Week13_예습과제_팽소원

BERT

BERT(Bidirectional Encoder Representations from Transformers)

: 2018년 구글에서 발표한 언어 모델로 트랜스포머 기반 양방향 인코더를 사용하는 자연어 처리 모델

- 양방향 인코더 : 입력 시퀀스를 양쪽 방향에서 처리하여 이전과 이후의 단어를 모두 참조하면서 단어의 의미와 문맥 파악

  → 단방향 모델들보다 더 정확하게 문맥 파악 가능, 다양한 자연어 처리 작업에서 높은 성능

- BERT는 대규모 데이터를 사용해 사전 학습되어 있어, 전이 학습에 주로 활용
- 트랜스포머 인코더 모듈만 사용해 입력 문장 처리

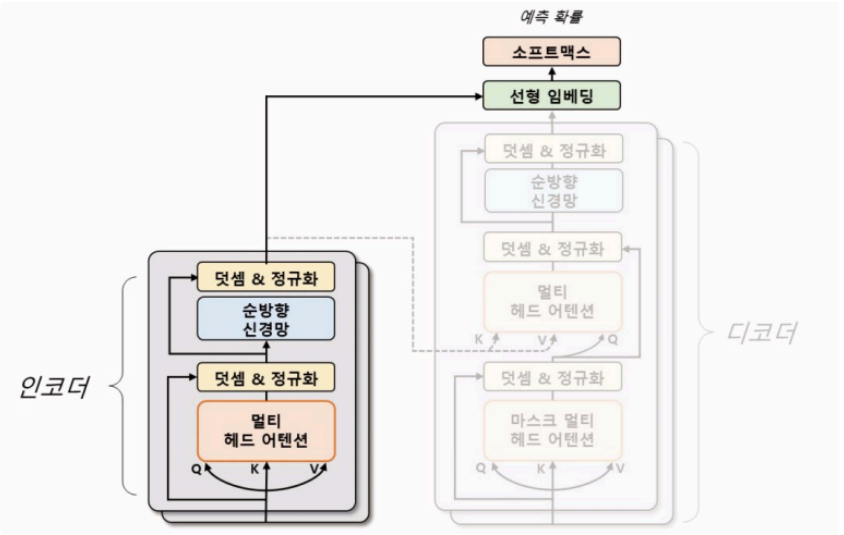

그림 7.13 트랜스포머와 BERT 모델 비교

사전 학습 방법

BERT는 사전 학습을 위해 **마스킹된 언어 모델링(MLM)**과 **다음 문장 예측 방법(NSP)**을 사용한다

- **마스킹된 언어 모델링(MLM)** : 입력 문장에서 임의로 일부 단어를 마스킹하고 해당 단어를 예측하는 방식

ex) “I’m learning PyTorch”라는 문장에서 ‘learning’을 마스킹하면 “I’m [MASK] Pytorch”가 되어 BERT가 [MASK]에 들어갈 단를 양방향 문맥 정보를 참고해 예측

⇒ 입력 문장에서 누락된 단어를 추론하는 능력을 갖게 되며 이를 통해 BERT는 문장 전체 의미를 이해하는 능력이 향상됨

- **다음 문장 예측(NSP)** : 두 개의 문장이 주어졌을 때, 두 번째 문장이 첫 번째 문장의 다음에 오는 문장인지 여부를 판단하는 작업

ex) “I’m learning PyTorch”, “PyTorch is a machine learning library”라는 두 문장이 주어지면 BERT는 이 두 문장이 연속적인 관계인지 아닌지 예측

⇒ BERT는 두 문장 간의 관계를 학습하고, 문장 간의 의미적인 유사성을 파악

- **BERT에서 사용하는 특수 토큰**

[CLS] 토큰 : 입력 문장의 시작 부분에 추가되는 토큰

→ 입력 문장이 어떤 유형의 문장인지 미리 파악 가능 (EX. 긍정 OR 부정)

[SEP] 토큰 : 입력 문장 내에서 두 개 이상의 문장을 구분하기 위해 사용되는 토큰

→ 입력 문장을 두 개의 독립적인 문장으로 인식하고, 각각의 문장에 대한 정보를 정확하게 파악 가능

[MASK] 토큰 : 입력 문장 내에서 임의로 선택된 단어를 가리키는 특별한 토큰

→ 주어진 문장에서 일부 단어를 가려 모델의 학습과 예측에 활용

→ 텍스트 토큰 중 15% 에 해당하는 단어를 대상으로 마스킹 수행

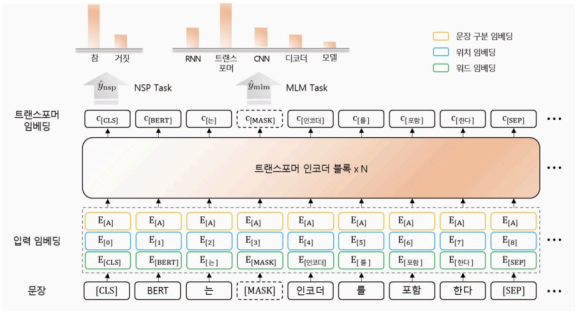

그림 7.14 BERT 모델의 입력 임베딩과 손실 함수

입력 문장 : “BERT는 트랜스포머 인코더를 포함한다”

⇒ [CLS] BERT는 트랜스포머 인코더를 포함한다 [SEP]

BART

BART(Bidirectional Auto-Regressive Transformer)

: 2019년 FAIR 연구소에서 발표한 트랜스포머 기반 모델

- BERT의 인코더와 GPT의 디코더를 결합한 시퀀스-시퀀스 구조로 노이즈 제거 오토인코더로 사전학습

- 오토인코더 사전 학습 방법 : 입력 데이터에 잡음 추가, 잡음이 없는 원본 데이터를 복원하도록 학습
- BART는 사전학습시 BERT의 인코더와 GPT의 디코더가 학습하는 방법을 일반화하여 학습 진행

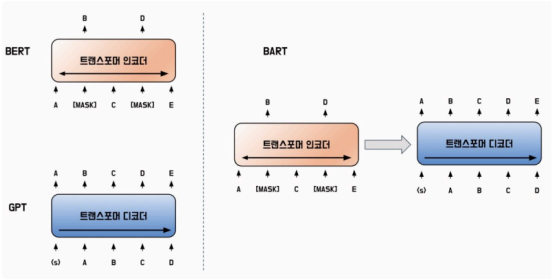

그림 7.15 BART 사전 학습 시각화

노이즈가 추가된 텍스트를 인코더에 입력하고 원본 텍스트를 디코더에 입력해 디코더가 원본 텍스트를 생성할 수 있게 학습

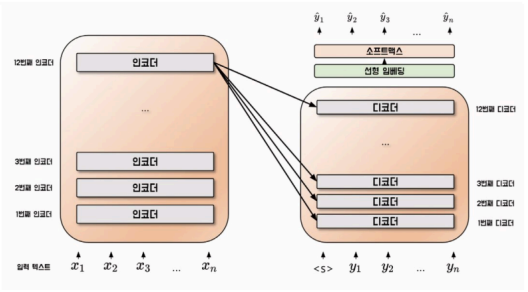

그림 7.16 BART 모델 구조

BART : 인코더와 디코더를 모두 사용, 트랜스포머와 유사한 구조

트랜스포머 - 인코더의 모든 계층과 디코더의 모든 게층 사이의 어텐션 연산을 수행

BART - 인코더의 마지막 계층과 디코더의 각 계층 사이에*만* 어텐션 연산 수행

사전 학습 방법

BART의 인코더 : BERT의 마스킹된 언어 모델링 + 그 외 다양한 노이즈 기법 사용

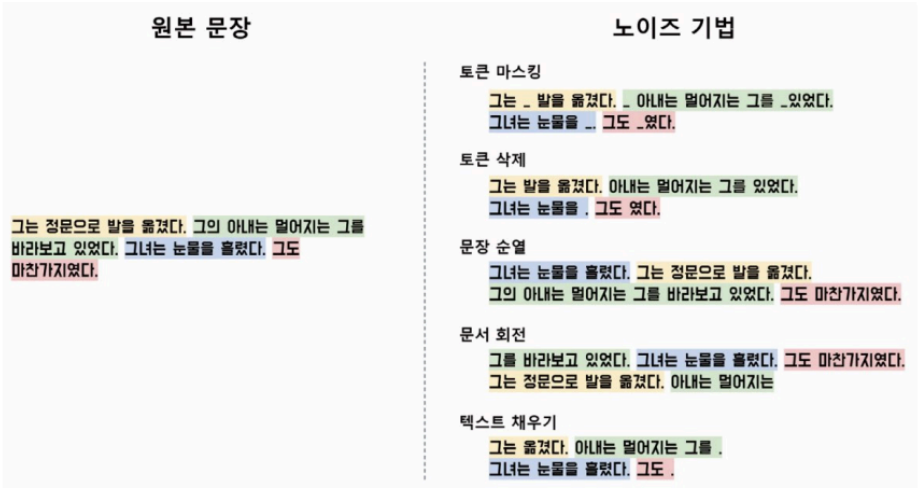

그림 7.17 BART 노이즈 기법

토큰 마스킹 : BERT에서 사용한 MLM과 동일한 기법으로, 입력 문장의 일부 토큰을 마스크 토큰으로 치환하는 방법

토큰 삭제 : 입력 문장의 일부 토큰을 치환하는 것이 아닌 삭제하는 방법

문장 순열 : 마침표를 기준으로 문장을 나눈 뒤, 문장의 순서를 섞는 방법

문서 회전 : 임의의 토큰으로 문서가 시작하도록 하며, 모델은 문서의 원래의 시작 토큰을 맞춰야 함

텍스트 채우기 : 몇 개의 토큰을 하나의 구간으로 묶고 일부 구간을 마스크 토큰으로 대체

미세 조정 방법

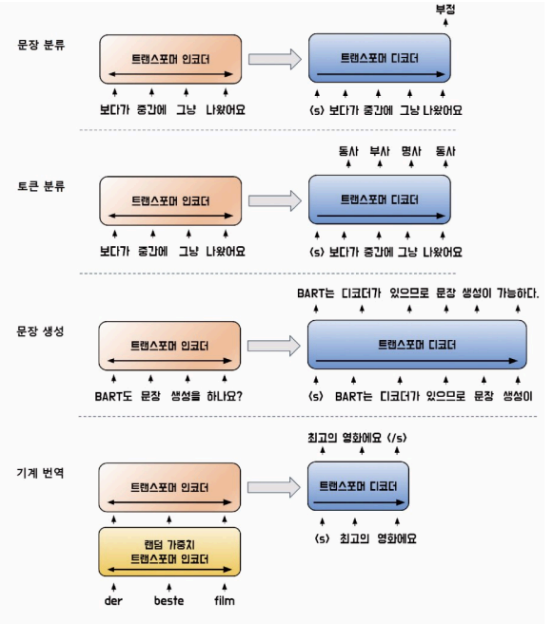

그림 7.18 BART 미세 조정 구조

문장 분류 작업 - 입력 문장을 인코더와 디코더에 동일하게 입력하고 디코더의 마지막 토큰 은닉 상태를 선형 분류기의 입력값으로 사용

토큰 분류 작업 - BART는 입력 문장을 인코더와 디코더에 동일하게 입력, 디코더의 각 시점별 마지막 은닉 상태를 토큰 분류기의 입력값으로 사용

BART는 트랜스포머 디코더를 사용하기 때문에 BERT가 해결하지 못했던 문장 생성 작업 (ex. 추상적 질의응답, 문장 요약) 수행 가능

모델 실습

BART 모델을 학습한 후, 평가할 때는 문장 생성 기법에서 자주 사용되는 루지(ROUGE) 점수를 사용

ROUGE 점수 : 생성된 요약문과 정답 요약문이 얼마나 유사한지 평가하기 위해 토큰의 N-gram 정밀도와 재현율을 이용해 평가하는 지표

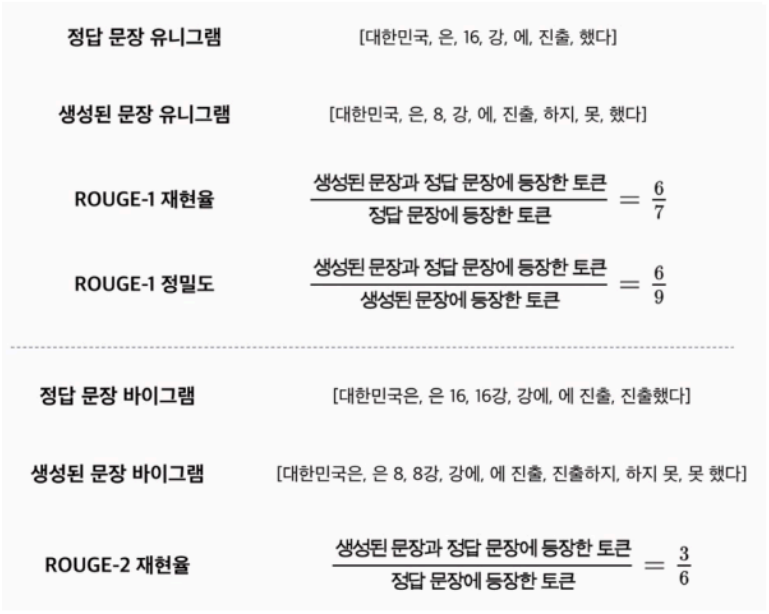

그림 7.19 루지 점수 계산 방법

ELECTRA

ELECTRA : 2020년 구글에서 발표한 트랜스포머 기반의 모델

- 입력을 마스킹하는 대신 생성자와 판별자를 사용해 사전학습 수행
- GAN과 유사한 방법으로 학습 수행
- 모델은 실제 데이터와 비슷하게 토큰을 생성해 다른 토큰으로 대체하고 판별 모델이 생성 모델이 만든 데이터와 실제 데이터를 입력 받아 어떤 데이터가 실제인지, 생성된 것인지 구분
- BERT 보다 매개변수가 더 작아 빠르고 더 적은 메모리 사용

사전 학습 방법

ELECTRA의 생성자 모델과 판별자 모델은 모두 트랜스포머 인코더 구조를 따름

생성자 모델 - 입력 문장의 일부 토큰을 마스크 처리하고 마스크 처리된 토큰이 원래 어떤 토큰이었는지 예측하며 학습

판별자 모델 - 각 입력 토큰이 원본 문장의 토큰인지 생성자 모델로 인해 바뀐 토큰인지 맞히며 학습 (= RTD)

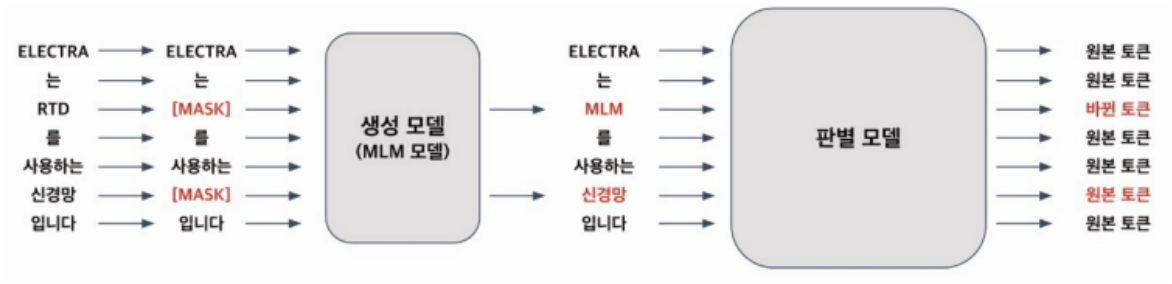

그림 7.20 ELECTRA 모델 학습 방법

T5

T5(Text-to-Text Transfer Transformer) : 2019년 구글에서 발표한 자연어 처리 분야의 딥러닝 모델로 트랜스포머 구조를 기반

- 다양한 자연어 처리 작업에서 높은 성능
- 입력과 출력을 모두 토큰 시퀀스로 처리하는 text-text 구조
- 입력과 출력의 형태를 자유롭게 다룰 수 있으며, 모델 구조상 유연성과 확장성이 뛰어남

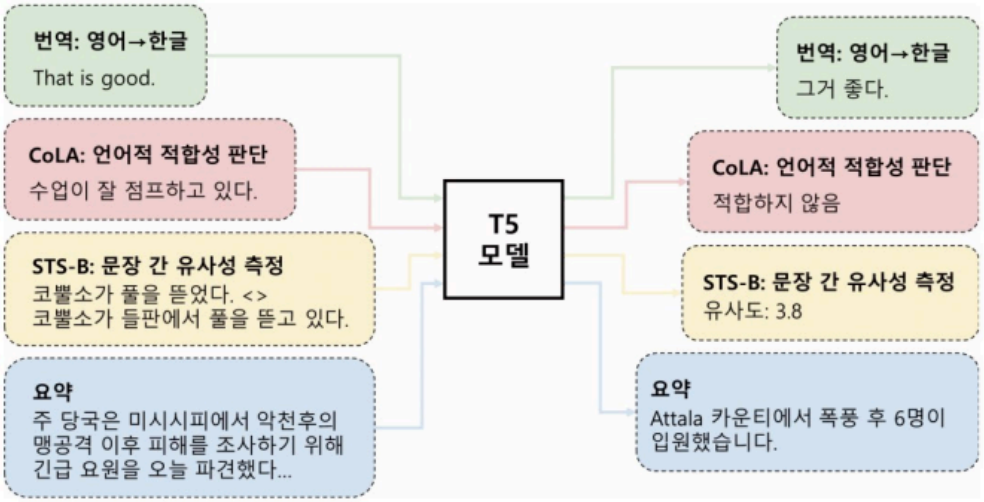

그림 7.21 T5 모델 학습 유형

- 사전 학습 후 미세 조정 간계에서 해당 작업의 데이터를 이용해 모델을 조정해 최적의 성능 얻을 수 있음
- C4 데이터세트를 활용해 다양한 자연어 처리 작업 수행 가능
- 사전 학습 방식은 비지도 학습 방식으로 입력 문장의 일부 구간 마스킹해 입력 시퀀스를 처리하며, 출력 시퀀스는 실제 마스킹된 토큰과 마스크 토큰의 연결로 구성
- 유일한 마스크 토큰을 의미하는 **센티널 토큰** 사용