AiDA Lab.
Artificial Intelligence & Data Analysis Laboratory

AI 고급 : 자연어 처리 과정(정규)

19. BERT (2)

Seokhwan Yang

BERT 활용하기

- 사전 학습된 BERT를 사용하는 방법을 알아보자
  - 사전 학습된 BERT 모델의 구성 살펴보기
  - 사전 학습된 BERT 모델을 특징 추출기로 사용하는 방법
  - 허깅페이스의 트랜스포머 라이브러리 살펴보기
  - 사전 학습된 BERT 모델에서 임베딩 추출하기
  - BERT 모델의 모든 인코더 레이어에서 임베딩 추출하기
  - 하위 작업을 위해 사전 학습된 BERT 모델을 파인 튜닝 하는 방법

사전 학습된 BERT 모델 탐색

- 이전 수업 내용
  - MLM, NSP를 사용하여 BERT를 사전 학습 시키는 방법

    → 그러나 BERT를 처음부터 사전 학습 시키는 것은 많은 계산 비용이 요구됨

    → 사전 학습된 공개 BERT 모델을 이용하는 것이 효과적

  - Google에서 제공하는 BERT 모델을 찾아서 활용하자

사전 학습된 BERT 모델 탐색

- **Google에서 제공하는 사전 학습된 BERT 모델**
  - **https://github.com/google-research/bert**
    - 해당 URL에서 제공하는 오른쪽과 같은 테이블에서 원하는 각 모델을 선택하여 사전 학습된 BERT 모델을 다운로드
    - Cased/Uncased 구분된 모델도 제공
      - Cased: 토큰에 대소문자가 모두 존재하는 모델
      - Uncased: 토큰에 대하여 소문자화를 수행한 모델
    - WWM(Whole Word Masking) 방법으로 학습된 모델도 제공

	H=128	H=256	H=512	H=768
L=2	**2/128 (BERT-Tiny)**	2/256	2/512	2/768
L=4	4/128	**4/256 (BERT-Mini)**	**4/512 (BERT-Small)**	4/768
L=6	6/128	6/256	6/512	6/768
L=8	8/128	8/256	**8/512 (BERT-Medium)**	8/768
L=10	10/128	10/256	10/512	10/768
L=12	12/128	12/256	12/512	**12/768 (BERT-Base)**

L: 인코더 레이어 수
H: 은닉 유닛의 크기 (표현의 크기)

사전 학습된 BERT 모델 탐색

- 사전 학습된 모델의 활용법
  - 임베딩을 추출해서 특징 추출기로 사용
  - 사전 학습된 BERT 모델을 텍스트 분류, 질문-응답 등과 같은 하위 작업에 맞게 파인 튜닝해서 사용

사전 학습된 BERT에서 임베딩 추출하기

- 다음의 문장에서 각 단어의 문맥 임베딩을 추출해보자
  - 입력 문장: I love Paris (나는 파리를 사랑한다).

- 문장에서 각 단어의 문맥 임베딩을 추출하려면
  1. 문장을 토큰화하고
  2. 사전 학습된 BERT에 토큰을 입력하여
  3. 토큰에 대한 임베딩을 반환

토큰 수준(단어 수준) 표현 외에도
문장 수준의 표현을 얻을 수도 있음

사전 학습된 BERT에서 임베딩 추출하기

- **사전 학습된 BERT에서 단어 수준 및 문장 수준 임베딩 추출하기**
  - **감정 분석 작업을 위한 데이터셋 예시**

문장	레이블
I love Paris	1
Sam hated the movie	0
It was a great day	1
The song is not good	0
…	…
We loved the game	1

1: 긍정적인 감정
0: 부정적인 감정

- 데이터셋은 텍스트 데이터 → 모델에 직접 입력 불가
  → 벡터화가 우선 되어야 함

- 벡터화 방안
  - TF-IDF, Word2Vec 등 다양한 방법 사용 가능(문맥 독립적)
  - BERT 모델을 이용한 벡터화(문맥 기반 임베딩)

사전 학습된 BERT에서 임베딩 추출하기

- **BERT 모델을 이용한 벡터화 방법**

I love Paris

1. **워드피스 토크나이저로 문장 토큰화**

tokens=[I, love, Paris] → tokens=[[CLS], I, love, Paris, [SEP]]

2. **학습셋의 모든 문장을 토큰화**

3. **그런데 모든 토큰의 길이를 동일하게 유지해야 함 → 토큰 길이를 7로 유지**

tokens=[[CLS], I, love, Paris, [SEP], [PAD], [PAD]]

사전 학습된 BERT에서 임베딩 추출하기

4. [PAD] 토큰은 실제 토큰이 아니라는 것을 모델에게 이해시키기

   → 어텐션 마스크 적용

   attention_mask=[1, 1, 1, 1, 1, 0, 0]

   [PAD] 토큰이 있는 위치에만 "0" 설정

5. 모든 토큰을 고유한 토큰 ID에 매핑

   token_ids=[101, 1045, 2293, 3000, 102, 0, 0]

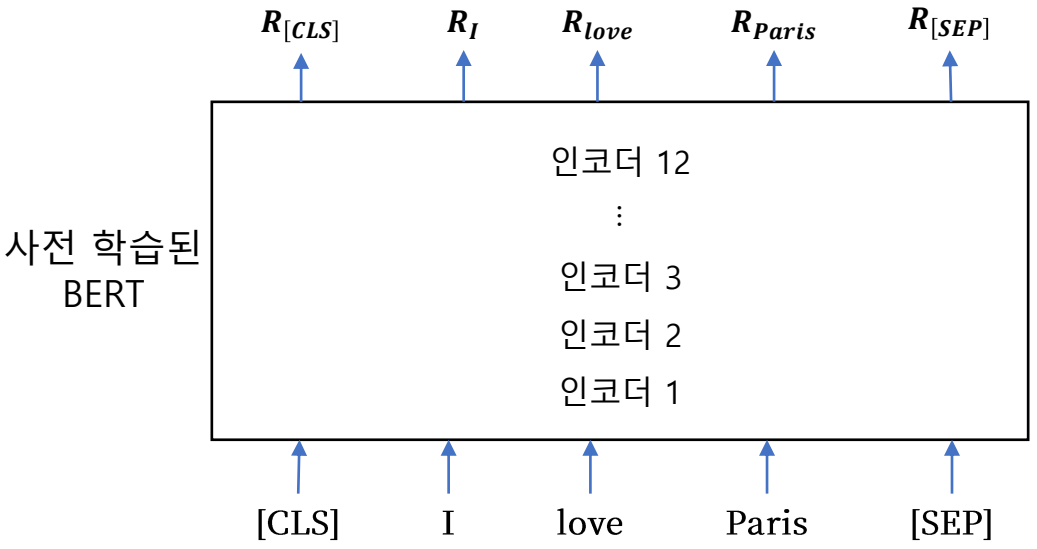
   id 값은 임의 값이므로 신경쓰지 말것

- ID 101: [CLS]
- ID 1045: I
- ID 2293: Paris
- ID 3000: [SEP]

어텐션 마스크, token_ids를 사전 학습된 BERT 모델에 입력

→ 각 토큰의 벡터 표현(임베딩) 획득

사전 학습된 BERT에서 임베딩 추출하기



- 토큰을 입력으로 공급하면
- 인코더 1은 모든 토큰의 표현을 계산해서 인코더 2로 보냄
- 인코더 2는 인코더 1이 계산한 표현을 입력으로 가져와서 다음 인코더인 인코더 3으로 전송
- … 반복 …
- 최종 인코더인 인코더 12는 문장에 있는 모든 토큰의 최종 표현 벡터(임베딩)를 반환

- 사전 학습된 BERT-base 모델을 사용한다면 각 토큰의 표현 크기는 768

사전 학습된 BERT에서 임베딩 추출하기

- 전체 문장의 표현은 어떻게 얻을 수 있나?
  - 문장 시작 부분에 있는 [CLS] 토큰의 표현은 전체 문장의 집계 표현을 보유
    → 다른 모든 토큰의 임베딩을 무시하고 [CLS] 토큰의 임베딩을 가져와서 문장의 표현으로 할당할 수 있음
  - "I love Paris" 문장의 표현은 [CLS] 토큰에 해당하는 $R_{[CLS]}$의 표현 벡터
  - 유사한 방식으로 학습셋의 모든 문장의 벡터 표현 계산 가능
  - 학습셋의 모든 문장의 문장 표현을 얻은 후 → 해당 표현을 입력으로 제공
    → 분류기 학습 → 감정 분석 작업 수행

사전 학습된 BERT에서 임베딩 추출하기

- 문장의 표현을 얻는 효율적인 방법
  - [CLS] 토큰의 표현을 문장 표현으로 사용하는 것은 항상 좋은 방식은 아님.
    적절하지 않은 경우도 있음
  - 문장의 표현을 얻는 효율적인 방법은
    → 모든 토큰의 표현을 평균화 또는 풀링하는 것

허깅페이스 트랜스포머

- 허깅페이스(Hugging Face)
  - 자연어 기술의 민주화(?)를 추구하는 조직
  - 오픈 소스 트랜스포머 라이브러리 → 자연어 처리 커뮤니티에서 많은 인기
- 허깅페이스 트랜스포머 라이브러리
  - 라이브러리 내에 100개 이상의 언어로 사전 학습된 수천 개의 모델 포함
  - 파이토치, 텐서플로 모두와 호환됨

BERT 임베딩 생성하기

- 실습

하위작업을 위한 BERT 파인 튜닝 방법

- 실습
  - 텍스트 분류
  - 질문-응답
- 내용 소개
  - 자연어 추론(NLI)
  - 개체인식(NER)

질문-응답

- **질문-응답 태스크에서는**
  - **데이터셋: 질문에 대한 응답이 포함된 단락과 함께 질문이 제공됨**
  - **태스크의 목표:주어진 질문에 대한 단락에서 답을 추출하는 것**
  - **BERT의 입력: 질문-단락 쌍**
    - BERT에 질문과 응답을 담은 단락을 입력하고
    - 단락에서 응답을 추출해야 함
  - **BERT의 출력**
    - 단락에서 응답에 해당하는 텍스트의 범위 반환

질문-응답

- **예시**

질문	면역 체계는 무엇입니까?
단락	면역 체계는 질병으로부터 보호하는 유기체 내의 다양한 생물학적 구조와 과정의 시스템입니다. 제대로 기능하려면 면역 체계가 바이러스에서 기생충에 이르기까지 병원균으로 알려진 다양한 물질을 탐지하고 유기체의 건강한 조직과 구별해야 합니다.
응답	질병으로부터 보호하는 유기체 내의 다양한 생물학적 구조와 과정의 시스템입니다.

- **해결 방안**
  - 모델: 주어진 단락의 답을 포함하는 텍스트 범위의 시작과 끝의 인덱스를 이해해야 함

면역 체계는 **질병으로부터 보호하는 유기체 내의 다양한 생물학적 구조와 과정의 시스템입니다**. 제대로 기능하려면 면역 체계가 바이러스에서 기생충에 이르기까지 병원균으로 알려진 다양한 물질을 탐지하고 유기체의 건강한 조직과 구별해야 합니다.

질문-응답

- **고민해야 할 것은?**
  - 답을 포함하는 텍스트 범위의 시작과 끝의 인덱스를 어떻게 찾을까?
  - 단락 내 답의 시작과 끝 토큰(단어)의 확률을 구하면 쉽게 답을 추출할 수 있을 것인가?
  - 어떻게 하면 이렇게 동작하게 할 수 있는가?

질문-응답

- **방안**
  - 시작 벡터 S와 끝 벡터 E라는 2개의 벡터를 사용. 시작 및 끝 벡터의 값은 학습이 되는 값
  - 단락 내 각 토큰이 응답의 시작 토큰이 될 확률 계산

$$P_i = \frac{e^{S \cdot R_i}}{\sum_j e^{S \cdot R_i}}$$

- 각 토큰 i에 대하여 R_i 토큰 표현 벡터와 시작 벡터 S 간의 내적을 계산한다.
- 내적 $S \cdot R_i$에 $softmax$ 함수를 적용하고 확률 획득

  - 다음으로 시작 토큰이 될 확률이 높은 토큰의 인덱스를 선택하여 시작 인덱스 계산
  - 유사한 방식으로 단락의 각 토큰이 응답의 끝 토큰이 될 확률을 계산

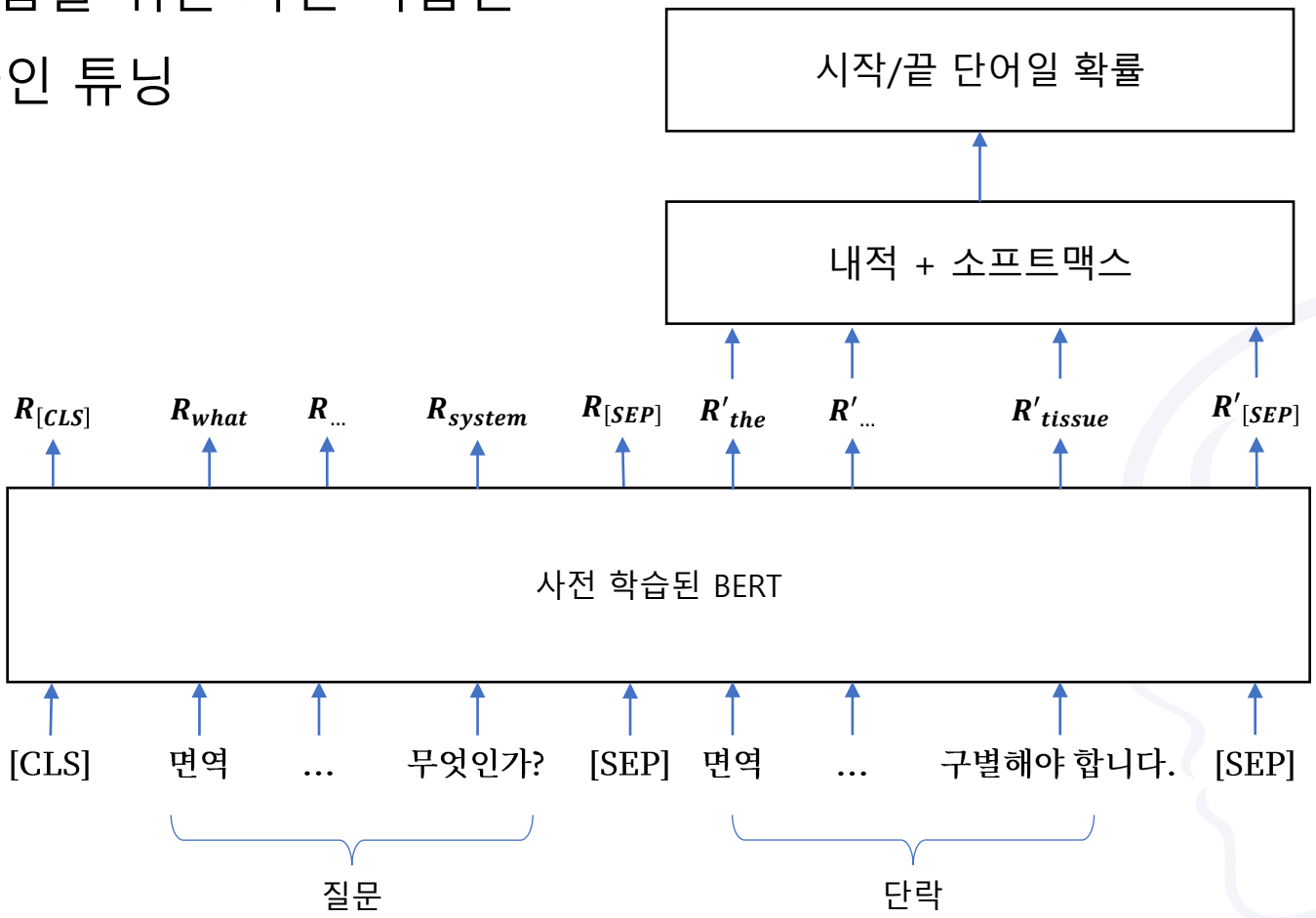
$$P_i = \frac{e^{E \cdot R_i}}{\sum_j e^{E \cdot R_i}}$$

- 각 토큰 i에 대하여 R_i 토큰 표현 벡터와 끝 벡터 E 간의 내적을 계산한다.
- 내적 $E \cdot R_i$에 $softmax$ 함수를 적용하고 확률 획득

  - 다음으로 끝 토큰이 될 확률이 높은 토큰의 인덱스를 선택하여 끝 인덱스 계산

질문-응답

- 질문-응답을 위한 사전 학습된 BERT 파인 튜닝



자연어 추론

- **자연어 추론(Natural Language Inference, NLI)**
  - **NLI에서 모델은**
    - 가정이
    - 주어진 전제에 대하여 참인지 거짓인지 중립인지 여부를
    - 결정하는 태스크
  - **모델의 목표**
    - 주어진 문장 쌍(전제, 가설 쌍)에 대해서 참, 거짓, 중립 여부를 결정하는 것

자연어 추론

- **샘플 NLI 데이터**

전제	가설	레이블
그는 놀고 있다	그는 자고 있다	거짓
여러 남성이 플레이하는 축구 게임	몇몇 남자들이 스포츠를 하고 있다	참
웃고 있는 노인과 청년	두 남자가 바닥에서 놀고 있는 강아지들을 보고 웃고 있다	중립

전제: He is playing (그는 놀고 있다).

가설: He is sleeping (그는 자고 있다).

tokens=[[CLS], He is playing [SEP], He is sleeping , [SEP]]

- 사전 학습된 BERT에 토큰 입력, 각 토큰의 임베딩 획득
- [CLS] 토큰의 $R_{[CLS]}$ 표현 벡터를 가져와 분류기(FFN, Softmax)에 입력
- 분류기는 문장이 참, 거짓, 중립일 확률 반환

개체명 인식

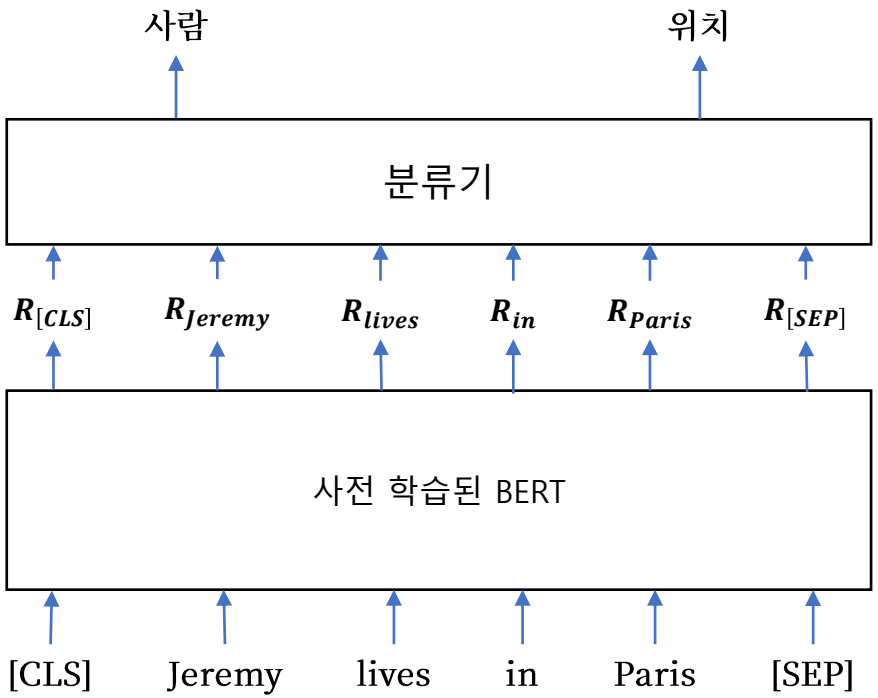
- 개체명 인식 (Named Entity Recognition, NER)
  - 목표: 개체명을 미리 정의된 범주로 분류하는 것
    - 예: Jeremy lives in Paris (제레미는 파리에 산다) → "제레미: 사람, 파리: 위치"로 분류
- BERT 모델의 파인 튜닝으로 NER 수행하기
  - 문장 토큰화 → [CLS][SEP] 추가

    → 사전 학습된 BERT 모델에 토큰 입력 → 모든 토큰의 표현 벡터 획득

    → 토큰 표현을 분류기(FFN+Softmax 함수)에 입력

    → 분류기가 개체명이 속한 범주를 반환

개체명 인식


사람
위치
분류기
$R_{[CLS]}$
R_{Jeremy}
R_{lives}
R_{in}
R_{Paris}
$R_{[SEP]}$
사전 학습된 BERT
[CLS]
Jeremy
lives
in
Paris
[SEP]

BERT의 특징

- BERT의 세가지 특징
  - 문맥에 의존한 단어 벡터 표현을 만들 수 있게 되었다.
  - 자연어 처리 작업에서 파인 튜닝이 가능해 졌다.
  - Attention에 의한 설명과 시각화가 간편해 졌다.

BERT의 특징

- **문맥에 의존한 단어 벡터 표현을 만들 수 있게 되었다.**
  - **어떤 언어라도 단어의 의미가 단 하나인 경우는 적음**
    - (예) bank: 은행, 강변 이라는 의미가 있음
  - **다양한 의미를 가진 각 단어들은 문맥에 따라 단어의 의미가 바뀜**
  - **BERT는 문맥에 맞는 단어의 벡터 표현이 가능함**

BERT의 특징

- **BERT는 12단 Transformer를 사용**
  - Embedding 모듈에서 단어ID를 단어 벡터로 변환할 때는 은행의 bank와 강변의 bank는 동일한 길이(768)의 단어 벡터
  - 12단의 Transformer를 거치는 동안 단어 bank의 위치에 있는 특징량 벡터는 변화함
  - 변화 결과, 12단 째의 출력인 단어, bank의 위치에 있는 특징량 벡터는 최종적으로 은행 bank와 강변 bank가 서로 다른 벡터가 됨
  - 여기서 말하는 특징량 벡터란, 사전 학습의 MLM이 풀어놓은 특징량 벡터
  - 문장 중의 단어 bank와 그 주변 단어와의 관계성을 바탕으로 하여 Transformer의 Self-Attention 처리로 작성됨
  - 동일한 단어도 주변 단어와의 관계성에 따라 문맥에 맞는 단어 벡터가 생성됨

BERT의 특징

- **자연어 처리 작업에서 파인 튜닝이 가능해 졌다.**
  - **BERT를 기반으로 다양한 자연어 처리를 수행하려면**
    - 두 언어 작업에서 사전 학습한 가중치 파라미터를 BERT 모델의 가중치로 설정
    - BERT 모델 구조 그림에서 나타낸 (seq_len x hidden)=(512x768) 텐서와 (hidden)=(768)의 두 텐서를 출력
    - 두 텐서를 실행하고 싶은 자연어 처리 작업에 맞춘 어댑터 모듈에 투입
    - 작업에 따른 출력 획득
    - (예) 긍정적/부정적 감정 분석의 경우, 어댑터 모듈로 하나의 전결합층을 추가하는 것 만으로 문자의 판정이 가능해짐

BERT의 특징

- 모델을 학습할 때, 기반이 되는 BERT와 어댑터 모듈의 전결합층 양쪽 모두를 파인 튜닝으로 학습
- BERT의 출력에 어댑터 모듈을 연결하여 다양한 자연어 처리 작업 수행 가능
- Object Detection을 위한 SSD 모델, 자세 추정을 위한 OpenPose 모델에서 사용된 기반 네트워크인 VGG와 같은 역할을 BERT가 수행
- 적은 문서 데이터로도 성능 좋은 모델의 작성이 가능함
- 자연어 처리 작업도 화상 작업처럼 전이학습 및 파인 튜닝을 적용할 수 있게 된 것이 BERT가 주목받은 요인의 하나임

BERT의 특징

- **BERT는 어떻게 화상 작업의 기본 모델인 VGG와 같은 전이학습 및 파인 튜닝의 기반 역할을 수행할 수 있을까?**
  - VGG 모델과 같이 화상 처리에서 화상 분류가 가능한 네트워크는 물체 감지나 시맨틱 분할에도 유효함
  - BERT도 사전작업 MLM을 풀 수 있는 **단어를 문맥에 맞는 특징량 벡터로 변환할 수 있는 능력**이 단어의 의미를 정확하게 파악할 수 있게 함
  - 사전작업 NSP로 **문장이 의미 있게 연결되었는지 여부를 판정할 수 있는 능력**이 문장의 의미를 이해할 수 있게 함
  - 단어와 문장의 의미를 이해할 수 있도록 사전학습을 하고 있으므로 자연어 처리 작업인 감정 분석 등에도 응용이 가능해짐

BERT의 특징

- **선구적인 범용 언어 모델의 사례를 만듦**
  - 단어와 문장의 의미를 제대로 파악해야 하는 사전 작업의 수행
  - 사전 작업으로 학습한 가중치를 기반으로, 어댑터를 자연어 처리 작업에 맞게 교체하여 파인 튜닝을 수행

→ 이러한 처리의 흐름이 자연어 처리에서의 하나의 표준이 될 것으로 기대됨

BERT의 특징

- Attention에 의한 설명과 시각화가 간편해 졌다.
  - Attention: 결과에 영향을 준 단어의 위치정보
  - Attention을 시각화 함으로써 인간이 추론 결과를 설명하기가 쉬워짐

참고 도서

- 구글 BERT의 정석 (수다르산 라비찬디란 저 / 전희원, 정승환, 김형준 역 | 한빛미디어)
- 텐서플로 2와 머신러닝으로 시작하는 자연어 처리 (전창욱, 최태균, 조중현, 신성진 저 | 위키북스)