(6주차) 언어모델(Language Model)

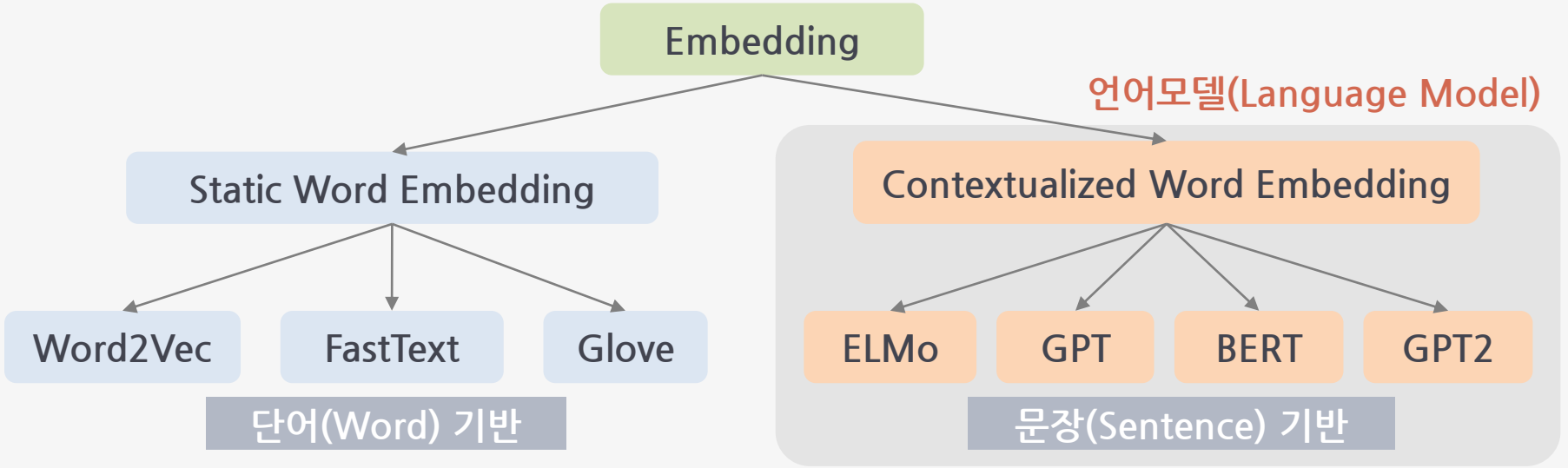
명지대학교
MYONGJI UNIVERSITY

언어모델 개요

언어모델(Language Model)

문맥을 고려하여 문장(Sentense) 기반으로 임베딩을 수행

ELMo, GPT, BERT, GPT2 등의 임베딩 방법



〈출처: 워드 임베딩. https://towardsdatascience.com. 20. 9〉

인공지능 언어모델 개요

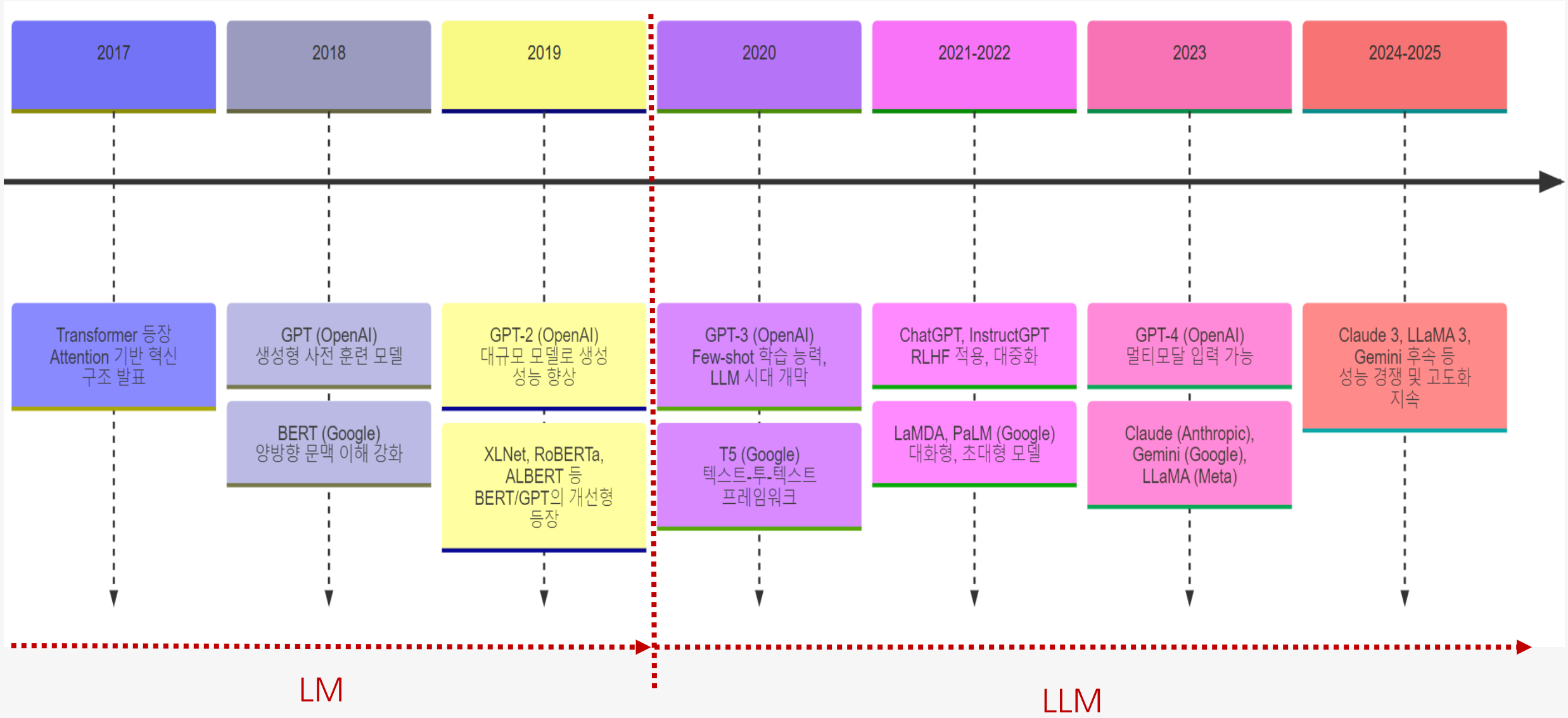
언어모델의 성장

트랜스포머 등장 이후 2018년 ~ 2020년 기간 동안 트랜스포머를 이용한 다양한 언어모델의 등장 및 인공지능 언어모델의 폭발적 성장

언어모델	발표시기	모델 발표 기관
ELMo	2018년 2월	Allen AI와 Washington University
GPT-1	2018년 5월	Open AI
BERT	2018년 10월	Google
GPT-2	2019년 2월	Open AI
KoBERT	2019년 6월	ETRI
XLNet	2019년 7월	Carnegie Mellon University와 Google Brain
RoBERTa	2019년 7월	Facebook AI Research
ALBERT	2019년 9월	Google과 TTIC(Toyota Technological at Chicago)
T5	2019년 10월	Google
KoGPT-2	2020년 4월	SKT와 AWS
GPT-3	2020년 6월	Open AI

인공지능 언어모델 개요

언어모델의 발전과정



인공지능 언어모델 개요

언어모델의 분류

문장의 학습 방향에 따라
순방향, 역방향, 양방향 언어모델로 분류

순방향 언어모델	OpenAI GPT 계열, ELMo
역방향 언어모델	ELMo
완전 양방향 언어모델	BERT 계열, XLNet

언어모델의 분류

ELMo는 순방향, 역방향 레이어를 각각 독립적으로 학습하기 때문에
진정한 의미의 양방향 모델로 보기 어려움

BERT

BERT(Bidirectional Encoder Representations from Transformers)

2018년 10월 구글이 발표한 Pre-Trained 기반 딥러닝 언어모델

- 대형 코퍼스에서 비지도학습(Unsupervised Learning)으로 General-Purpose Language Understanding 모델을 구축(Pre-training)
- 지도학습(Supervised Learning)으로 Fine-Tuning해서 QA, STS 등의 Downstream NLP 태스크에 적용하는 Semi-Supervised Learning 모델

BERT

트랜스포머로 구현되어 다양한 자연어 처리 분야에 활용

트랜스포머의 인코더만 사용, BERT Base는 12개의 트랜스포머 블록, Bert Large는 24개의 트랜스포머 블록을 사용

사전 훈련 데이터로 BooksCorpus 8억 단어, Wikipedia 25억 단어를 사용

- BERT
  - BERT의 학습 절차

BERT

BERT의 학습 절차

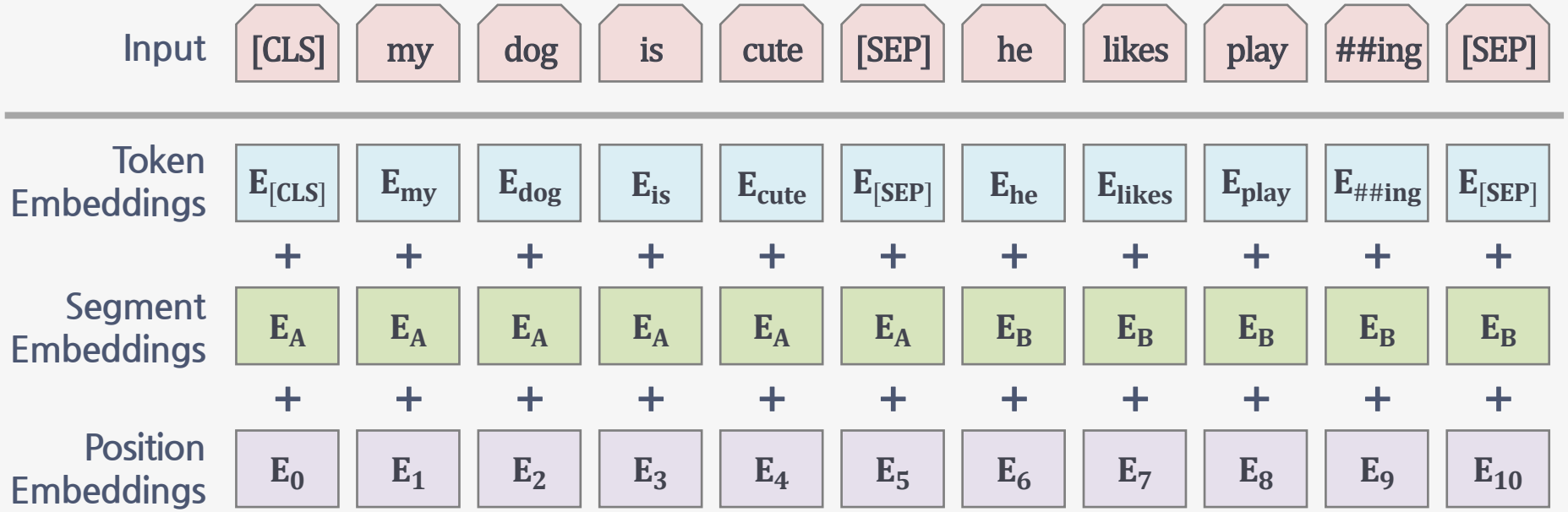
Input Embeddings

- BERT는 트랜스포머와 달리 Positional Encoding을 사용하지 않고 Position Embeddings를 사용
- Input Embeddings

  = Position Embeddings + Segment Embeddings + Token Embeddings

BERT

BERT의 학습 절차



[SEP] 문장의 끝을 나타내는 식별자로 두 문장을 구분하는 역할

[CLS] 문장의 맨 앞에는 클래스를 뜻하는 CLS를 추가

BERT

BERT의 학습 절차

Pre-Training

- Pre-Training 단계에서 BERT는 언어의 특성을 잘 학습하기 위해 MLM(Masked Language Model)과 NSP(Next Sentence Prediction) 방식 사용

BERT

BERT의 학습 절차

MLM(Masked Language Model)

- 문장의 다음 단어를 예측하는 것이 아니라 문장 내 랜덤한 단어를 마스킹하고 예측하도록 하는 방식
- Word2Vec의 CBOW 모델과 유사

CBOW는 Context 토큰을 Center 토큰이 되도록 학습하고 Weights를 벡터로 가짐

MLM은 마스킹된 토큰을 맞추도록 학습한 결과를 직접 벡터로 갖기 때문에 보다 직관적인 방식

BERT

BERT의 학습 절차

MLM(Masked Language Model)

- 문장의 다음 단어를 예측하는 것이 아니라 문장 내 랜덤한 단어를 마스킹하고 예측하도록 하는 방식
- Word2Vec의 CBOW 모델과 유사

마스킹은 전체 단어의 15% 정도의 비율 사용

모든 토큰을 마스킹 하는게 아니라
80% 정도의 토큰에 대해서만 <MASK>로 처리

10%는 Random한 단어

나머지 10%는 정상적인 단어를 그대로 사용

BERT

BERT의 학습 절차

Attention Masking

- BERT는 Inference 단계에서 Zero Padding으로 입력된 토큰에 대해서는 항상 마스킹 처리

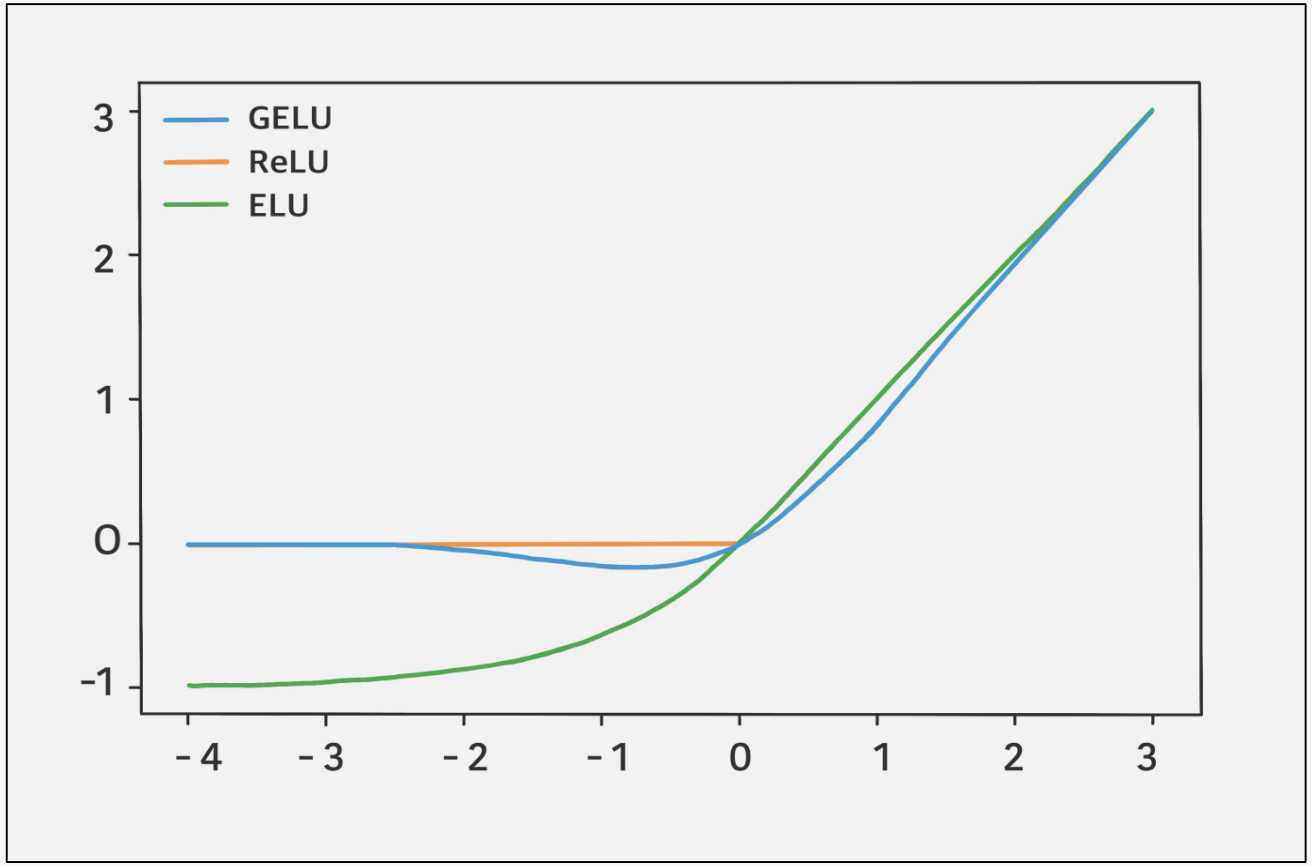
  해당 토큰에 대해서는 Penalty를 부과하여 Attention score 계산

- 트랜스포머는 FFNN(Feed Forward Neural Network)에 ReLU Activation을 적용했지만 BERT는 GELU(Gaussian Error Linear Unit) Activation을 적용

BERT

BERT의 학습 절차

Attention Masking



BERT

BERT의 학습 절차

NSP(Next Sentence Prediction)

- 주어진 두 문장에 대해 두 번째 문장이 코퍼스 내에서 첫 번째 문장 바로 다음에 오는지 여부를 예측
- 이 방식의 사용 이유: BERT는 전이 학습으로 사용되고 QA와 NLI(Natural Language Inference) 등의 태스크에서는 MLM으로 학습하는 것은 충분하지 않았기 때문

두 문장이 실제로 연결된 문장인지의 여부를 예측하기 위해

- 50% 비율의 실제 연결된 문장
- 랜덤하게 추출된 연결되지 않은 문장 사용

BERT

BERT의 학습 절차

NSP(Next Sentence Prediction)

- 주어진 두 문장에 대해 두 번째 문장이 코퍼스 내에서 첫 번째 문장 바로 다음에 오는지 여부를 예측
- 이 방식의 사용 이유: BERT는 전이 학습으로 사용되고 QA와 NLI(Natural Language Inference) 등의 태스크에서는 MLM으로 학습하는 것은 충분하지 않았기 때문
- [CLS] 벡터의 Binary Classification 결과를 예측하도록 학습

BERT

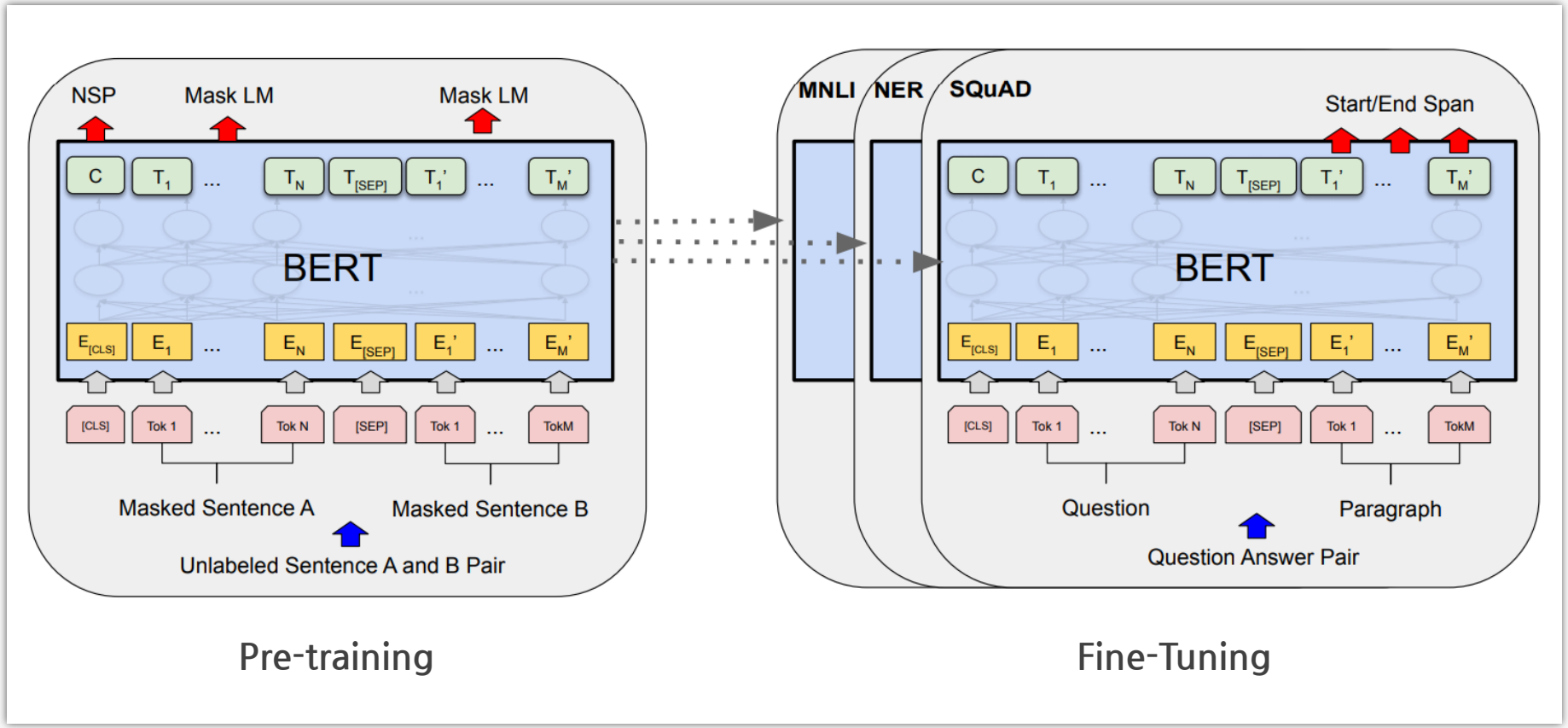
BERT의 학습 절차

Transfer Learning

- 대량 코퍼스로 BERT 언어모델을 적용하고, BERT 언어모델 출력에 추가적인 모델을 추가하여 문제 해결을 위한 의도에 맞는 모델 생성

BERT

BERT의 학습 절차

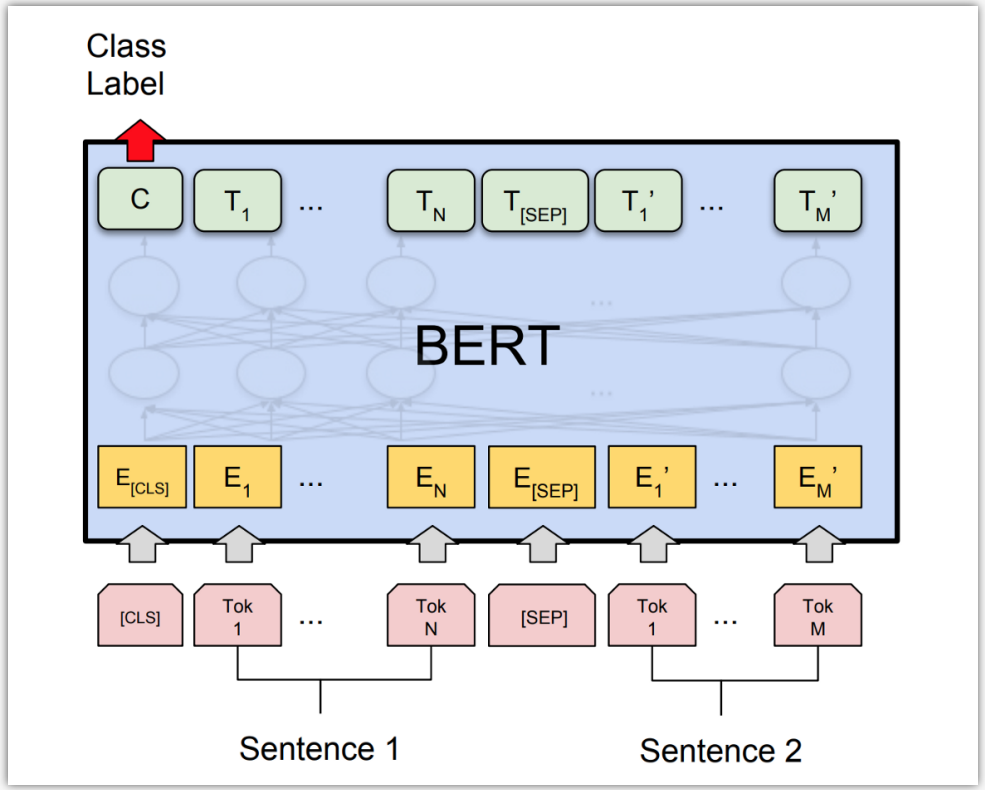


BERT

BERT의 학습 절차

두 문장의 분류

- 두 문장을 입력 데이터로 사용하여 문장을 분류하는 TASK

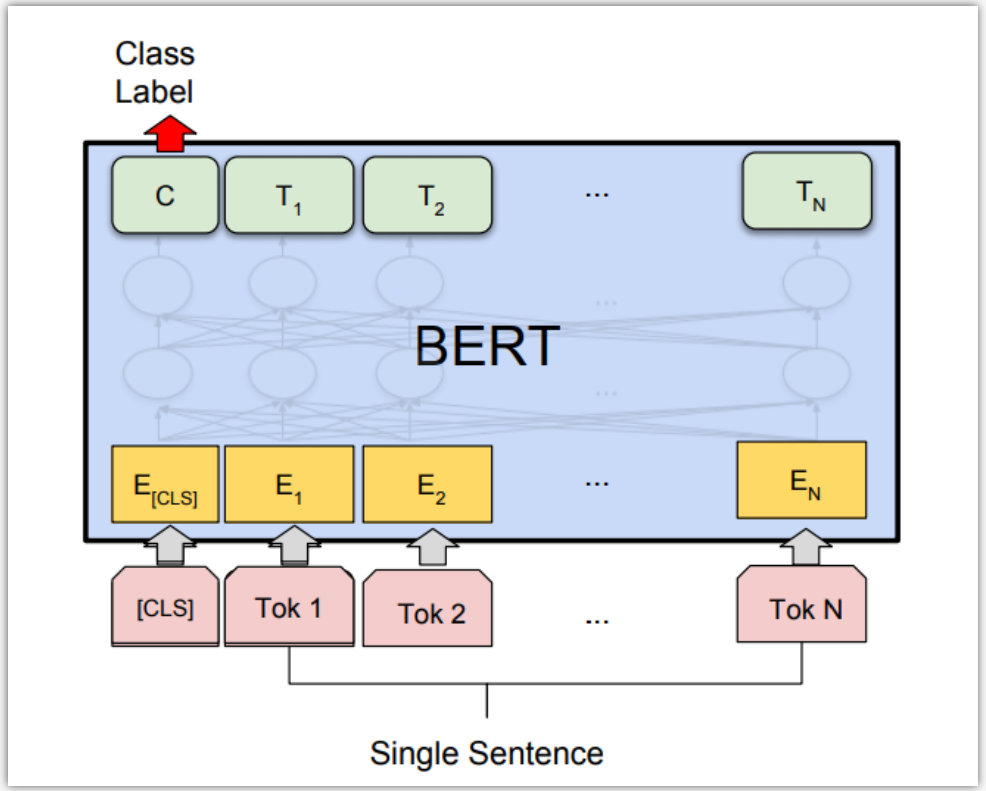


BERT

BERT의 학습 절차

한 문장의 분류

- 한 문장을 토큰화 하여 입력 데이터로 사용하여 분류

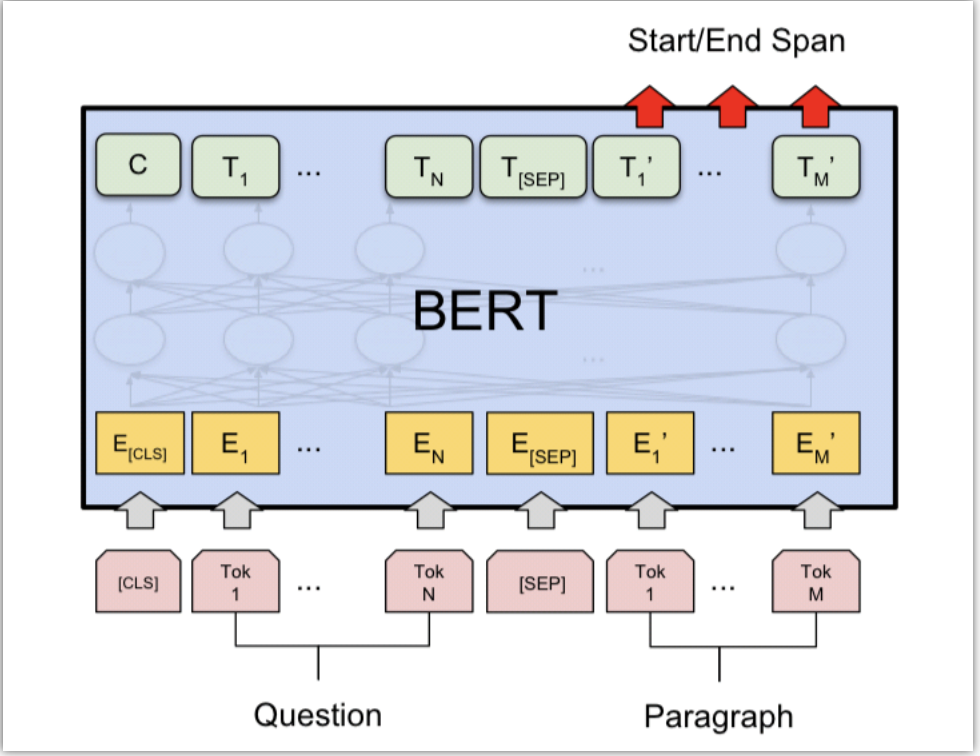


BERT

BERT의 학습 절차

Question Answering

- Question과 Paragraph를 입력 데이터로 사용하여 Answer를 예측

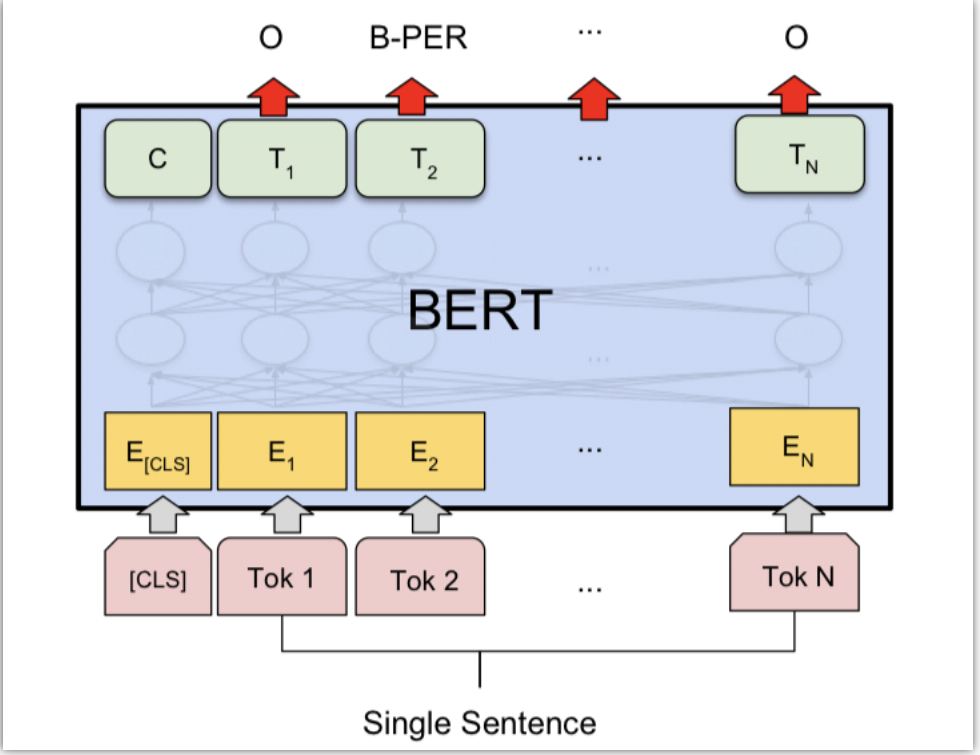


BERT

BERT의 학습 절차

개체명 인식
(Named Entity Recognition)

- 한 문장을 입력 데이터로 사용하여 문장 내 토큰명 개체명 분류



RoBERTa

RoBERTa(A Robustly Optimized BERT)

BERT보다 성능을 한 단계 업그레이드한 버전

- 모델 크기를 크게 하여 성능을 높이기 위한 방법 적용
- 훈련 데이터의 양(13GB→160GB), 학습 횟수(125,000회→500,000회), 배치 크기(256→8,192), 사전 크기(32,000→50,000)
- BERT의 Next Sentence Prediction은 훈련에서 제외

ALBERT

ALBERT(A Lite BERT)

BERT보다 가벼운 모델

- 모델 매개변수 수를 줄여 같은 구조의 모델에서의 메모리 사용량을 줄이고 학습 속도는 개선
- 각 단어를 저차원의 임베딩 벡터로 표현

  → 이를 다시 모델 은닉층의 차원 수만큼 확장

- 트랜스포머 인코더 블록 간 매개변수를 공유

ALBERT

ALBERT(A Lite BERT)

- BERT(Large)에 비해 매개변수 수는 1/18

  3억 3,400만개 → 1,800만개

- GLUE 성능을 일정 수준 유지하면서(85.2→82.4) 학습 속도를 1.7배 개선

- BERT의 Next Sentence Prediction 대신 두 문장의 연관 관계를 예측하는 문장 순서 예측(Sentence Order Prediction, SOP)을 통해 훈련 성능 개선

T5

T5

- 하나의 모델로 모든 문제를 해결할 수 있도록 모델의 크기를 대폭 확대
- GLUE와 SQuAD 등에 포함된 다양한 자연어 이해(NLU) 데이터 셋을 사전 훈련
- 정제 텍스트 데이터 700GB, 모델 매개변수는 110억개 규모

  정제 텍스트 데이터는 RoBERTa의 4.4배 / 모델 매개변수는 32배

- 빈칸 하나의 예측값이 다른 빈칸 예측에 영향을 주지 않는 BERT의 한계를 극복하기 위해 Seq2Seq 구조의 Masked Language Model을 적용

T5는 SuperGLUE에서 인간과 비슷한 성능을 달성

KoBERT

KoBERT

- 엑소브레인 사업에서 한국어의 특성을 반영하여 개발한 BERT 언어모델

엑소브레인: 과학기술정보통신부와 IITP의 혁신성장동력 프로젝트로 추진 중인 사업

- ETRI 엑소브레인 연구진이 배포하는 한국어 최첨단 딥러닝 언어모델은 한국어분석·기계독해·문서분류 등 다양한 태스크에 활용 가능

5종의 한국어 처리 태스크에서 구글이 배포한 한국어 언어모델과 비교 평가한 결과, ETRI의 언어모델이 평균 4.5% 성능 우수

- 5종: 의미역 인식, 기계독해, 단락 순위화, 문장 유사도 추론, 문서 주제분류

GPT 언어모델

GPT(Generative Pre-training Transformers)

GPT

- OpenAI에서 발표(2018)
- Transformer Decoder로 구현
- 이전 단어를 입력으로 하여 다음 단어를 출력으로 하는 단방향 언어모델
- 자연어 생성(NLG)에서 우수한 성능을 보임

GPT 2

- 웹 페이지 크롤링을 통해 4500만개의 링크에서 Text 40GB를 활용(소설 3만5천개)
- 모델 파라미터 수는 15억개 (Bert-Large 3.4억개)
- 특정 Task에 맞게 Finetuning하지 않은 상태에서 기능을 측정 Zero-shot Learning

거대 언어모델(LLM)의 등장

GPT3

- 웹 페이지 크롤링을 통해 수집한 4990억개 데이터셋 중에서 가중치 샘플링을 통해서 3000억(300B)개로 구성된 데이터셋으로 사전 학습(pre-training)
- 모델 파라미터 수는 1750억개

Fine-tuning 없이
자연어 처리 벤치마크 테스트에서 최고의 성능 달성

OpenAI는 훈련된 모델 또는 전체 소스코드는 미공개

전이 학습의 개요

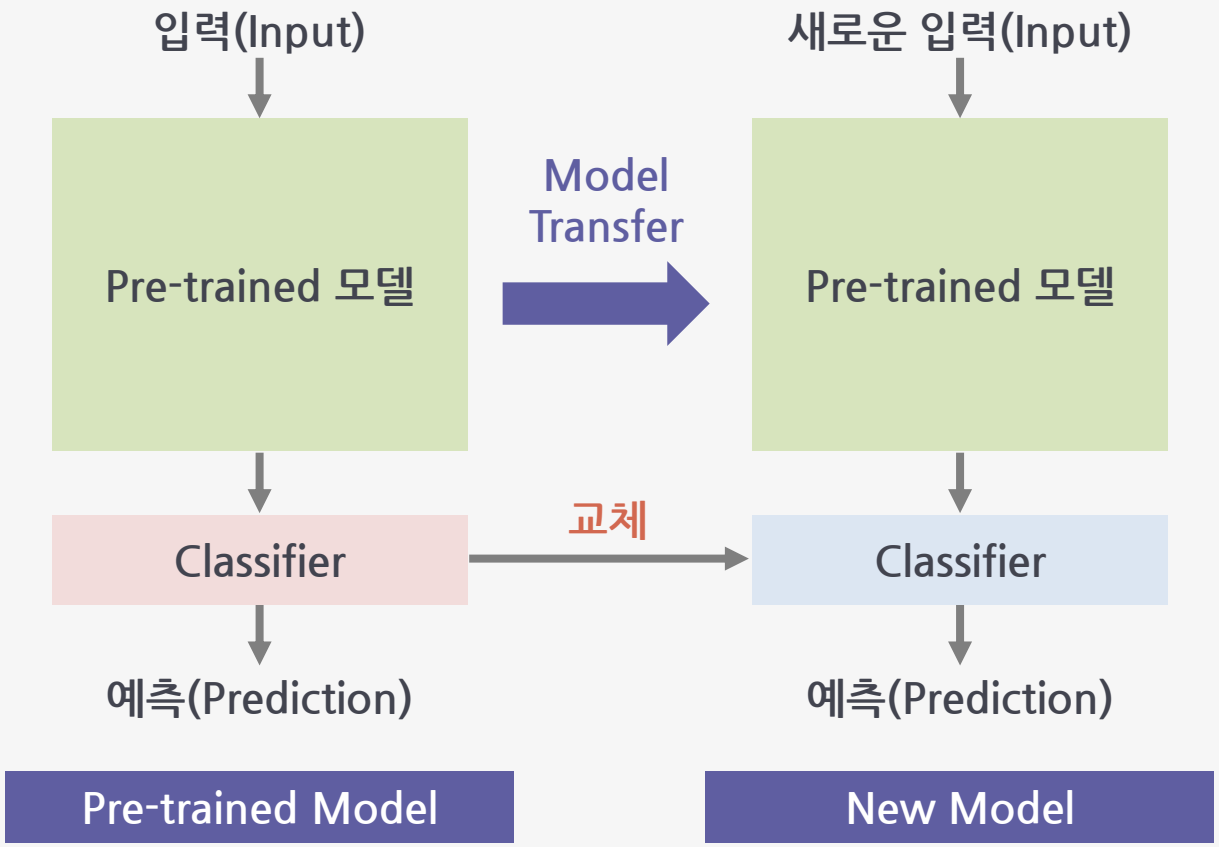
전이 학습(Transfer Learning)이란?

대규모 데이터 셋으로 사전에 학습이 완료된
언어모델(Pre-trained Language Model)을 사용

사전 학습된 모델을 해결하고자하는 문제에 적합하도록
미세조정(Fine-tuning)하여 학습시키는 기법

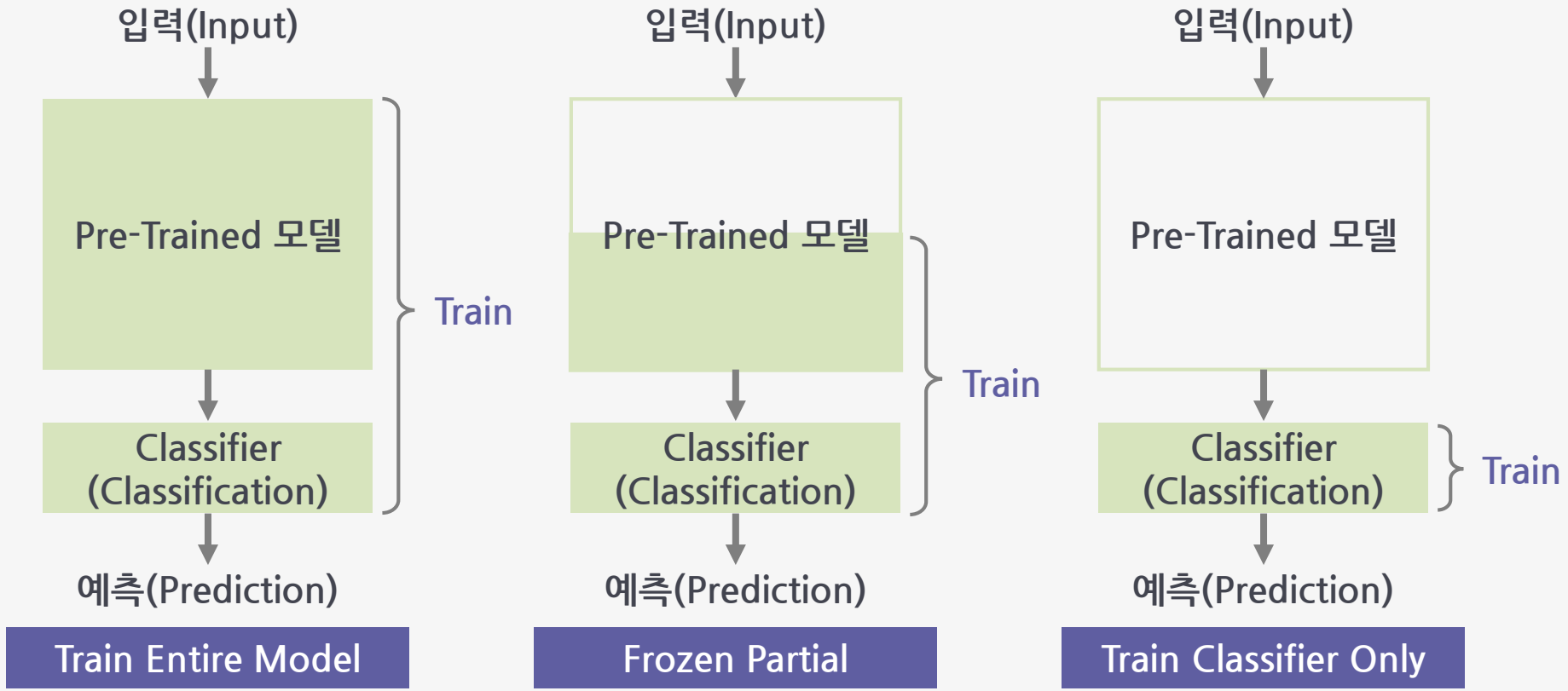
언어모델 전이 학습

전이 학습의 개요



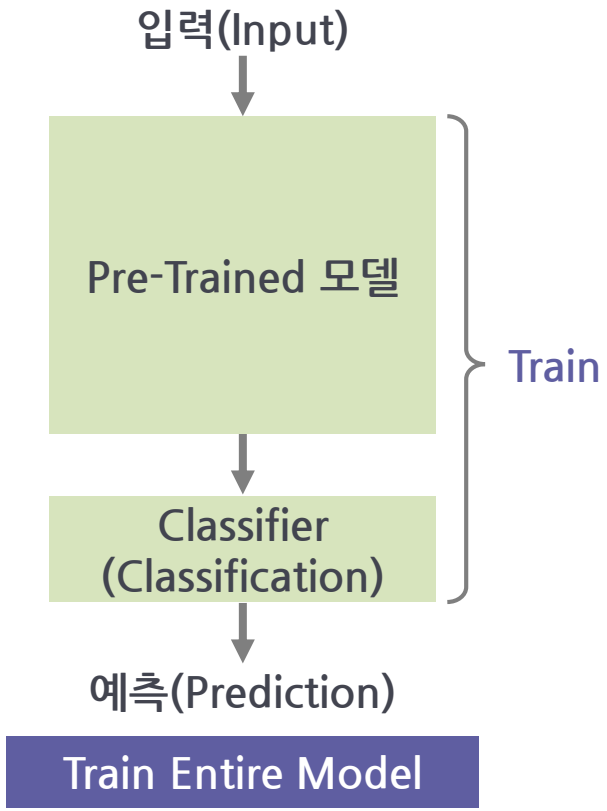
Pre-Trained 모델의 Fine-Tuning 전략

- 전이 학습을 통해 새롭게 생성된 New Model에 대한 최적화된 Train 전략
- Train Entire Model, Frozen Partial, Train Classifier Only 전략이 존재



Pre-Trained 모델의 Fine-Tuning 전략

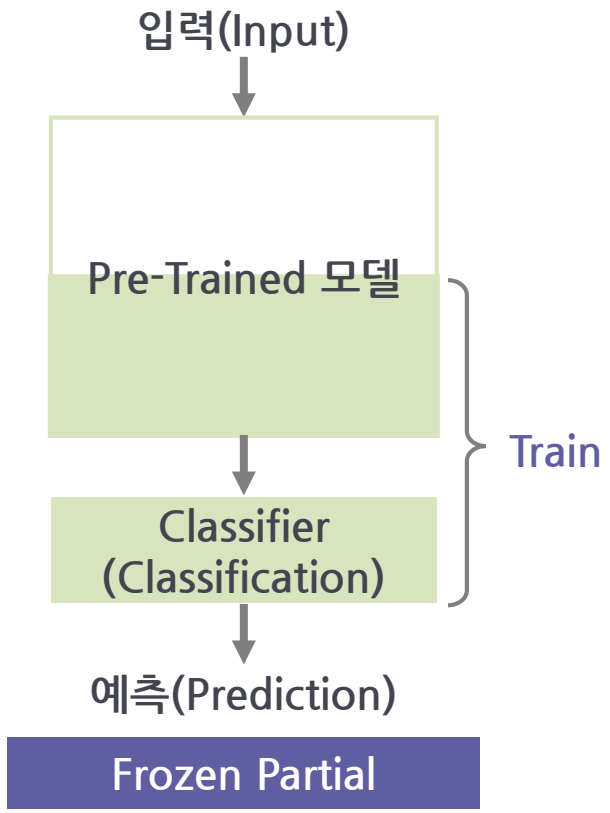
Train Entire Model 전략



- 모델 전체를 Training하는 전략
- Pre-Trained 모델의 구조만 사용하면서, New Model의 데이터 셋에 맞게 모델 전체를 새롭게 Training하는 전략

Pre-Trained 모델의 Fine-Tuning 전략

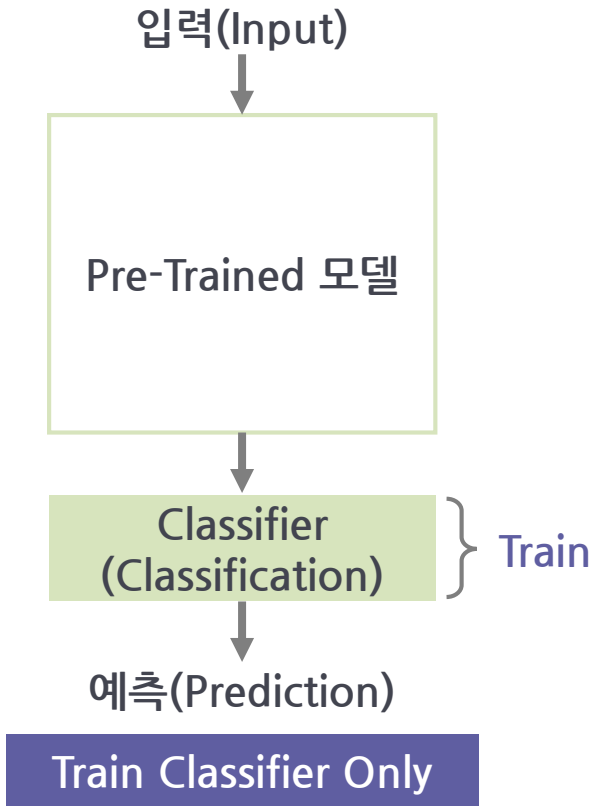
Frozen Partial 전략



- Pre-Trained 모델의 일부만 고정
- Pre-Trained 모델의 일부분은 고정시킨 상태에서 Pre-Trained 모델 나머지 부분과 Classifier만 Training하는 전략

Pre-Trained 모델의 Fine-Tuning 전략

Train Classifier Only 전략

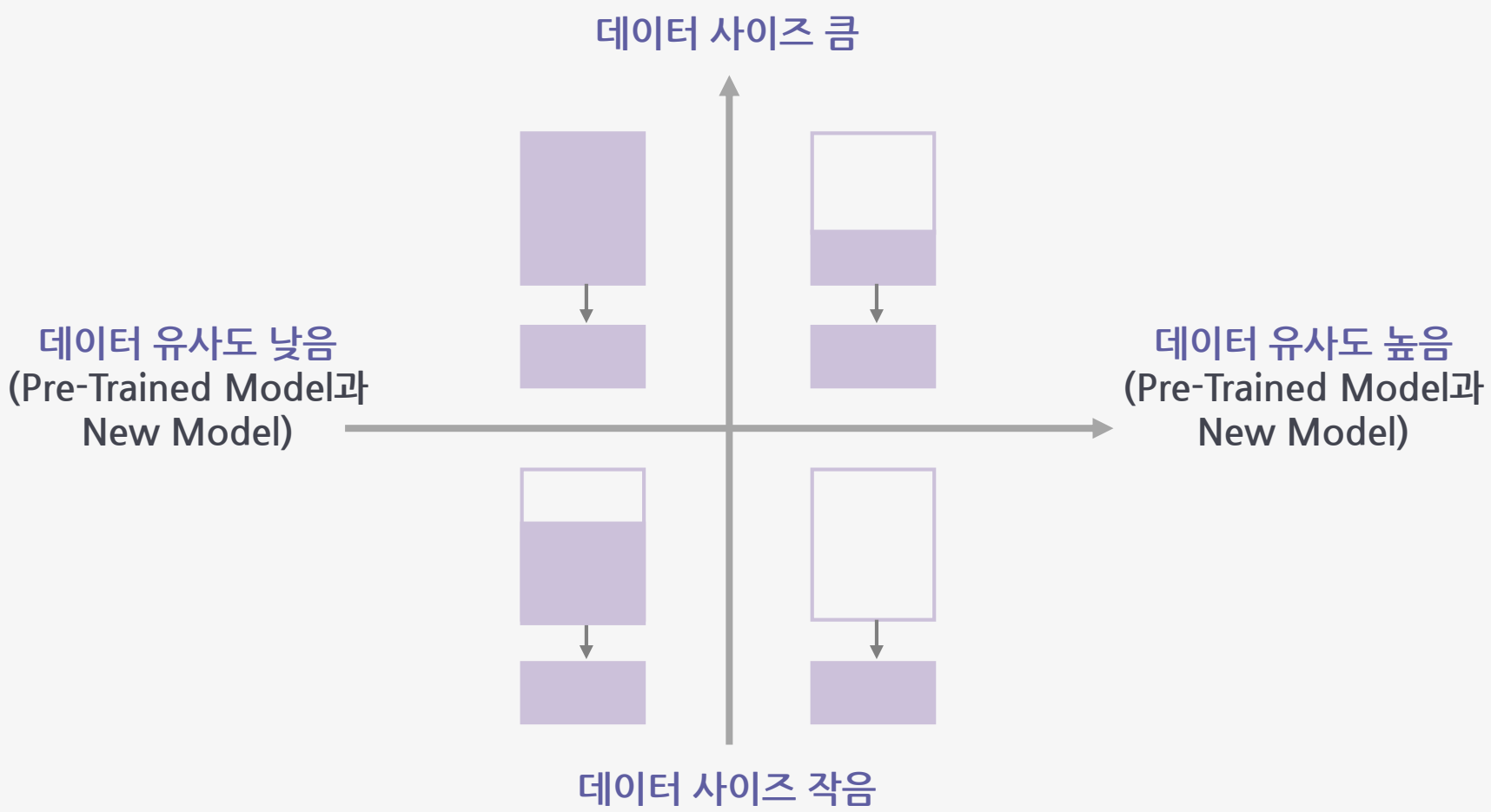


- Pre-trained 모델 **전체를 고정**한 상태에서, New Model의 Classifier만 Training하는 전략

언어모델 전이 학습

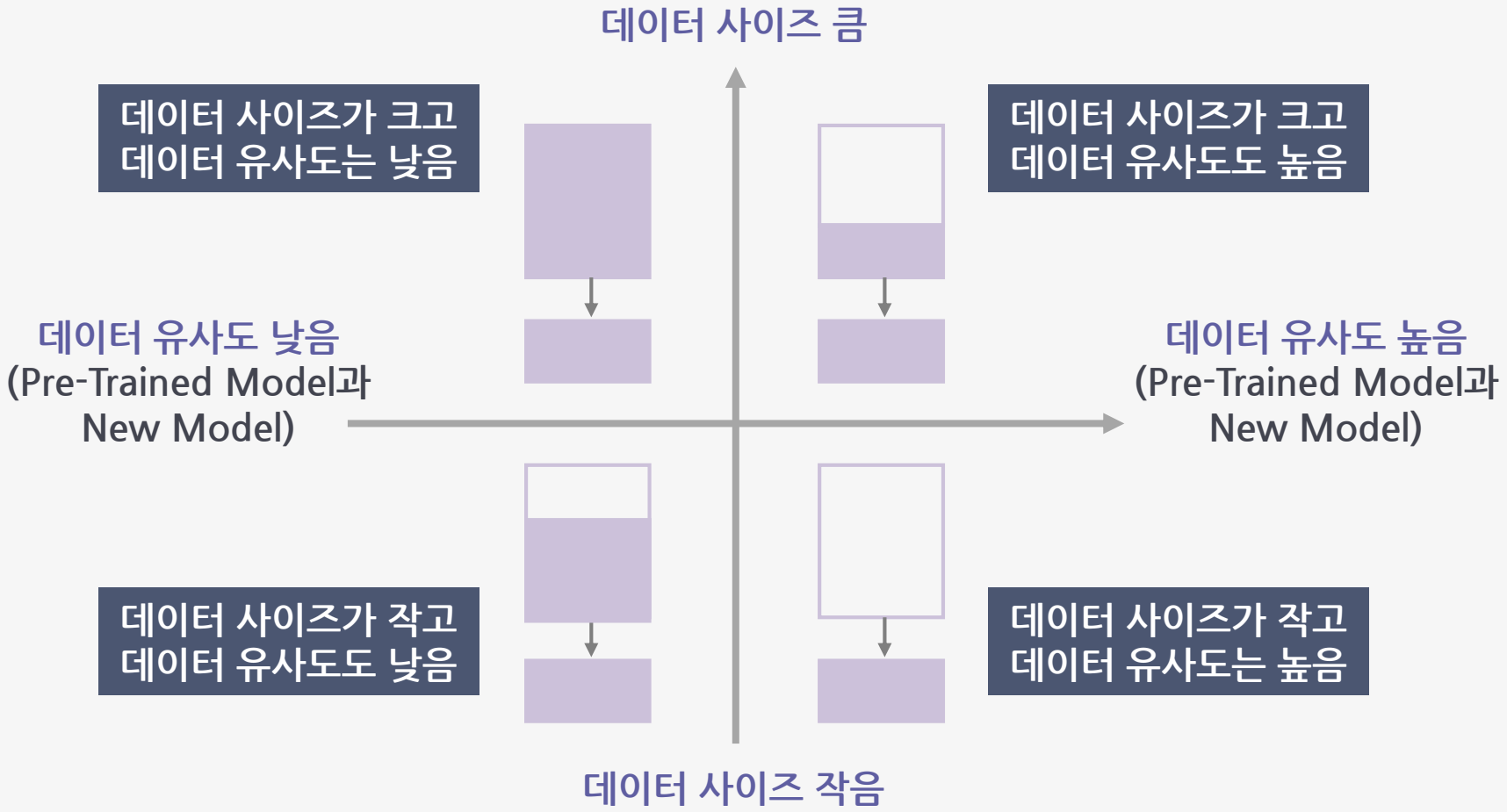
전이 학습 전략의 선택

데이터 사이즈와 데이터 유사도에 따라 선택 필요



전이 학습 전략의 선택

데이터 사이즈와 데이터 유사도에 따라 선택 필요



(참고) Word2Vec 개요

희소 표현과 원 핫 인코딩

희소 표현(Sparse Representation)이란?

표현하고자 하는 단어의 인덱스의 값만 1이고,
나머지 인덱스에는 전부 0으로 표현되는 벡터 표현 방법(One Hot Encoding)

(참고) Word2Vec 개요

희소 표현과 원 핫 인코딩

원 핫 인코딩(One Hot Encoding)이란?

전체 레이블(Label) 수에 대해 표현하고자 하는
특정 단어(Word)만 '1'로 활성화한 벡터로 변환하는 방법

사과	1	0	0	0	0
개	0	1	0	0	0
배	0	0	1	0	0
고양이	0	0	0	1	0
토마토	0	0	0	0	1

단어의 의미와 관계를 전혀 고려하지 않음

벡터의 크기는 총 단어 수만큼의 Sparse Vector

정답(Label)을 인식하게 하는 용도로 주로 사용

(참고) Word2Vec 개요

분산 표현과 Word2Vec

분산 표현(Distributed Representation)이란?

분포 가설(Distributional Hypothesis) 가정 하에 문자열을 표현하는 방법

비슷한 위치에서 등장하는 단어들은
비슷한 의미를 가질 확률이 높다는 가설

» '강아지'는 주로 '귀엽다', '예쁘다', '멍멍' 등의 단어와 함께 등장

(참고) Word2Vec 개요

분산 표현과 Word2Vec

Word2Vec이란?

대표적인 분산 표현 방법으로 분포 가설을 이용

코퍼스(말뭉치)에서 각 단어를 단어의 의미에 따라
여러 차원에 **분산**하여 벡터로 표현

단어 간 유사도 계산 등의 **연산 가능**

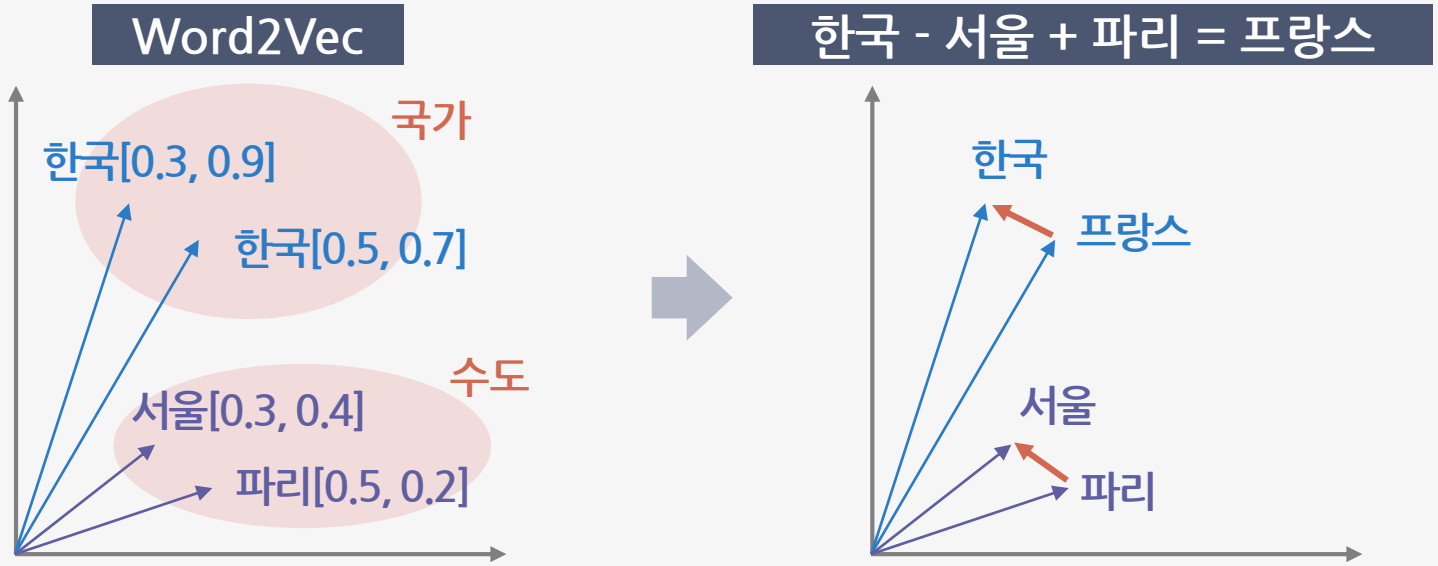
(참고) Word2Vec 개요

분산 표현과 Word2Vec

Word2Vec을 이용한 워드 임베딩

Word2Vec은 유사한 단어가 유사한 위치에 임베딩

단어 벡터 간 연산을 통해 의미 파악 가능



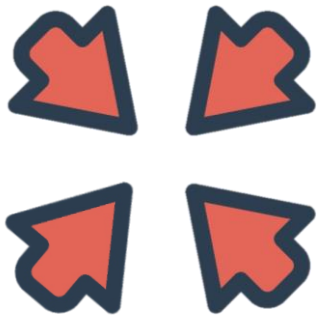
Word2Vec은 CBOW와 Skip-gram 등 두 가지 방법이 존재

(참고) CBOW와 Skip-gram

CBOW와 Skip-gram 개요

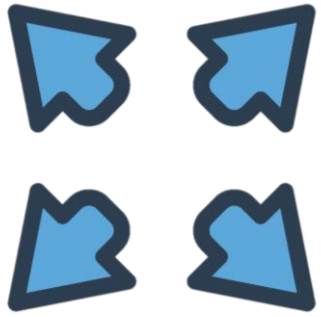
CBOW란? (Continuous Bag-of-Words)

주변 단어(Context Word)를
임베딩하여
중심 단어(Target Word)를
예측하기 위한 Word2Vec 방법



Skip-gram이란?

중심 단어(Target Word)를
임베딩하여
주변 단어(Context Word)를
예측하기 위한 Word2Vec 방법

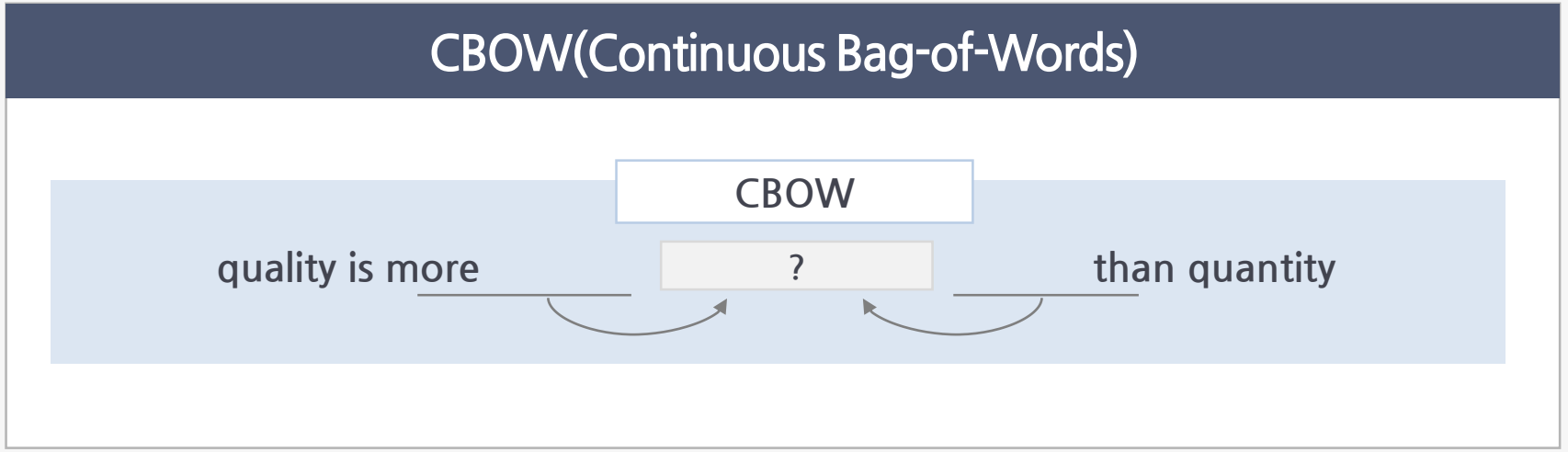


(참고) CBOW와 Skip-gram

CBOW과 Skip-gram의 예측 단어

문장에 대해 CBOW와 Skip-gram을 이용해 단어 예측

CBOW는 주변 단어(context word)를 임베딩(embedding)하여
중심 단어(target word)를 예측

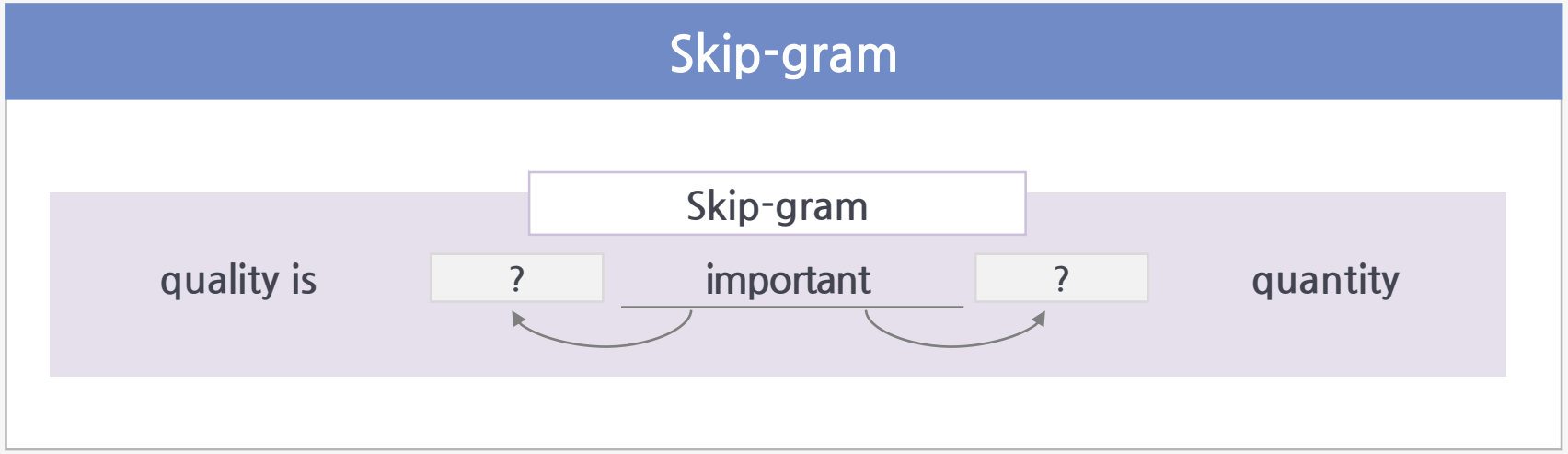


(참고) CBOW와 Skip-gram

CBOW과 Skip-gram의 예측 단어

문장에 대해 CBOW와 Skip-gram을 이용해 단어 예측

skip-gram은 중심 단어(target word) 를 임베딩(embedding)하여
주변 단어(context word)를 예측



(참고) CBOW와 Skip-gram

CBOW와 Skip-gram의 구조

CBOW와 Skip-gram은 서로 대칭적 구조로 이루어 짐