BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension

Hwang Hyeon Tae

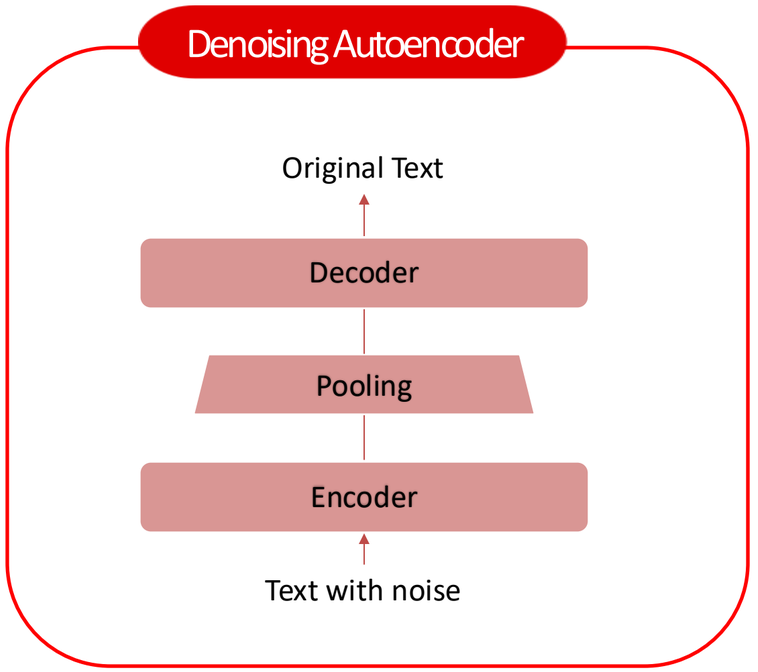
01

Introduction

Self-supervised Method

- NLP Task 전반에 좋은 성과
  - 가장 성공적인 방법은 Denoising Autoencoder
    - Masked Language Model의 변형
    - But, 일반적으로 특정 유형의 Task에 초점을 맞추기 때문에 적용성이 제한되는 단점

- 목표
  - 특정 Task를 위한 모델이 아닌, 광범위한 Task에 적용가능한 모델 제작의 필요성

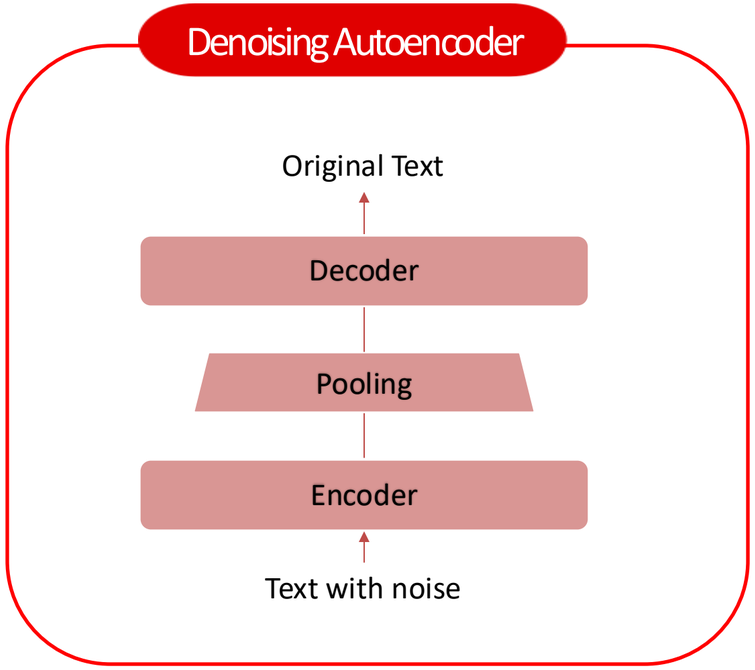


What is BART?

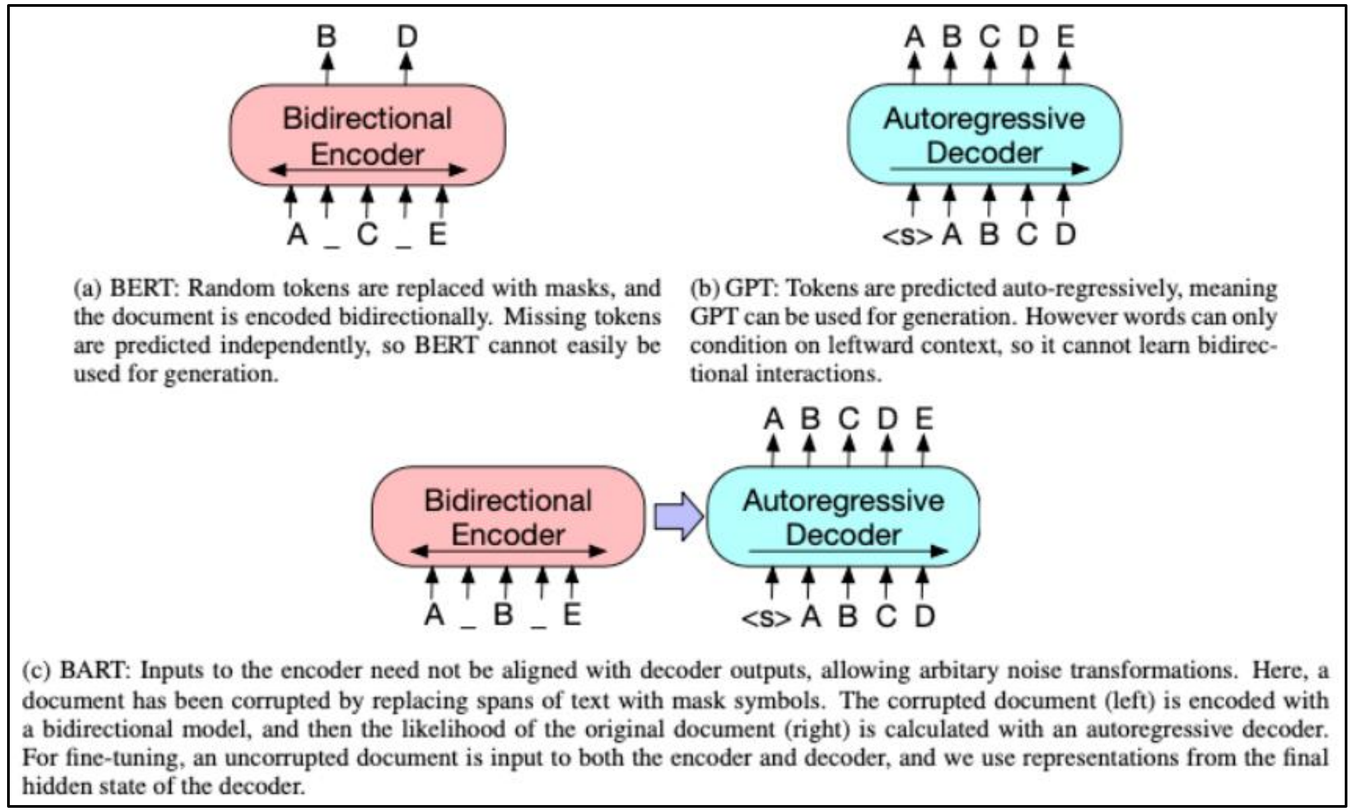
- BART
  - Bidirectional과 Auto-Regressive Transformer를 결합한 모델
    - 즉, Sequence-to-Sequence Model로 구축된 Denoising Autoencoder

- Pre-training Step
  1) 텍스트가 임의의 noising function에 의해 손상되고
  2) Seq-to-Seq 모델이 original text를 재구성

- In Paper,
  - 여러 Noising Approach를 소개 및 성능 평가



What is BART?



(a) BERT: Random tokens are replaced with masks, and the document is encoded bidirectionally. Missing tokens are predicted independently, so BERT cannot easily be used for generation.

(b) GPT: Tokens are predicted auto-regressively, meaning GPT can be used for generation. However words can only condition on leftward context, so it cannot learn bidirectional interactions.

(c) BART: Inputs to the encoder need not be aligned with decoder outputs, allowing arbitary noise transformations. Here, a document has been corrupted by replacing spans of text with mask symbols. The corrupted document (left) is encoded with a bidirectional model, and then the likelihood of the original document (right) is calculated with an autoregressive decoder. For fine-tuning, an uncorrupted document is input to both the encoder and decoder, and we use representations from the final hidden state of the decoder.

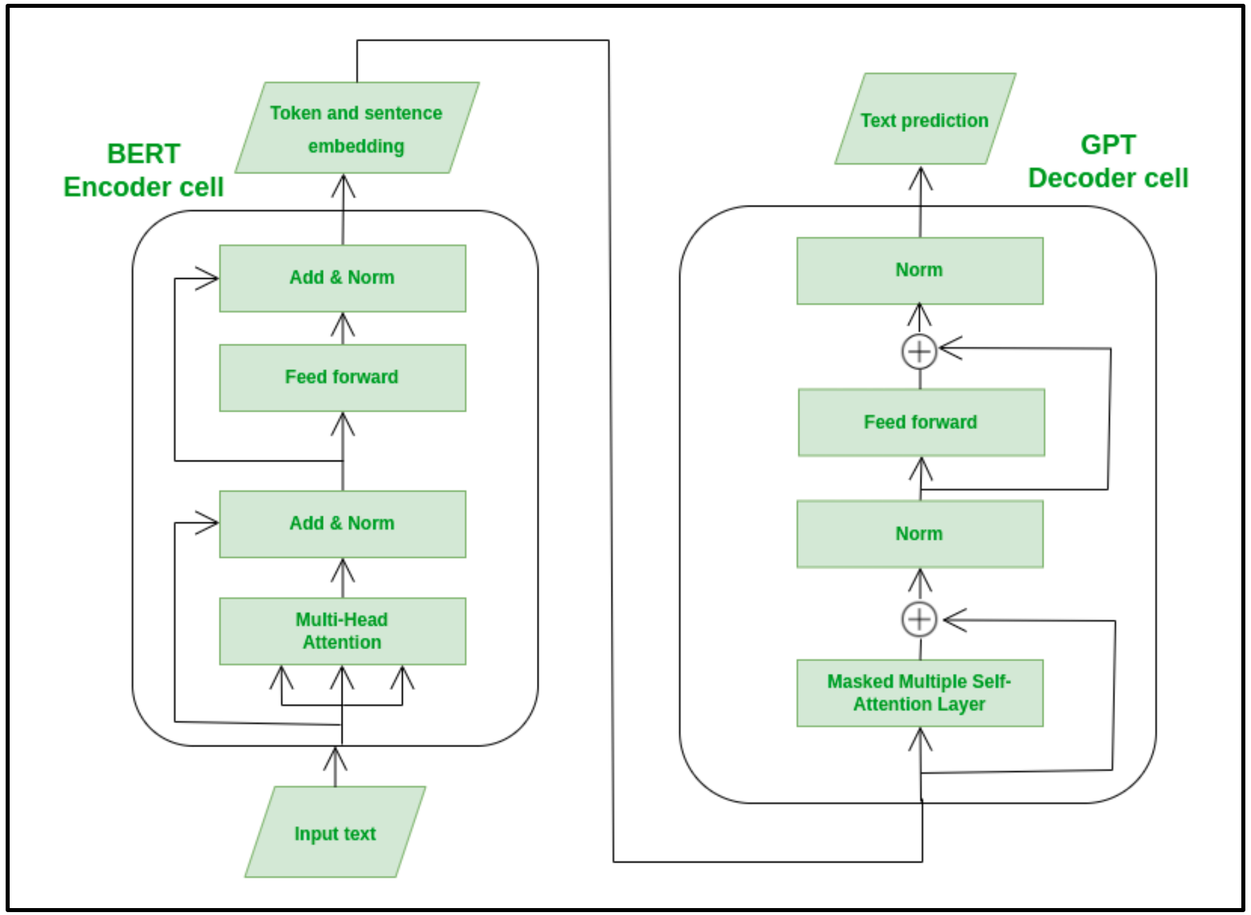
- BERT
  - 토큰을 무작위로 masking하고 문서가 양방향으로 인코딩된다
    - 누락된 토큰은 독립적으로 예측되므로 BERT를 생성에 쉽게 사용 X
- GPT
  - 생성에 사용될 수 있지만 left-to-right Autoregressive decoder기 때문에 bidirectional interaction을 학습 X
- BART
  - 손상된 문서를 bidirectional model로 encoding
  - 원본 문서의 likelihood를 decoder로 계산

02

Pretraining BART

BART

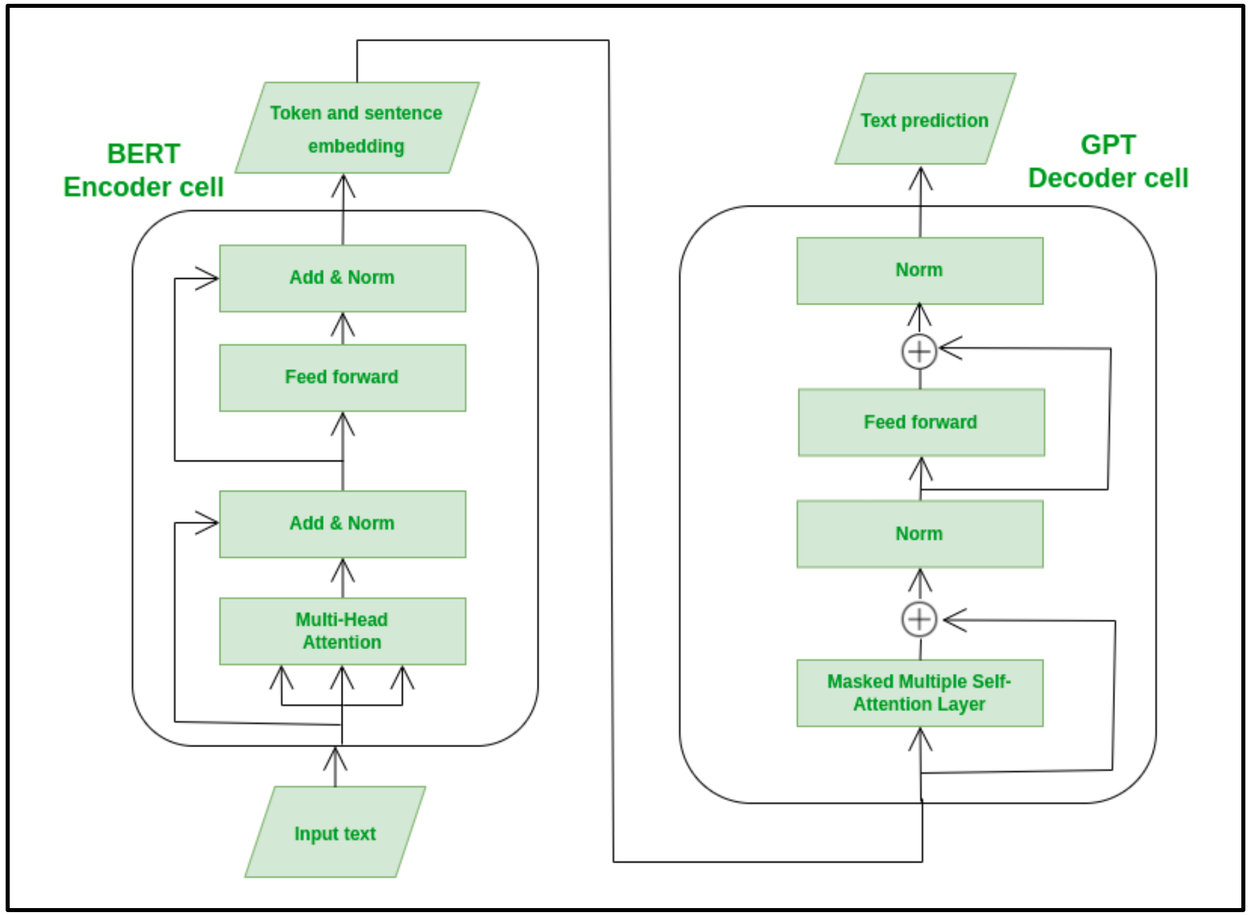
- Architecture



- Activation Function
  - ReLU -> GeLU(GPT와 동일)
- Base Model
  - Encoder Layer : 6
  - Decoder Layer : 6
- Large Model
  - Encoder Layer : 12
  - Decoder Layer : 12

BART

- Architecture



- BART vs BERT
  1) Decoder의 각 Layer는 Encoder의 마지막 hidden layer에 대한 cross-attention 수행
  2) BERT는 Text Prediction 전에 추가적인 FeedForward Network를 사용하지만 BART에는 x
  3) 전체적으로 BART는 동일한 크기의 BERT 보다 약 10% 더 많은 parameter를 포함

Noising Approach

- 5 가지의 Noising Approach를 실험에 사용
  - Token Masking
    - Random token이 샘플링되어 [MASK]로 대체
  - Token Deletion
    - Random token이 input으로부터 삭제
    - 모델이 어떤 위치에 input이 없는지 결정

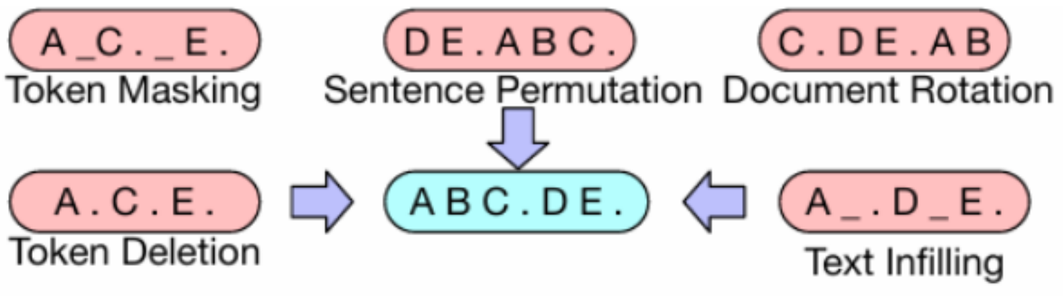


Figure 2: Transformations for noising the input that we experiment with. These transformations can be composed.

Noising Approach

- 5 가지의 Noising Approach를 실험에 사용
  - Text Infilling
    - 여러 text span을 샘플링 -> span은 단일 [MASK] token으로 대체(span의 길이는 포아송 분포에서 λ = 3)
    - ex) original text : "나는 오늘 아침에 학교에 갔다"
      - ex1) 나는 오늘 아침에 [MASK] -> 1개 span
      - ex2) 나는 오늘 [MASK] 학교에 [MASK] -> 2개 span

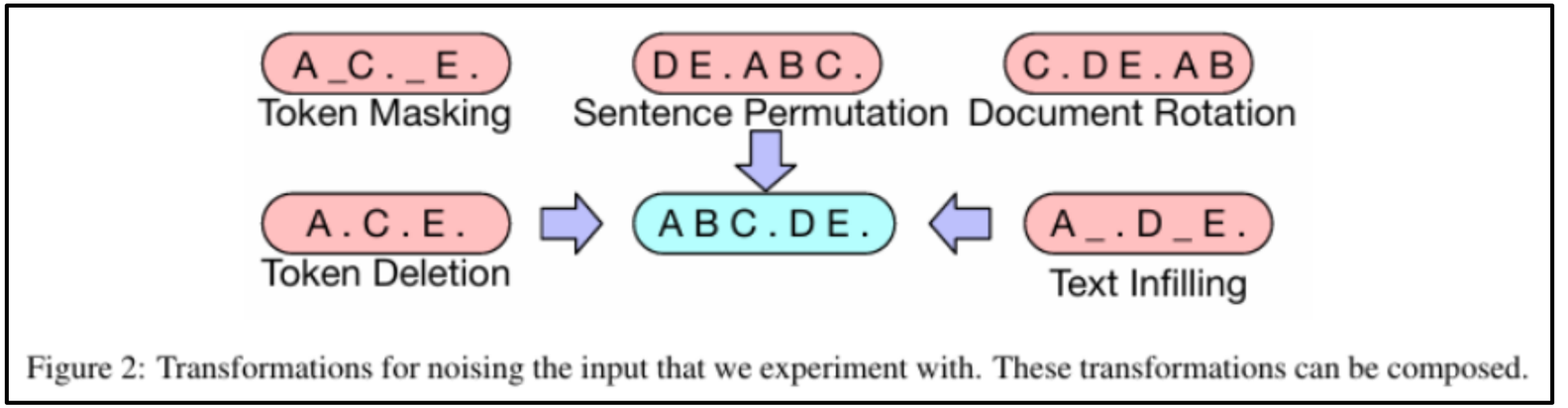


Figure 2: Transformations for noising the input that we experiment with. These transformations can be composed.

Noising Approach

- 5 가지의 Noising Approach를 실험에 사용
  - Sentence Permutation
    - Document를 마침표를 기준으로 문장으로 나눈 후, 무작위 순서로 섞는다.
  - Document Rotation
    - 토큰을 무작위로 선택해 Document가 해당 토큰으로 시작되도록 회전
      - 모델이 문서의 시작을 식별하도록 훈련

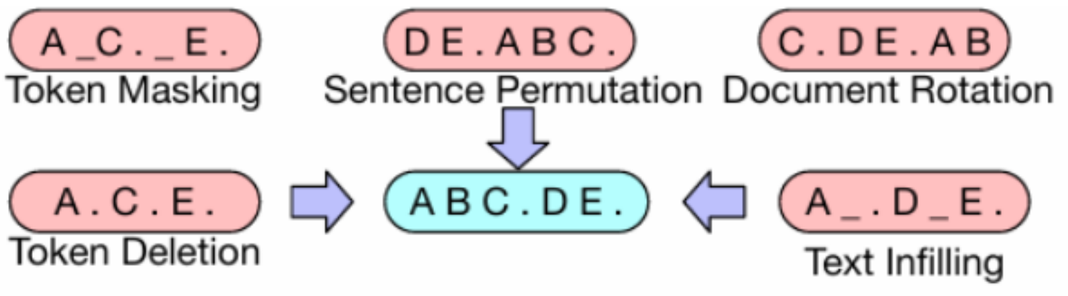


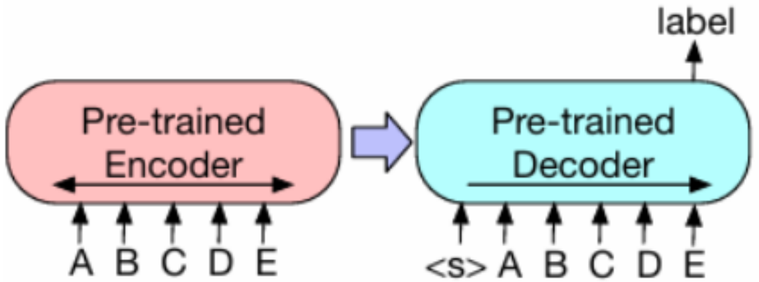
Figure 2: Transformations for noising the input that we experiment with. These transformations can be composed.

03

Comparing Pre-training Objectives

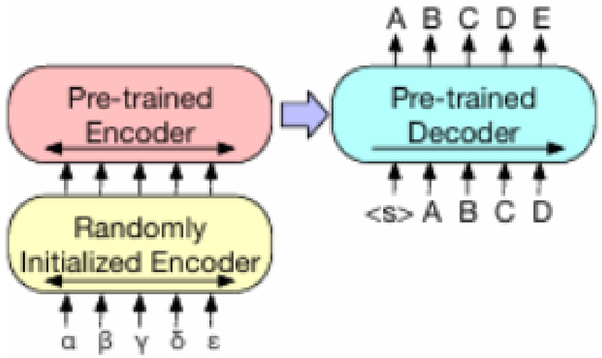
Fine-tuning BART

- BART의 활용
  - Sequence Classification Tasks
    - Final Decoder layer의 hidden state가 새로운 Multi-class Linear classifier에 입력
  - Token Classification Tasks
    - Final Decoder layer의 hidden state을 각 단어의 표현으로 사용
      - 토큰을 분류하는데 사용



Fine-tuning BART

- BART의 활용
  - Sequence Generation Tasks
    - BART에는 Auto-Regressive Decoder가 있기 때문에 sequence 생성 작업에 대해 finetuning 작업 진행 가능
  - Machine Translation
    - Bitext로 학습된 새로운 Encoder Parameter set을 추가해 BART모델을 : 번역을 위한 단일 Pretrained Decoder로 활용 가능



Comparison Objectives

1. Language Model
   - left-to-right Transformer Language Model 학습(GPT와 유사)
     - Cross-attention이 없는 BART Decoder와 동일
2. Permuted Language Model
   - XLNet을 기반으로 토큰의 1/6을 샘플링하고 이를 무작위 순서로 Auto-Regressively하게 생성
     - 다른 모델과의 일관성을 위해 segment 간 상대적인 positional embedding이나 attention은 구현 x
3. Masked Language Model
   - BERT : 토큰의 15%를 [MASK]로 대체하고 원래 토큰을 독립적으로 예측하도록 훈련

Comparison Objectives

4. Multitask Masked Language Model

- UniLM에서와 같이, 추가적인 self attention mask가 있는 Masked Language Model을 훈련

5. Masked Seq-to-Seq

- MASS에서 영감을 받아, token의 50%를 포함하는 범위를 mask하고 masked token을 예측하기 위해 seq-to-seq 모델 학습

Tasks

1. SQuAD
   - 위키피디아 문단에 대한 추출적 질의 응답 과제
   - 질문과 답변이 포함된 데이터셋으로, 주어진 텍스트에서 특정 질문에 대한 답을 찾는 문제
   - Ex) 텍스트 : "오늘의 날씨는 맑을 예정이고, 내일의 날씨는 비가 올 예정이다"
     질문 : "오늘의 날씨는 어떤가요?"
     정답 : 맑음
2. MNLI
   - 한 문장이 다른 문장을 수반하는지 예측하는 이중 텍스트 분류 작업
   - Ex) 전제(Premise) : 민수는 매일 아침 조깅을 한다.
     Hypothesis 1 : 민수는 아침에 운동을 한다 / Lable 1 : Entailment(포함)
     Hypothesis 2 : 민수는 아침에 절대 운동을 하지 않는다 / Label 2 : Contradiction(모순)
     Hypothesis 3 : 민수는 저녁에 조깅을 한다. / Label 3 : Neutral(중립)

3. ELI5
   - 질문 추상 질문 답변 데이터 셋
   - Ex) 질문 : 왜 우리는 잠을 자야 하나요?
     답변 : 잠을 자는 것은 우리 몸이 하루 동안 피곤해진 것을 다시 회복하기 위해서예요. 잠을 자면서 우리의 뇌는 하루 동안
     배운 것들을 정리하고, 몸은 고장난 곳을 고치고 힘을 다시 채우게 돼요. 마치 휴대폰을 충전하는 것처럼요.
     그래서 우리는 잠을 자고 나면 다시 힘이 나고, 더 잘 생각할 수 있게 되는 거예요.

Tasks

5. ConvAI2
   - 맥락과 페르소나에 따라 달라지는 대화 응답 생성 과제
   - 챗봇이 인간과 자연스럽게 대화할 수 있도록 설계되었으며, 대화의 연속성과 맥락 유지에 중점

6. CNN/DM
   - 뉴스 요약 데이터 셋

04

Results

Performance of pretraining methods varies significantly across tasks

- Pretraining의 효과가 task에 따라 달라진다.
  - Ex) Language Model은 최상의 ELI5 성능을 달성하지만 최악의 SQuAD 결과를 얻음

Model	SQuAD 1.1 F1	MNLI Acc	ELI5 PPL	XSum PPL	ConvAI2 PPL	CNN/DM PPL
BERT Base (Devlin et al., 2019)	88.5	**84.3**	-	-	-	-
Masked Language Model	90.0	83.5	24.77	7.87	12.59	7.06
Masked Seq2seq	87.0	82.1	23.40	6.80	11.43	6.19
Language Model	76.7	80.1	**21.40**	7.00	11.51	6.56
Permuted Language Model	89.1	83.7	24.03	7.69	12.23	6.96
Multitask Masked Language Model	89.2	82.4	23.73	7.50	12.39	6.74
BART Base						
w/ Token Masking	90.4	84.1	25.05	7.08	11.73	6.10
w/ Token Deletion	90.4	84.1	24.61	6.90	11.46	5.87
w/ Text Infilling	**90.8**	84.0	24.26	**6.61**	**11.05**	5.83
w/ Document Rotation	77.2	75.3	53.69	17.14	19.87	10.59
w/ Sentence Shuffling	85.4	81.5	41.87	10.93	16.67	7.89
w/ Text Infilling + Sentence Shuffling	**90.8**	83.8	24.17	6.62	11.12	**5.41**

Token masking is crucial

- Rotating Document 또는 Permuting Sentences에 기반한 Pretraining objective는 성능 Bad

- Token Deletion이나 Masking 또는 self attention mask를 사용하는 방법은 성능 Good

Model	SQuAD 1.1 F1	MNLI Acc	ELI5 PPL	XSum PPL	ConvAI2 PPL	CNN/DM PPL
BERT Base (Devlin et al., 2019)	88.5	**84.3**	-	-	-	-
Masked Language Model	90.0	83.5	24.77	7.87	12.59	7.06
Masked Seq2seq	87.0	82.1	23.40	6.80	11.43	6.19
Language Model	76.7	80.1	**21.40**	7.00	11.51	6.56
Permuted Language Model	89.1	83.7	24.03	7.69	12.23	6.96
Multitask Masked Language Model	89.2	82.4	23.73	7.50	12.39	6.74
BART Base						
w/ Token Masking	90.4	84.1	25.05	7.08	11.73	6.10
w/ Token Deletion	90.4	84.1	24.61	6.90	11.46	5.87
w/ Text Infilling	**90.8**	84.0	24.26	**6.61**	**11.05**	5.83
w/ Document Rotation	77.2	75.3	53.69	17.14	19.87	10.59
w/ Sentence Shuffling	85.4	81.5	41.87	10.93	16.67	7.89
w/ Text Infilling + Sentence Shuffling	**90.8**	83.8	24.17	6.62	11.12	**5.41**

Bidirectional encoders are crucial for SQuAD

- Left-to-right Decoder가 SQuAD에서 성능이 좋지 않은 이유는 미래 맥락이 classification에 중요하기 때문
  - 하지만 BART는 Bidirectional layer 포함되며 비슷한 성능을 달성

Model	SQuAD 1.1 F1	MNLI Acc	ELI5 PPL	XSum PPL	ConvAI2 PPL	CNN/DM PPL
BERT Base (Devlin et al., 2019)	88.5	**84.3**	-	-	-	-
Masked Language Model	90.0	83.5	24.77	7.87	12.59	7.06
Masked Seq2seq	87.0	82.1	23.40	6.80	11.43	6.19
Language Model	76.7	80.1	**21.40**	7.00	11.51	6.56
Permuted Language Model	89.1	83.7	24.03	7.69	12.23	6.96
Multitask Masked Language Model	89.2	82.4	23.73	7.50	12.39	6.74
BART Base						
w/ Token Masking	90.4	84.1	25.05	7.08	11.73	6.10
w/ Token Deletion	90.4	84.1	24.61	6.90	11.46	5.87
w/ Text Infilling	**90.8**	84.0	24.26	**6.61**	**11.05**	5.83
w/ Document Rotation	77.2	75.3	53.69	17.14	19.87	10.59
w/ Sentence Shuffling	85.4	81.5	41.87	10.93	16.67	7.89
w/ Text Infilling + Sentence Shuffling	**90.8**	83.8	24.17	6.62	11.12	**5.41**

05

Conclusion

- BART
  - 여러 Text Generation Task에서 좋은 성능
  - Discriminative Task에 대해서 RoBERTa와 유사한 성능
  - Future work,
    - Pretraining을 위한 새로운 Denoising Approach를 모색해야 한다
    - 특정 Task에 맞게 Fine-tuning 할 필요성이 있다.

Thank you

N. Hwang Hyeon Tae | E. gusxo3975@naver.com |
L..linktr.ee/oneul_