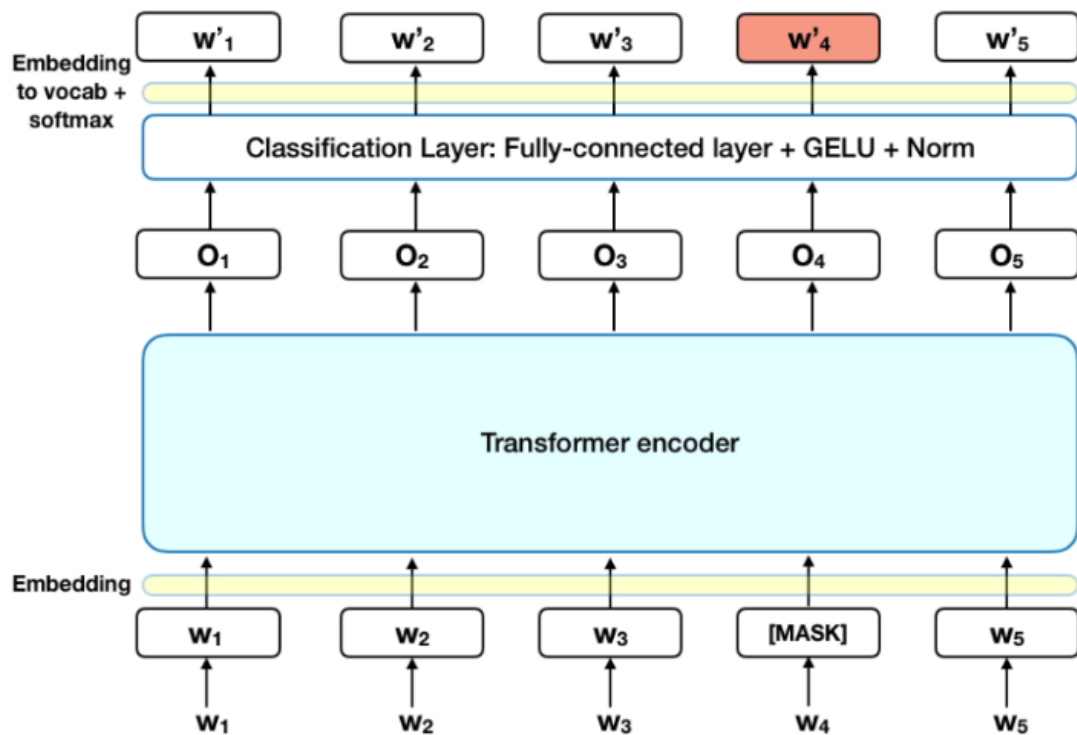


Figure 3: An example of the attention mechanism following long-distance dependencies in the encoder self-attention in layer 5 of 6. Many of the attention heads attend to a distant dependency of the verb 'making', completing the phrase 'making...more difficult'. Attentions here shown only for the word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







- 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]

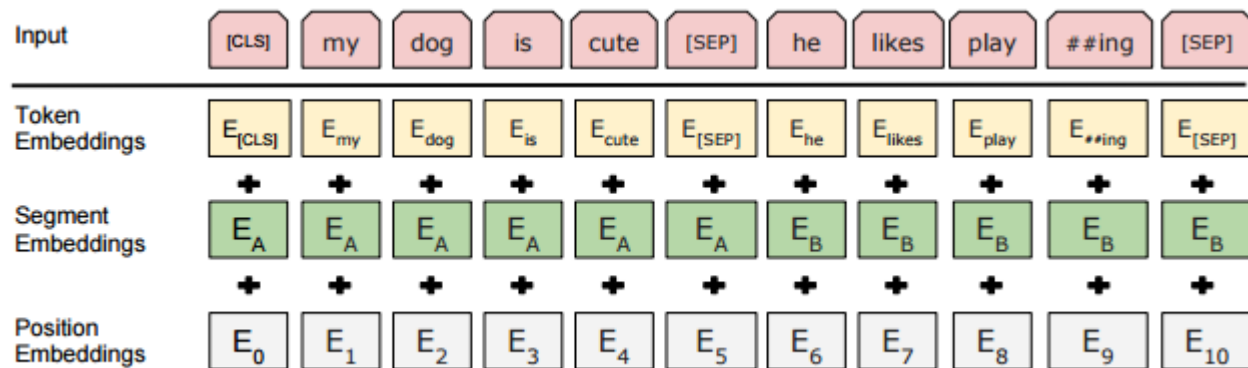
he bought a gallon [MASK] milk [SEP]

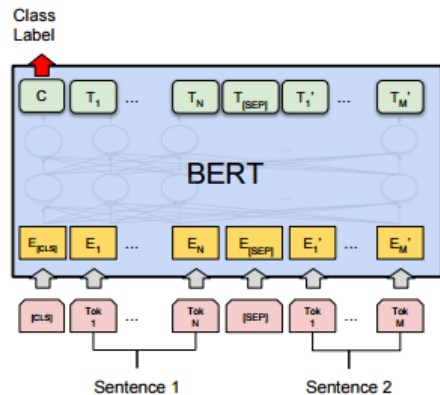
Label = IsNext

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

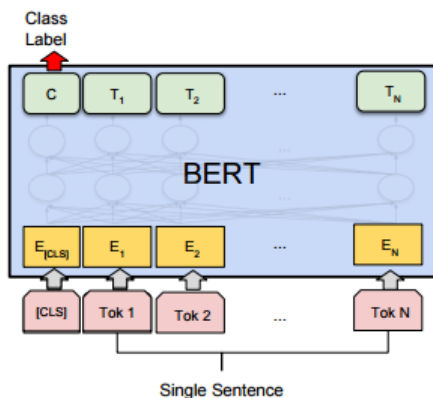
penguin [MASK] are flight ##less birds [SEP]

Label = NotNext

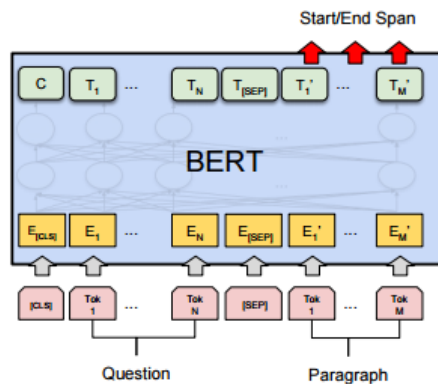




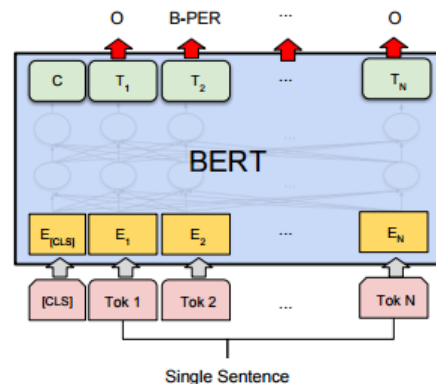
(a) Sentence Pair Classification Tasks:
MNLI, QQP, QNLI, STS-B, MRPC,
RTE, SWAG



(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.