

기존 방법들

Feature-based Approaches

좋은 word embedding 을 input으로 넣어주는 것이 목적 (model의 weight parameters X)

→ task마다 모델을 따로 만들어야 하며, 모델의 parameters는 pretraining 되지 않음

e.g. ELMo

Fine-tuning Approaches

- 모델의 weight parameters를 학습하여, 이를 이용해 fine-tuning 시 좋은 seed 값으로 시작할 수 있도록 함

- uni-directional 이라는 한계 존재 → 문장 전체에 대한 이해가 중요한 task에서 성능 향상이 어려웠음

(Feature-based approach인 ELMo 도 엄격하게 말하면 양방향은 아니었음)

e.g. GPT (cf. GPT-3: Fine-tuning approach 가 아닌 Meta Learning approach)

BERT(Bidirectional Encoder Representations from Transformers)

Transformer 의 Encoders 만을 활용

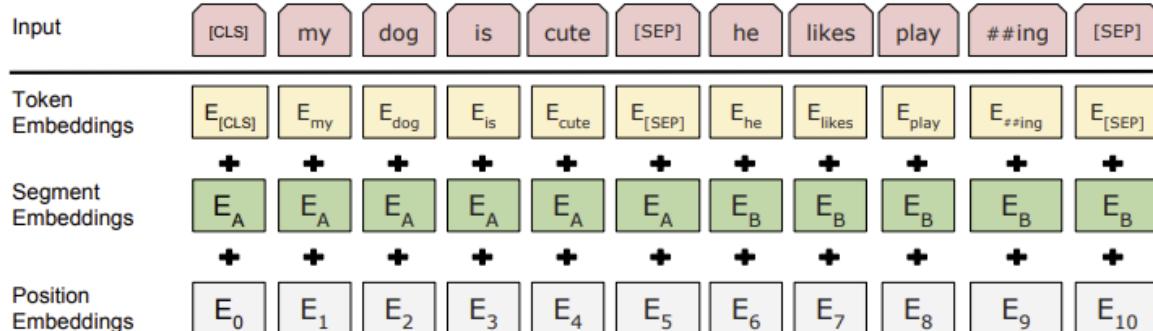
다양한 pretraining objectives 를 바탕으로 Bi-directional LM을 pre-training

Input representation

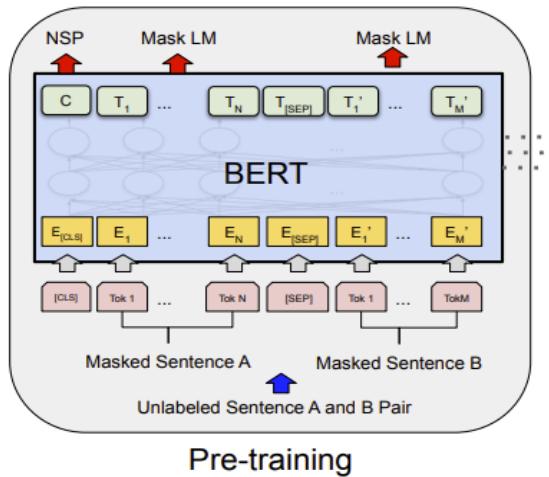
- Sentence Embedding 도 들어감

- Positional Encoding(그냥 순서대로 숫자붙임) 대신 Position Embedding 사용

- [CLS] token (classification token): pre-training, fine-tuning 수행시 항상 문장의 시작에 들어감



Pretraining: Objective



cf. GPT는 한 가지 objective: Auto-regressive 방식

1. Masked Language Model (MLM)

- 현재 time step의 masked token 을 복원하기 위해 앞, 뒤 컨텍스트 확인 → bi-directional
- Denoising Autoencoder 와는 달리, masking 된 단어만 예측 (cf. Denoising: 입력 + noise → 복원)
- 전체 학습 과정에서 일정 비율의 token 을 가리고, 원래의 문장을 복원하도록 학습
 - 전체 token 중 15%만 추론 대상으로 선정
 - 이 중 80% 를 <MASK>로 가림(=전체의 12%)
 - 이 중 10% 를 random token으로 변환(=전체의 1.5%)
 - e.g. I love to go to school 대신 I love to go to park 사용
 - 나머지는 그대로 놔둠(=전체의 1.5%)
 - 헷갈릴수록 언어모델은 고도화됨

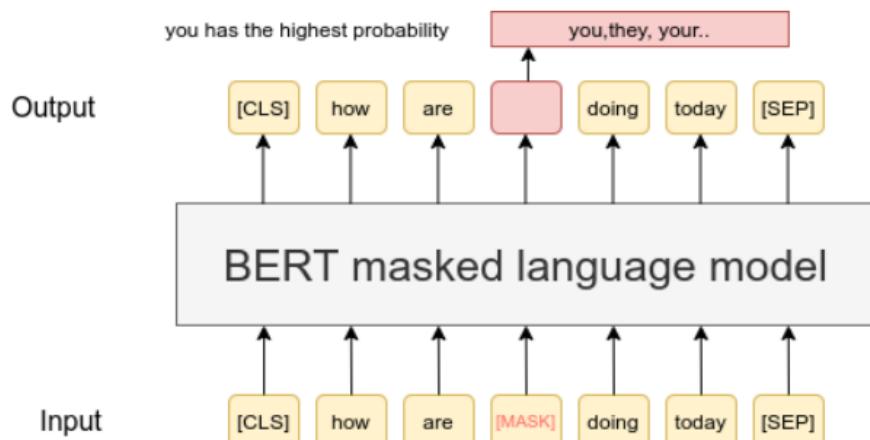


Fig. 1. BERT-Original sentence 'how are you doing today'

2. Next Sentence Prediction (NSP)

- 문장 사이의 관계를 이해하는 것이 중요할 때 사용 e.g. Question Answering, Textual Entailment task 등
- [SEP] token (seperation token): 문장 구분
- [CLS] token의 위치에서 대체 여부 예측 (e.g. 문장과 답이 다른 pair로 바뀔 수 있는가 등)

→ 잘 안쓰이게 됨

Fine-tuning: Task에 따른 구조

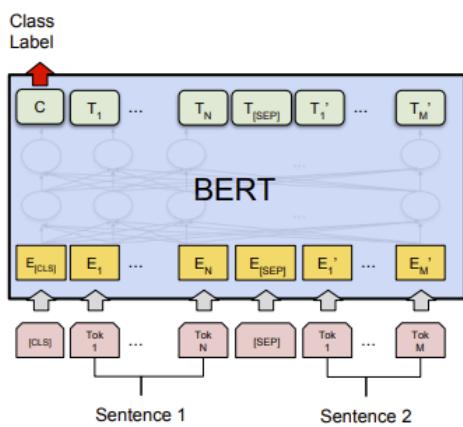
Task-specific layer 를 추가하여 fine-tuning 진행

1. Text Classification

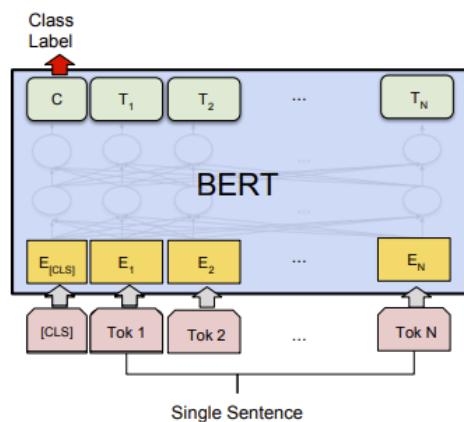
[CLS] token 의 맨 위층 hidden state에 softmax layer를 추가해서 fine-tuning

2. Spanning

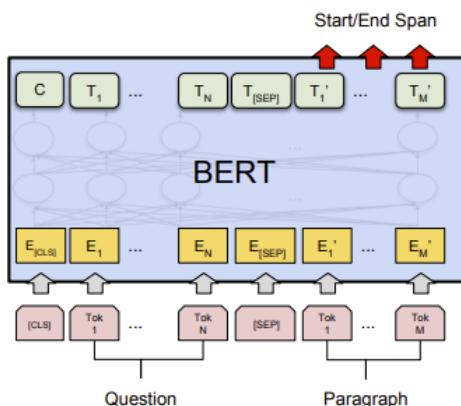
Weight vectors 추가: S (Start) & E (End)



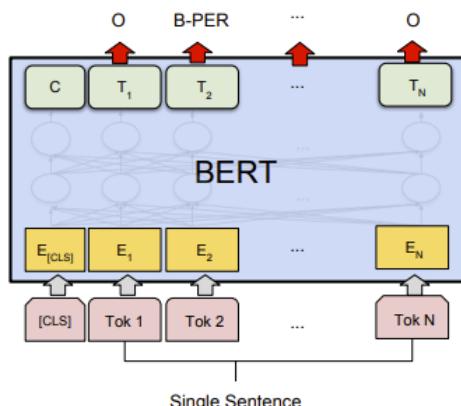
(a) Sentence Pair Classification Tasks:
MNLI, QQP, QNLI, STS-B, MRPC,
RTE, SWAG



(b) Single Sentence Classification Tasks:
SST-2, COLA



(c) Question Answering Tasks:
SQuAD v1.1



(d) Single Sentence Tagging Tasks:
CoNLL-2003 NER

의의

- ELMo 등과 비교했을 때, architecture 를 조금만 바꿔도 target task 에 적용 가능
e.g. classification 의 경우 마지막에 softmax layer 만 추가
- Bi-directional LM (MLM 방식 이용)
→ NLU 에서 성능 크게 향상 (문장 전체를 봐야하는 task들) e.g. Huggingface를 통해 쉽게 SOTA 달성 가능
- 기존 모델들에 비해 모델을 더욱 깊이, 크게 쌓을 수 있게 됨
- 가장 대중적인 PLM이 되었음

한계

- NLG 에는 적용 힘듦

Reference

<https://arxiv.org/pdf/1810.04805.pdf>
<https://tmaxai.github.io/post/BERT/>
<https://towardsdatascience.com/deconstructing-bert-distilling-6-patterns-from-100-million-parameters-b49113672f77>