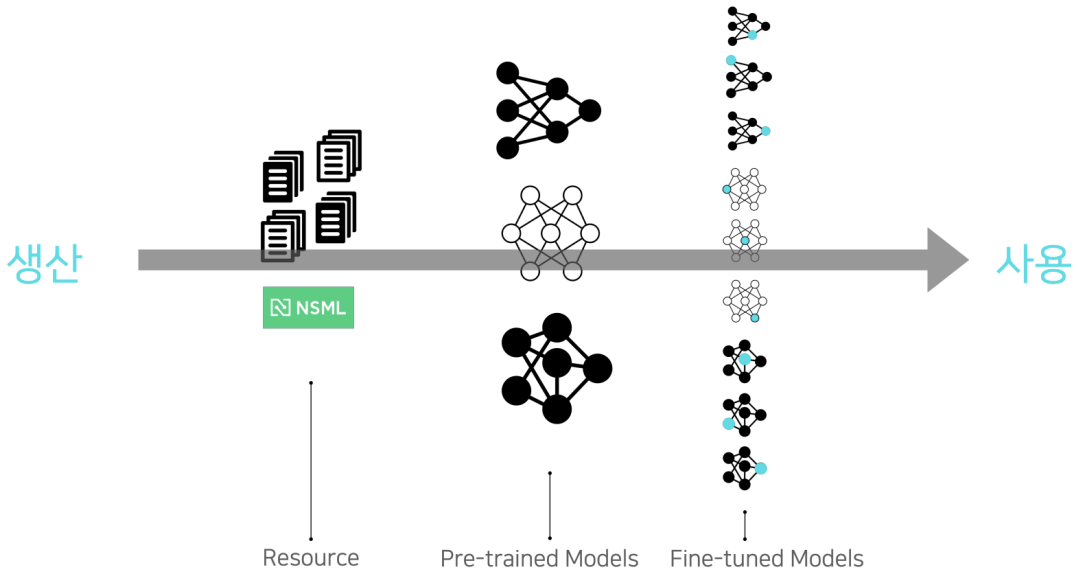
효율적이고 확장성있는 사내라이브러리 개발기

1. 프로젝트 개발 배경

- 보통 논문에서 공개된 코드 사용
  - 하지만 모델마다 다른 코드베이스 및 format
  - 사용 가능한 리소스 제한
  - ⇒ 따라서 다양한 모델을 통일하여 학습시킬 수 있는 라이브러리를 필요로함
- PLM 생산과 사용 라이브러리 구축



2. PPL 라이브러리

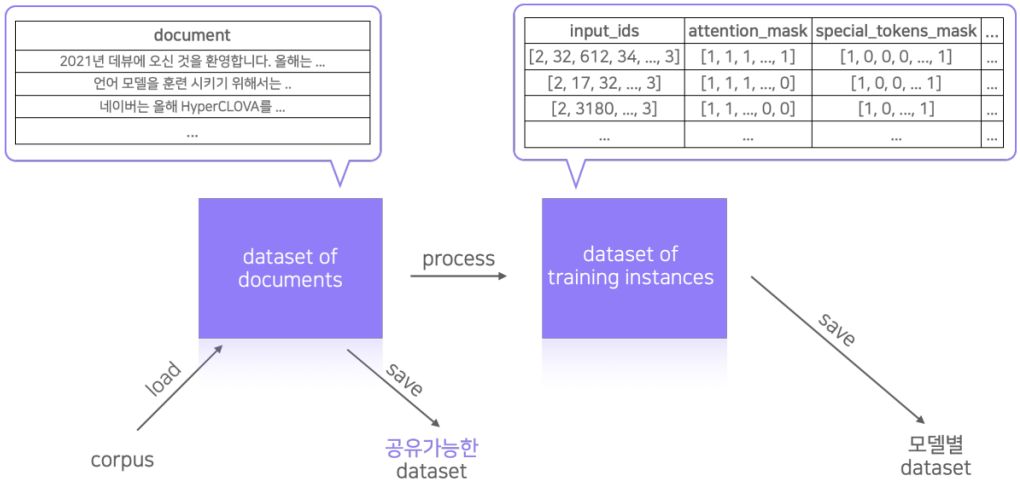
통일된 인터페이스를 중점으로

- 다양한 데이터, 모델, 학습 방법을 조합, 선택 가능하도록 일반화 ⇒ 계층적 설정 파일의 구현을 HYDRA 오픈 소스 이용

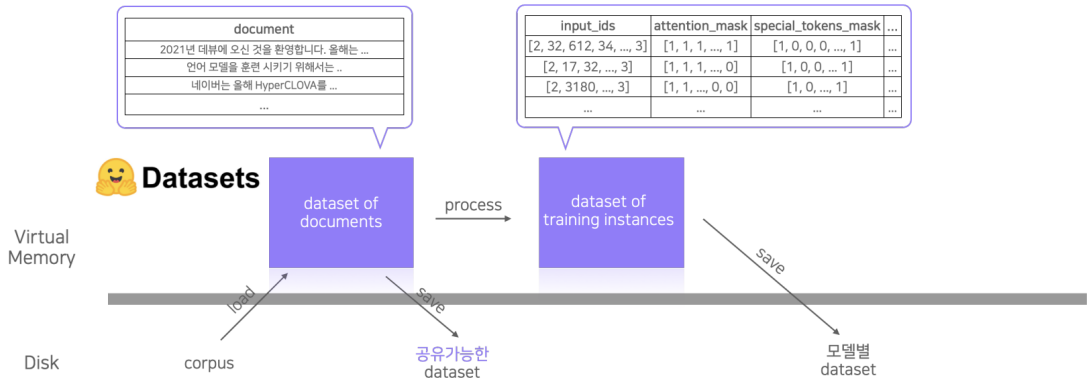
데이터 효율성 증대

- pretrain data 크기가 너무 큼

- Serialization : Raw Corpus를 학습에 사용할 수 있는 형태로 미리 바꾸기 ⇒ 훈련시간 단축
- Pipeline
  - 데이터 로딩, 전처리 과정은 모델간 공유
  - 최종결과는 모델별로 하나씩



- Arrow data
  - 메모리를 거의 사용하지 않음
  - on memory 데이터처럼 디버깅 가능



토크나이저

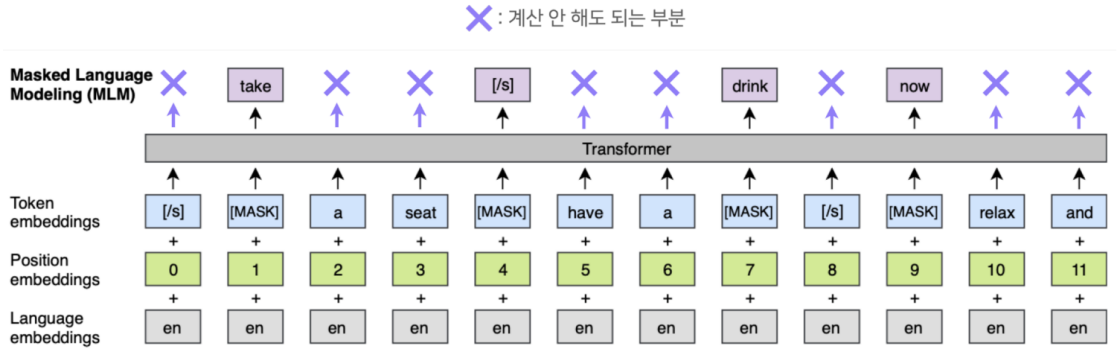
- Hugging Tokenizers를 사용하여 통일된 인터페이스 제공

- But, BPE 알고리즘으로 학습할 시 메모리 사용량 이슈

  ⇒ 해결책 : BPE 어휘 구축을 위한 학습 데이터 크기 축소 (실제로 실험을 했을때, 기존 데이터의 1%만 활용해도 기존 데이터의 vocab 크기의 98%를 차지하였음)

모델

- Hugging Transformers 사용
- loss 계산시 불필요한 부분 제거

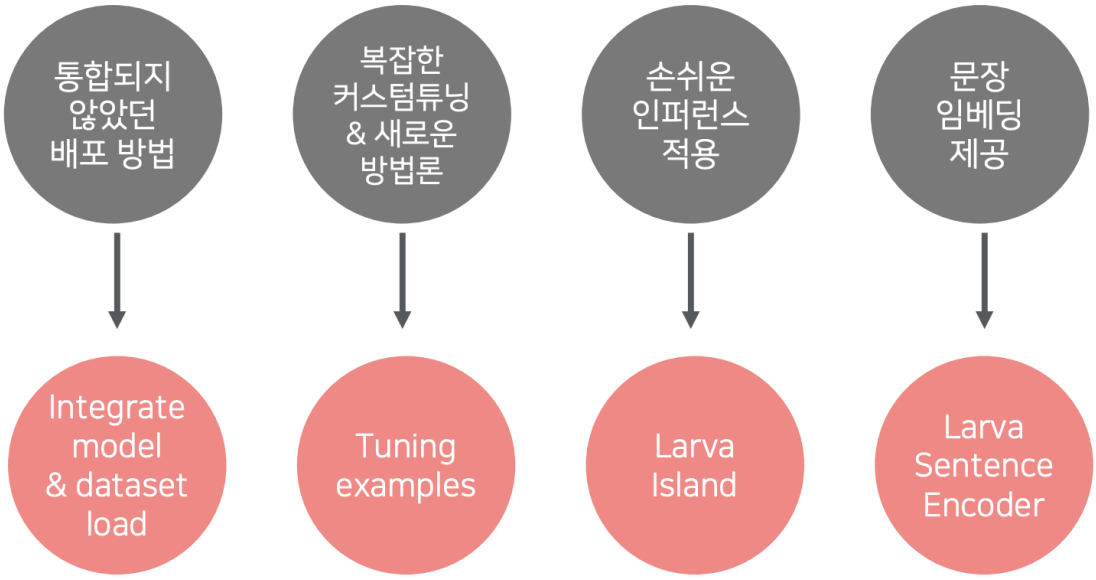


훈련

- 더 큰 모델, 더 큰 배치 사이즈 사용을 위해 DDP(Distributed Data Parallel) with pytorch lighting 사용
- 또한 모델 크기가 굉장히 큰 경우, mixed precision과 gradient accumulation 사용

3. LaRva

고려하고자 한 점



모델 사이즈 증가에 따른 finetuning 방법론 한계점

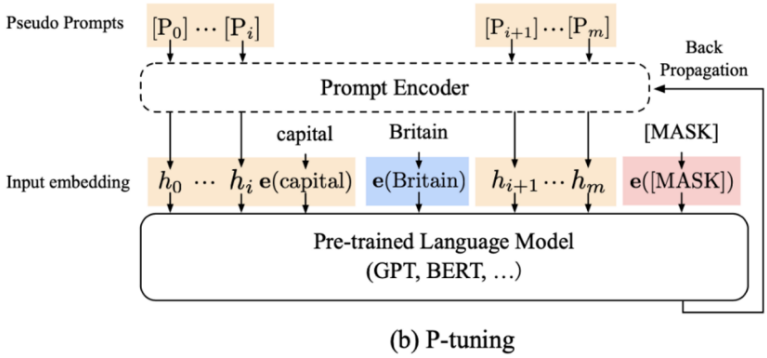
- Large scale 모델의 전체 파라미터 튜닝하기엔 어려움
- 세로운 방법론 P-tuning, LoRA를 tuning examples로 제공

In-context Learning (paper : Language models are few-shot learners)

- few-shot
- prompt engineering

Prompt based tuning (P-tuning) (paper : GPT Understands, Too.)

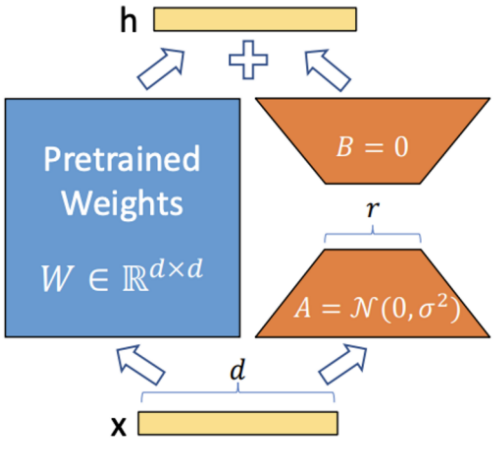
- prompt engineering 만을 학습시킴
- pretrained model의 파라미터 업데이터 X



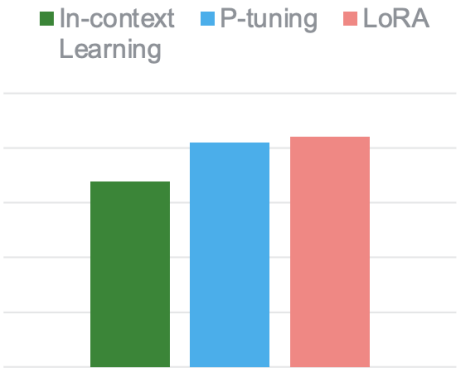
(b) P-tuning

LoRA : Low-Rank Adaption

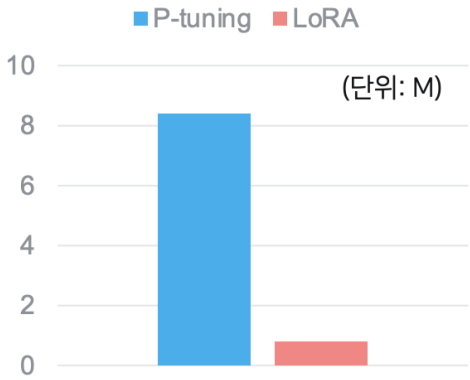
- 아래 그림의 주황식 모델만 파라미터 업데이트



실험 결과 : HyperCLOVA 1.3B실험 결과



(a) NSMC



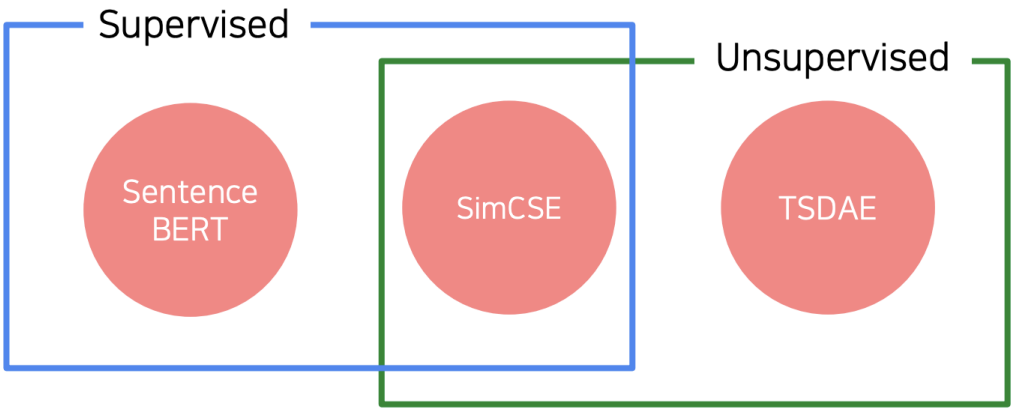
(b) 추가된 params

Larva Island

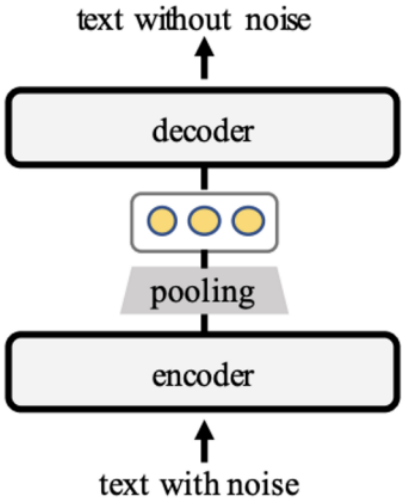
- 손쉬운 인퍼런스를 위한 프레임워크

sentence enconder

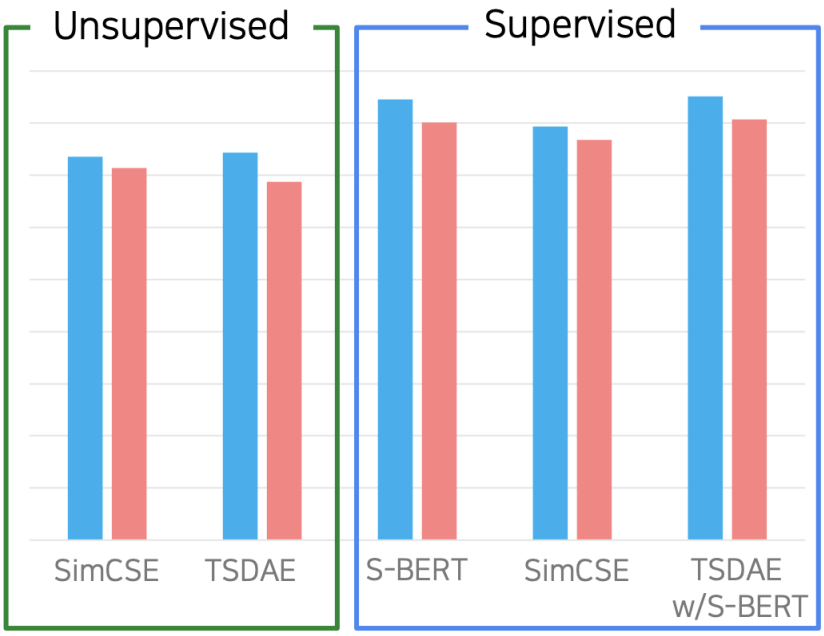
- 아래 모델 제공



- TSDAE
  - Transformer-based Sequential Denosing Auto-Encoder
  - 오토 인코더 형태로 노이즈 문장을 복원하도록 학습



- 모델 성능 결과



참고자료 링크

- https://tv.naver.com/v/23650773