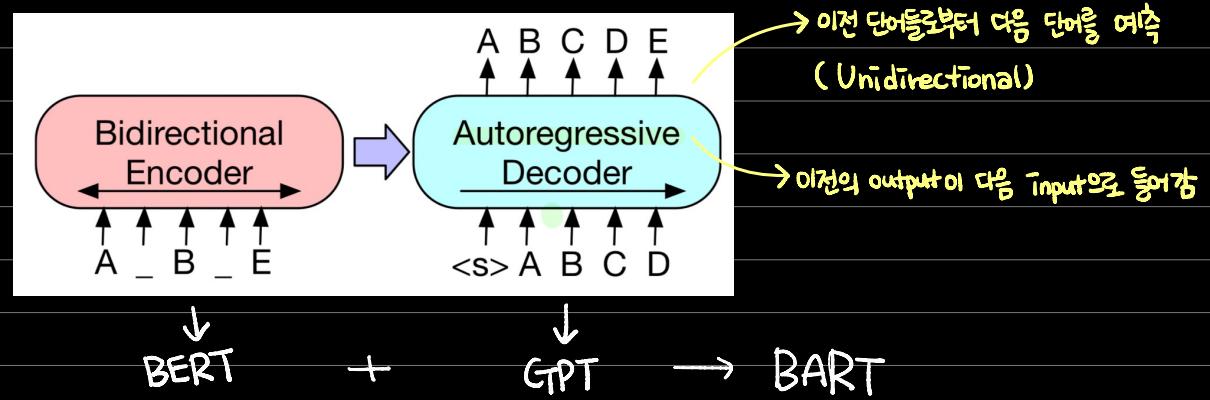


BART의 구조



Input : corrupted document

(BERT의 masked language model)

→ MLM: 문장 내 일부 단어를 가리고 문맥을 통해 어떤 단어가 들어갈지 맞힐

task : input data를 original document로 복원시키는 것

seq 2 seq

NLP에서 시퀀스를 출력시키는 모델

GRU를 인코더와 디코더의 모델로 사용

Self-supervised 방법론들은 광범위한 NLP task에서 괄목할 성과를 거둠

→ Word2Vec, ELMo, BERT, SpanBERT, XLNet, Roberta

가장 성공적인 접근은 Close task에서 영감을 받은 MLM의 변형

그러나 기존의 방법론들은 특정 End Task 형태에 집착하여 활용성이 떨어짐

→ XLNet, SpanBERT, UniLM

BART vs. BERT

BERT

NSP ←
주변 문장 대상에 두 가지
문장을 제시하여 훈련에는
문장의 문맥상 의미라는 문장
인지 막히게 하는 학습법

여러 단어를 동시에 예측할
때 해당 단어 간 상관관계를
고려하지 X

트랜스포머의 인코더로만 모델을 구축

MLM pre-train 목적함수 + Next

Sentence Prediction pre-train 목적함수로 학습

denoising autoencoding task 해설

Independence Assumption

→ 모든 masked token들은 독립적으로 재구축

pre-train & Finetune discrepancy

→ 파인튜닝 때 masked token이 들어가지 않아 발생

→ XLNet이 PLM을 제거하여 Autoregressive한

LM로 해결하려고 함

BART

트랜스포머의 인코더-디코더로, S2S 모델을 기본으로 함

→ 레이어 별로 크로스 어텐션을 디코더에서 추가로 수행

→ word prediction 전 추가적인 FFN 필요 X

GPT와 같은 Autoregressive decoder 사용

→ ReLU 대신 GELU 사용

↳ BERT의 두 가지 단점 보완

XLNet, SpanBERT, UniLM 등 기존의 MLM 변형체

채용

Noise flexibility: origin text에 임의의 변화를

적용할 수 있음

Pre-training BART

(다섯가지 노이즈 task)

① Token Masking

BERT의 MLM과 동일

랜덤 토큰들을 Sampling → [mask] token으로 대체

80% masking, 10% 임의의 다른 토큰으로 변형

② Token Deletion

입력에서 토큰을 임의로 제거하는 task

masking 과는 다르게 어느 자리에서 토큰이 유지되었는지를 결정해야 함

③ Text Infilling

다섯가지 task 중에서 가장 성능이 좋음

→ 긴 문장의 토큰

→ default 값이 3인 표마송 분포에서 Span length를 추출한 길이만큼의 text span sampling 토큰을 단일 마스크 토큰으로 대체

모델에게 span으로부터 얼마나 많은 토큰이 유지되었는지를 예측하도록 학습시킴

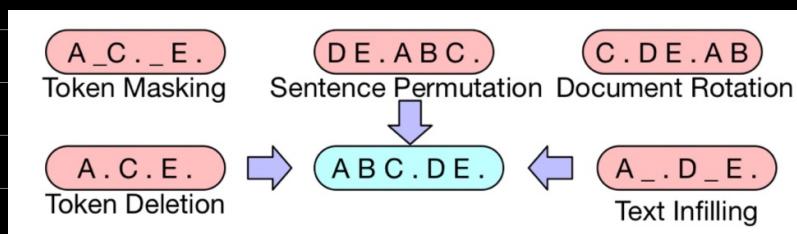
④ Sentence Permutation

Full stop 등을 기준으로 Document를 sentence로 나누고 이 문장들을 임의의 순서로 섞어준다

⑤ Document Rotation

임의로 token을 선택하고 Document를 해당 Token으로 시작하도록 Rotate

문장의 시작이 어디인지 학습할 수 있게 해준다



Encoder

(bsz, seq_len)
Original document → Corruption (masking 등) → embedding
→ encoding → output

Decoder

(bsz, seq_len, d)
Shift Input → embedding → Auto-regressive decoder → output
 $(bsz, seq_len, 1|V)$
(document reconstruction)

Loss

Decoder의 output과 original document 사이의 cross-entropy

$$\text{loss}(\vec{x}, \text{class}) = -\log \left(\frac{\exp(x[\text{class}])}{\sum_j \exp(x[j])} \right) = -x[\text{class}] + \log \left(\sum_j \exp(x[j]) \right)$$

Fine-tuning

Sequence Generation Tasks

Auto-regressive한 decoder가 있기 때문에 추가적인 레이어를 둘 필요 X
인코더에 유망한 입력을 넣어주면 디코더에서 학습적 생성

Metric

① ROUGE (Recall Oriented Understudy for Gisting Evaluation)

생성한 요약과 정답지를 대조하여 성능 점수를 계산

$$\text{ROUGE precision} = \frac{\text{Number of overlapped words}}{\text{Total words in system summary}}$$

$$\text{ROUGE recall} = \frac{\text{Number of overlapped words}}{\text{Total words in reference summary}}$$

$$f1\text{ score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

n을 결정하기 위해서
ROUGE f1 score를 계산해야함

ROUGE - N : 들어가는 단어의 크기를 n-gram을 적용해 성능을 측정

ROUGE - L : LCS를 사용해 최장 길이로 대체되는 문장열을 측정
Longest Common Subsequence

한계 : 같은 뜻을 가진 다양한 단어들을 수용하지 X (이음동의어)

semantic < syntactic

reference 여러 개에 대한 ROUGE Score의 평균을 사용하는 방식으로 상쇄할 수 있음