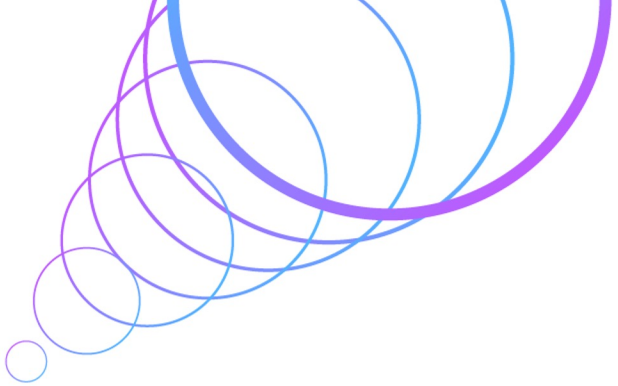


DEVIEW 2021

언어 모델 학습과 사용 쉽게 하세요!

효율적이고 확장성있는 사내 라이브러리 개발기

김정환 NAVER CLOVA
안휘진 NAVER CLOVA

CONTENTS

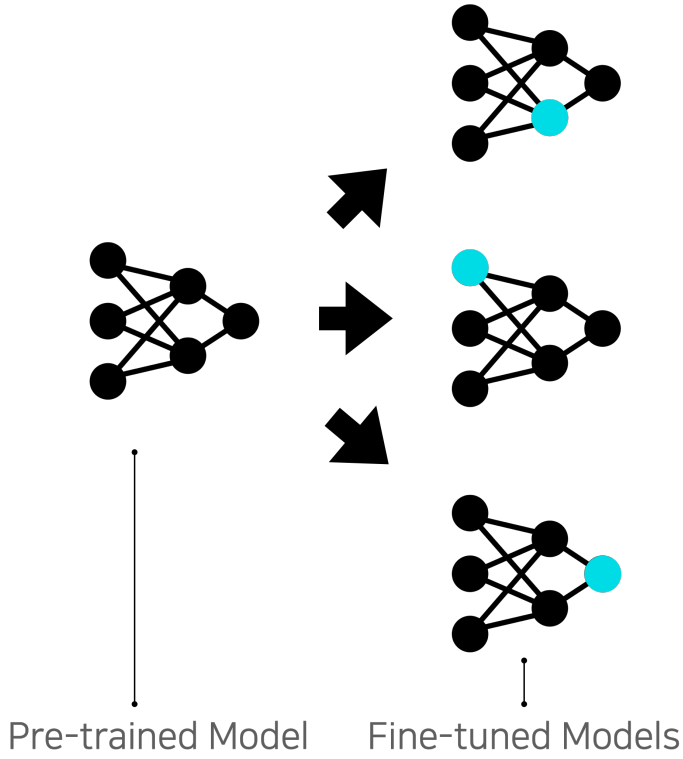
1.프로젝트 개발 배경

Transfer Learning

DEVIEW 2021

Fine-tuning

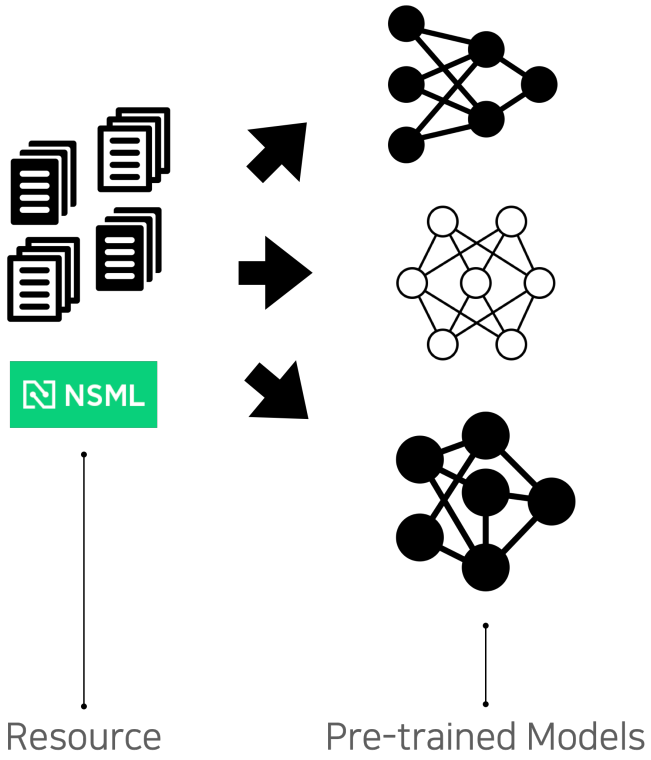
- Pre-train 단계에서 배운 지식의 전이
- 높은 성능
- 다양한 수요



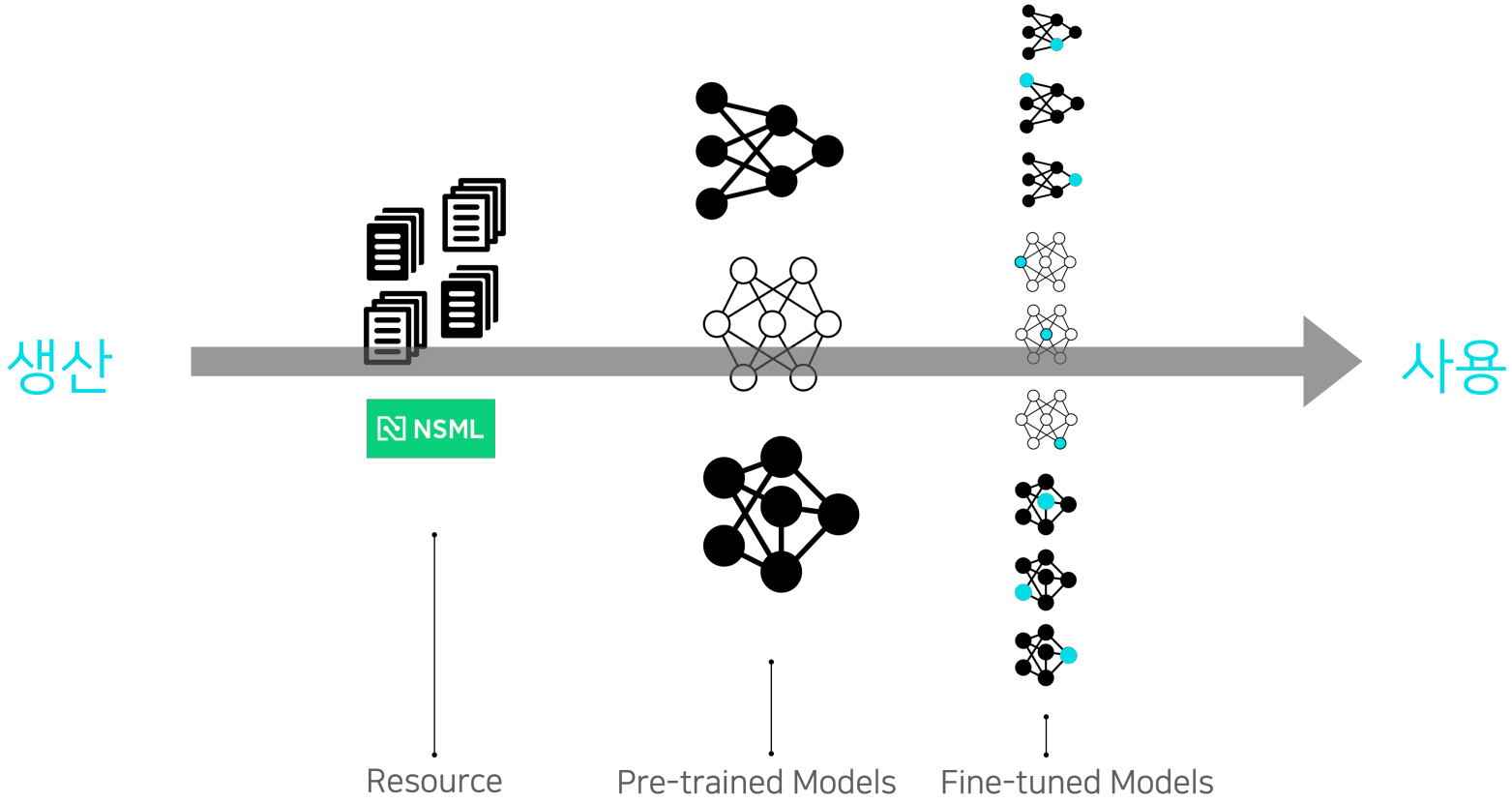
Transfer Learning

다양한 Pre-trained Language Model (PLM)

- 사용된 코퍼스
- 모델 사이즈
- 토크나이저
- …

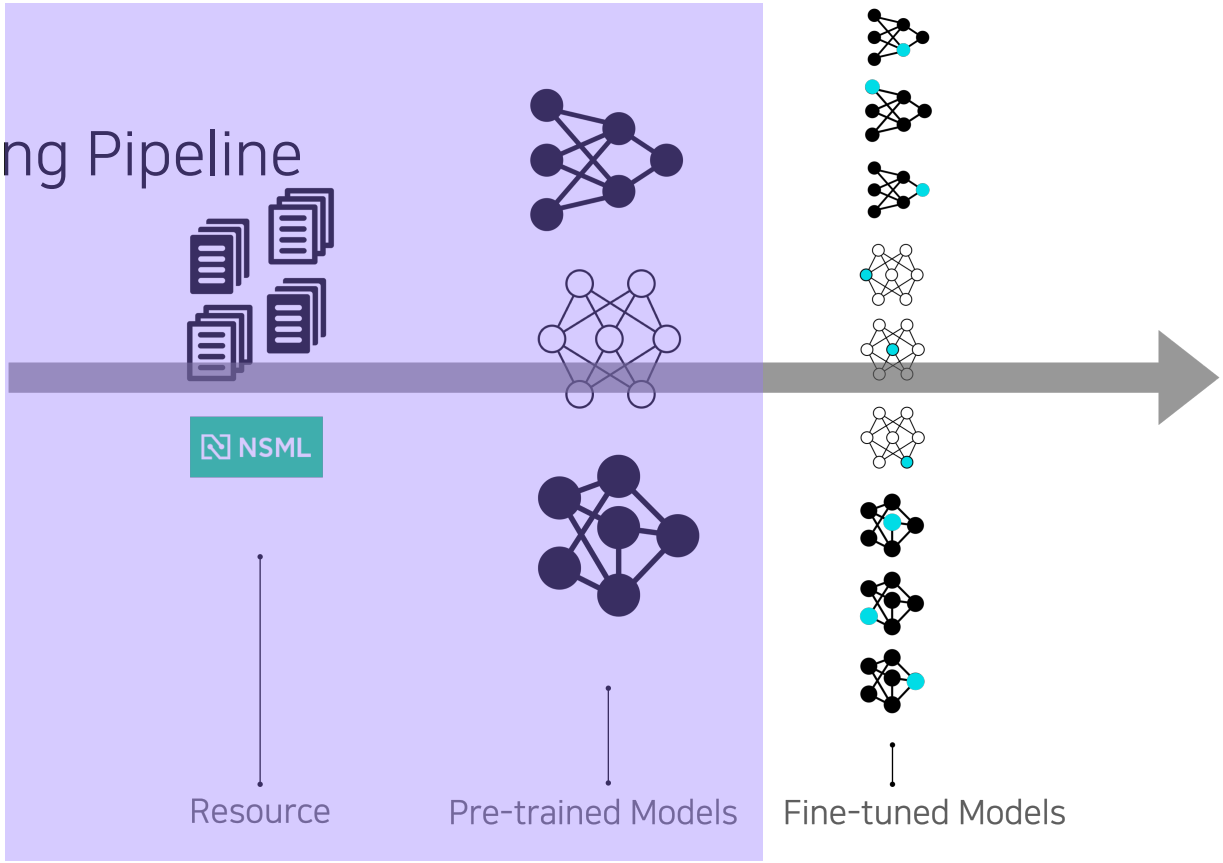


PLM 생산과 사용 라이브러리

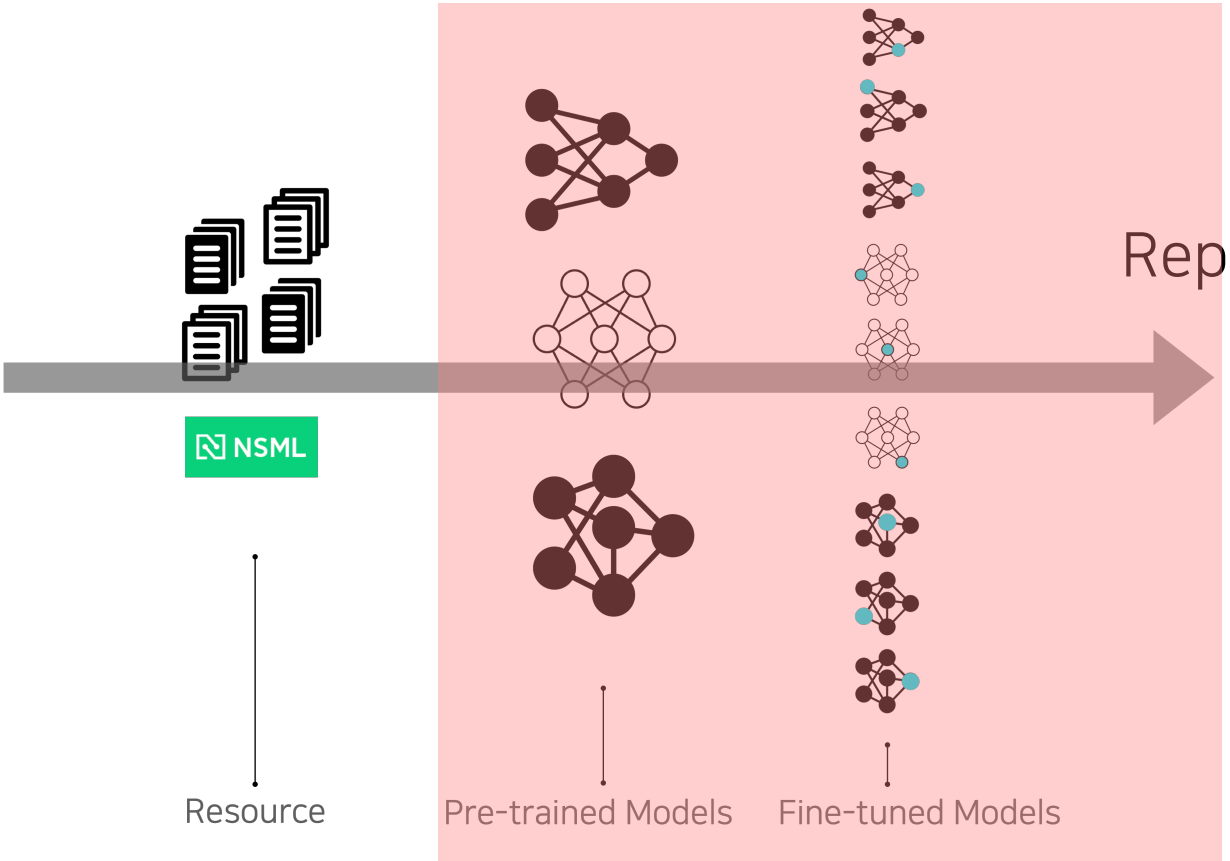


PLM 생산과 사용 라이브러리

PPL:
Pre-training Pipeline



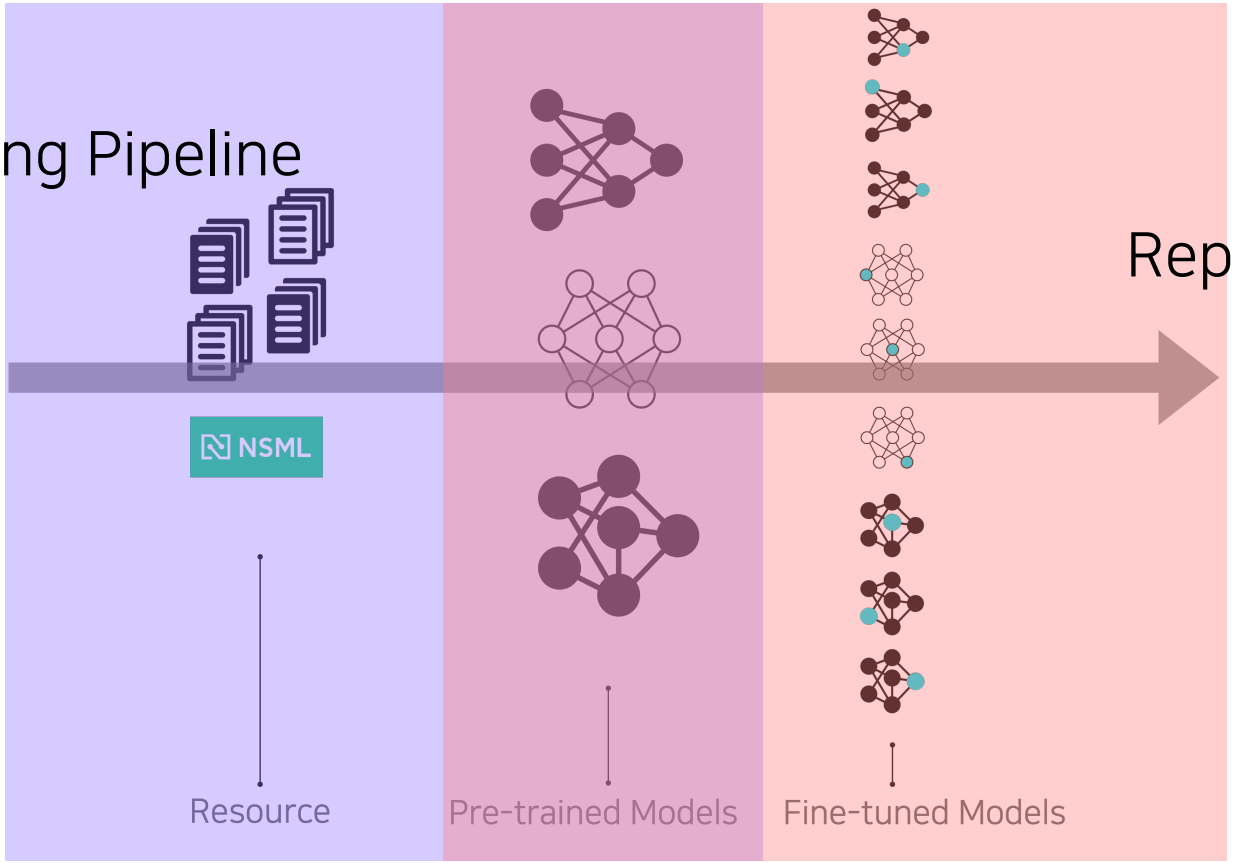
PLM 생산과 사용 라이브러리



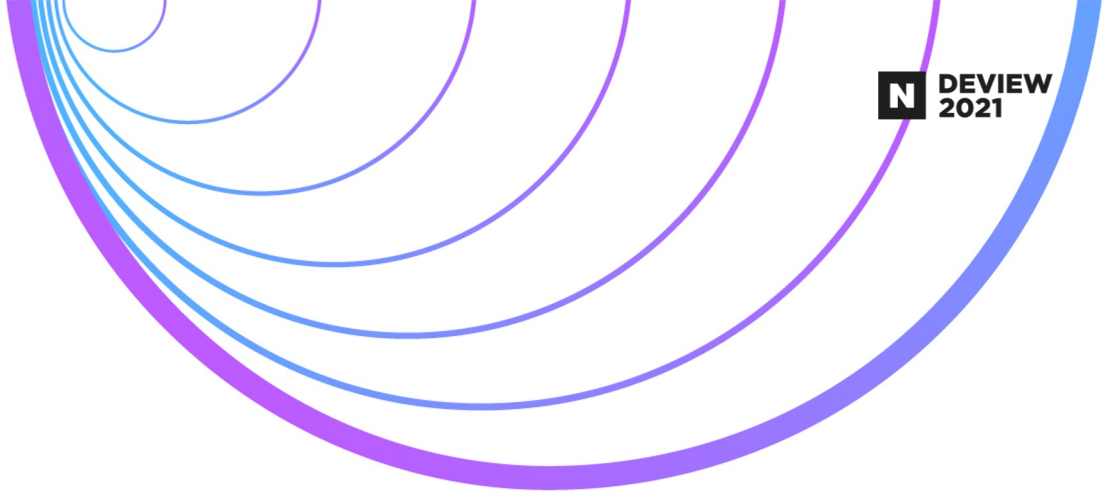
LaRva:
Language
Representation
by Clova

PLM 생산과 사용 라이브러리

PPL:
Pre-training Pipeline



LaRva:
Language
Representation
by Clova



2.PPL: Pre-training Pipeline

Pre-train을 하려면...

논문에서 공개된 코드 사용하기

- 모델마다 다른 코드 베이스
- 사용 가능한 리소스 제한
- 훈련 완료된 모델 사용자에게도 부담

google-research / **text-to-text-transfer-transformer**

google-research / **electra**

pytorch / **fairseq**

⋮

Pre-train을 하려면

DEVIEW 2021

논문에서 공개된 코드 사용하기

- 모델마다 다른 코드 베이스
- 사용 가능한 리소스 제한
- 훈련 완료된 모델 사용자에게도 부담

google-research / **text-to-text-transfer-transformer**

google-research / **electra**

pytorch / **fairseq**

⋮

여러 모델을 쉽게 pretrain 할 수 있도록

DEVIEW 2021

통일된 인터페이스

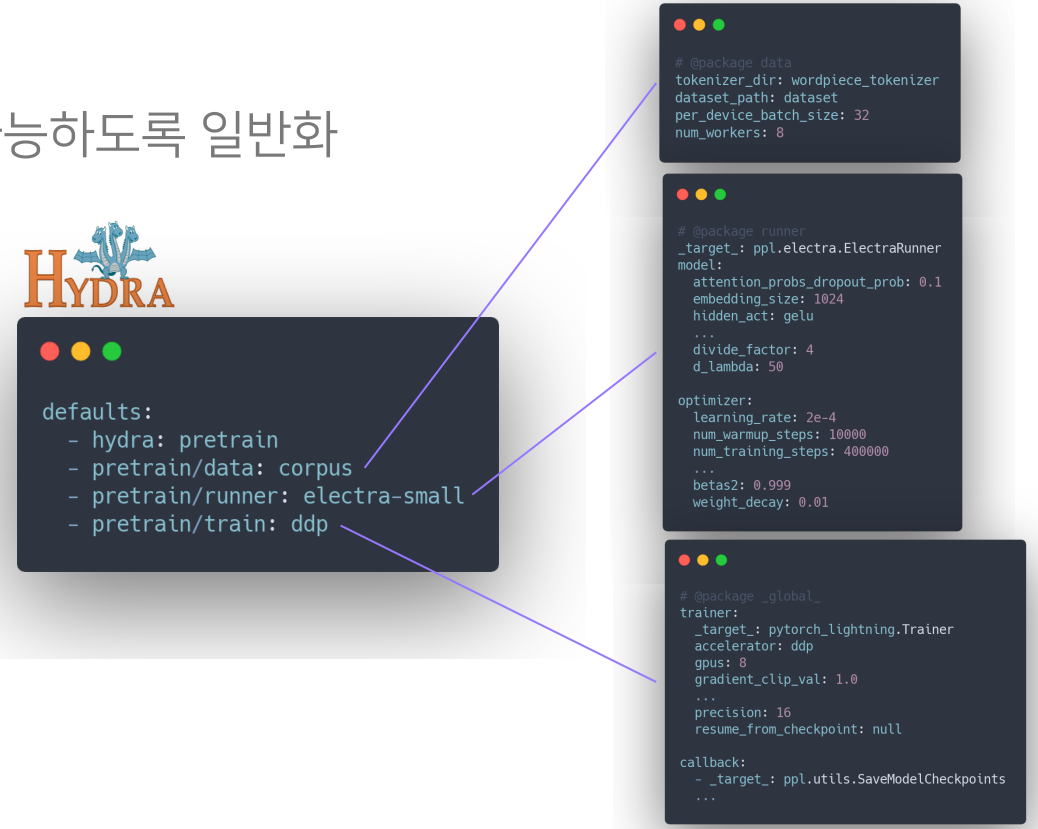
- 다양한 모델 등을 조합, 선택 가능하도록 일반화

여러 모델을 쉽게 pretrain 할 수 있도록

DEVIEW 2021

통일된 인터페이스

- 다양한 모델 등을 조합, 선택 가능하도록 일반화



데이터

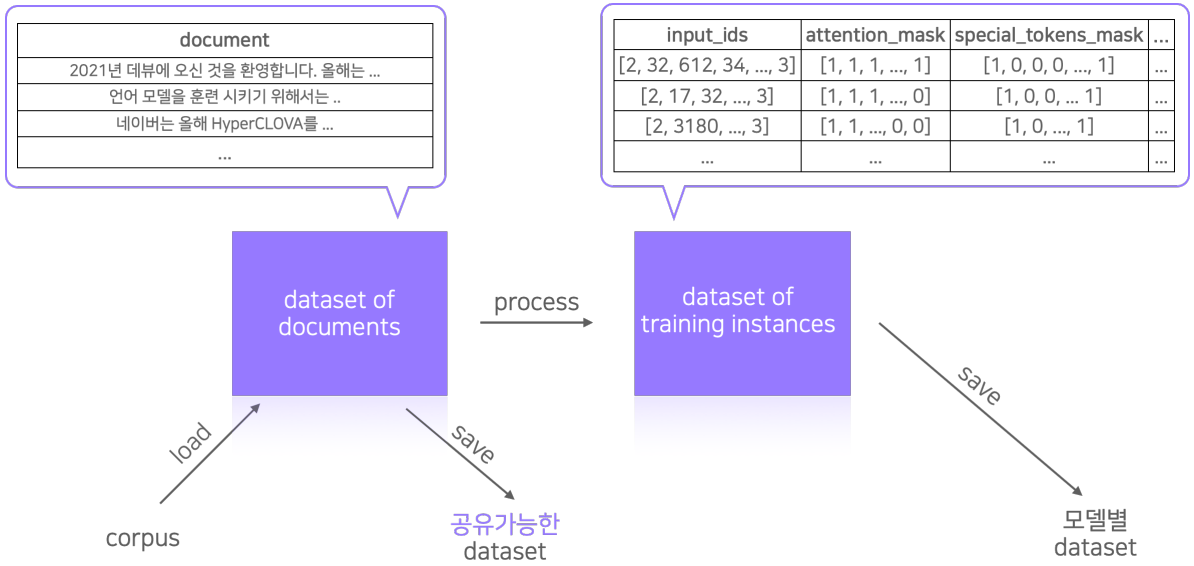
Serialization

- Raw Corpus를 학습에 사용할 수 있는 형태로 바꾸기
- 훈련 시간 단축

데이터

Pipeline

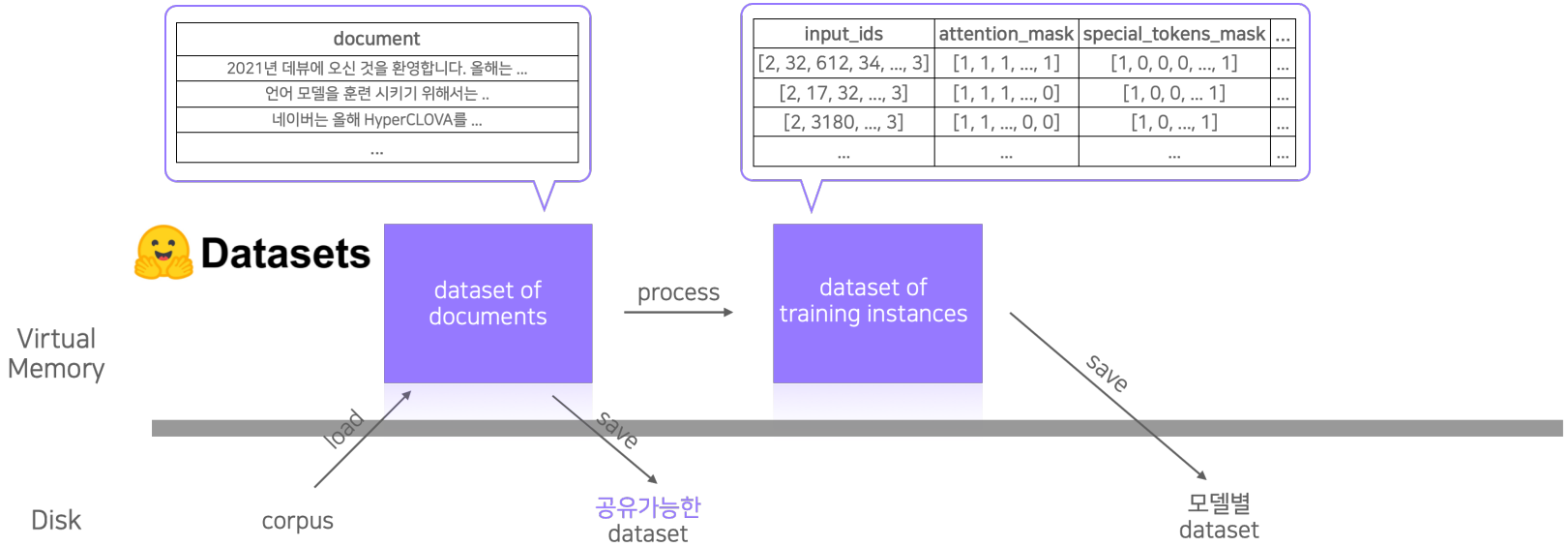
- 최종 결과는 모델 별로 하나씩
- 데이터 로딩, 전처리 과정은 모델간 공유



데이터

Arrow 데이터

- 메모리를 (거의) 사용하지 않음
- On memory 데이터처럼 디버그 가능



토크나이저

Vocabulary 학습 🤗 **Tokenizers**❤️

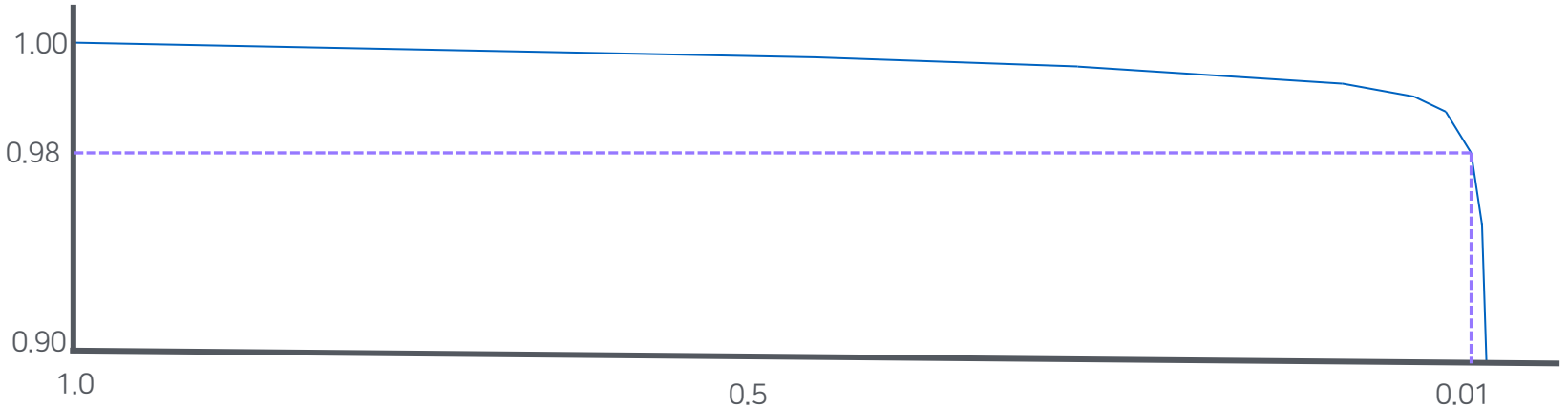
- 데이터 컬렉션 마다
- 형태소 정보가 반영된 BPE 알고리즘

토크나이저

DEVIEW 2021

BPE 메모리 사용량 이슈

- 훈련 데이터 샘플링



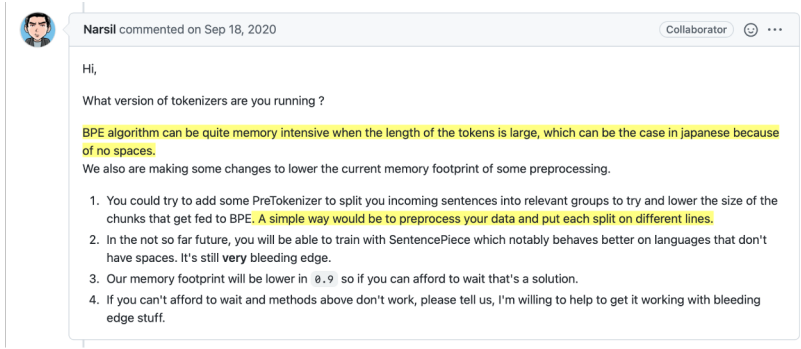
샘플링 비율에 따른 어휘 중복 비율

https://tv.naver.com/v/21681464

토크나이저

BPE 메모리 사용량 이슈

- 훈련 데이터 샘플링
- 코퍼스 포맷 전처리



Narsil commented on Sep 18, 2020 — Collaborator

Hi,

What version of tokenizers are you running ?

BPE algorithm can be quite memory intensive when the length of the tokens is large, which can be the case in japanese because of no spaces.
We also are making some changes to lower the current memory footprint of some preprocessing.

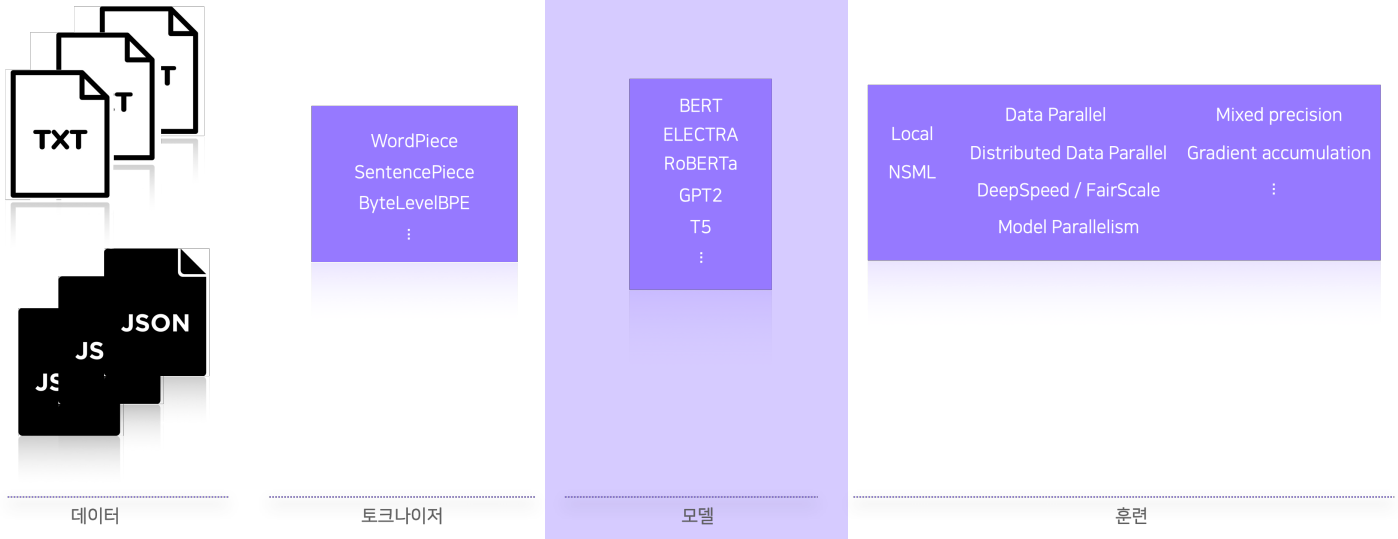
1. You could try to add some PreTokenizer to split you incoming sentences into relevant groups to try and lower the size of the chunks that get fed to BPE. A simple way would be to preprocess your data and put each split on different lines.
2. In the not so far future, you will be able to train with SentencePiece which notably behaves better on languages that don't have spaces. It's still **very** bleeding edge.
3. Our memory footprint will be lower in `0.9` so if you can afford to wait that's a solution.
4. If you can't afford to wait and methods above don't work, please tell us, I'm willing to help to get it working with bleeding edge stuff.

https://github.com/huggingface/tokenizers/issues/422

모델

다양한 모델

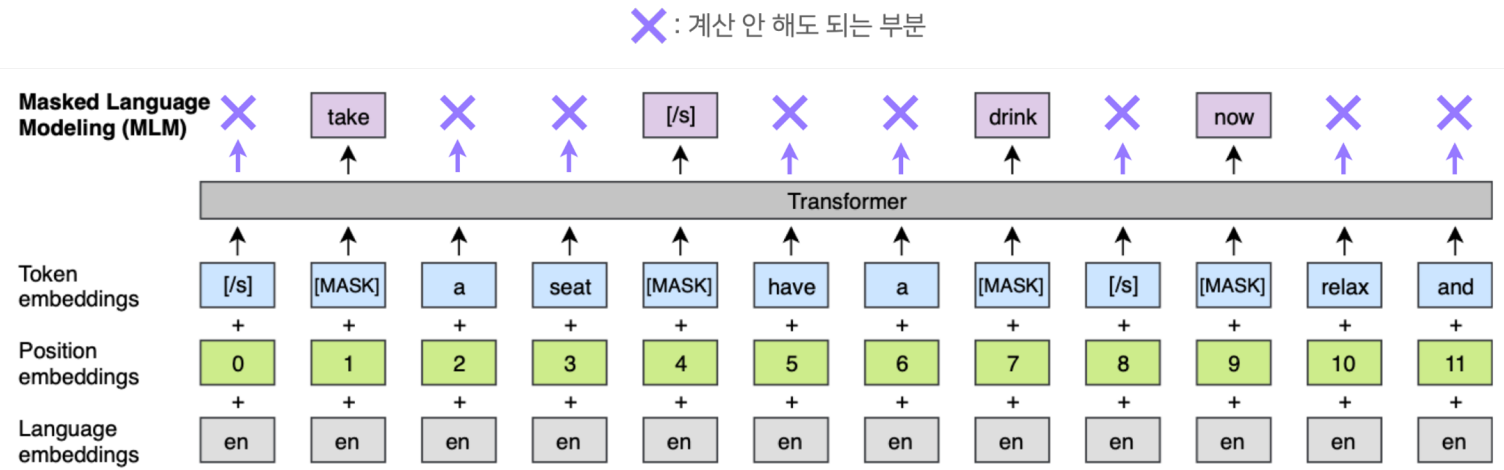
- 🤗 **Transformers** ❤️



모델

불필요한 연산을 줄이기

- Loss 계산시 불필요한 부분 제거 가능



훈련

더 큰 모델, 더 큰 배치 사이즈를 위해

- DDP with PyTorch Lightning
- Mixed precision, Gradient accumulation

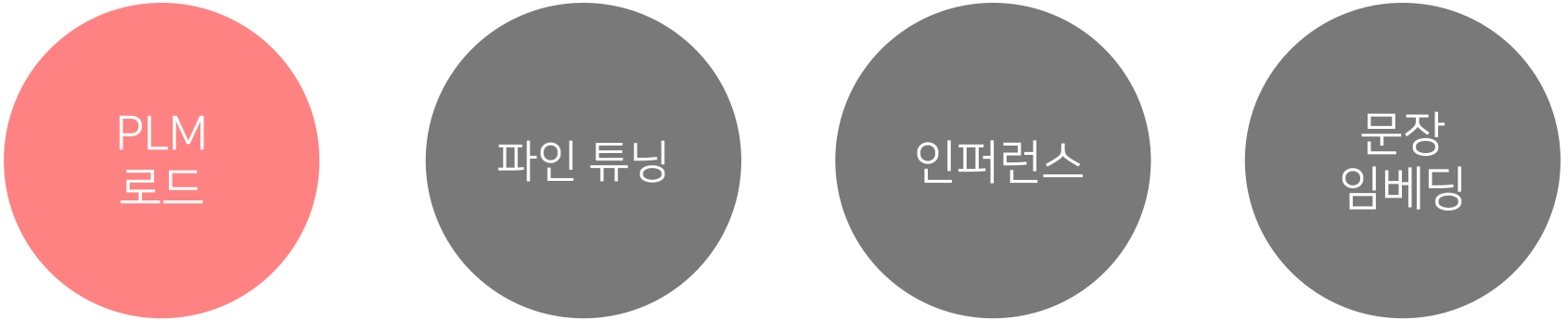
3.LaRva: Language Representations by Clova

모델 학습 이후

모델 학습 이후 어떤 기능이 필요할까?

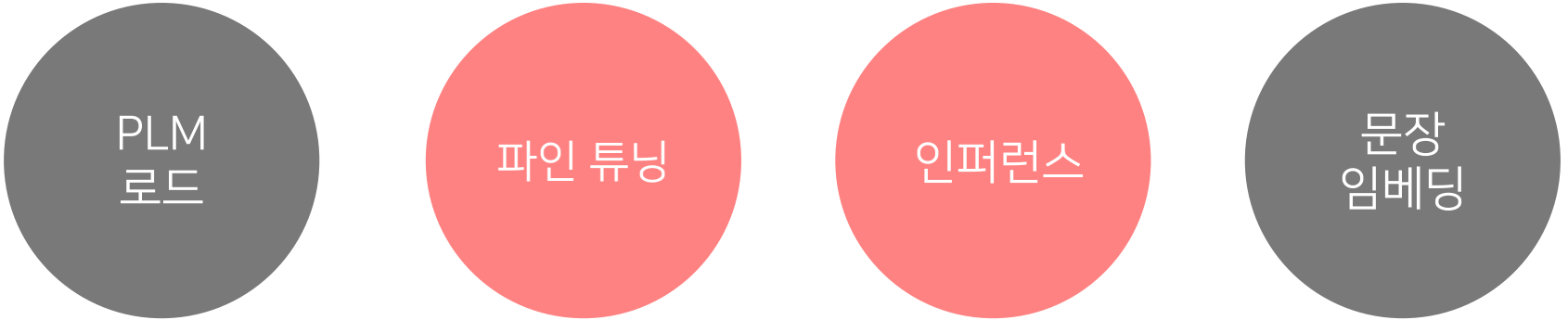
모델 학습 이후

모델 학습 이후 필요한 기능?



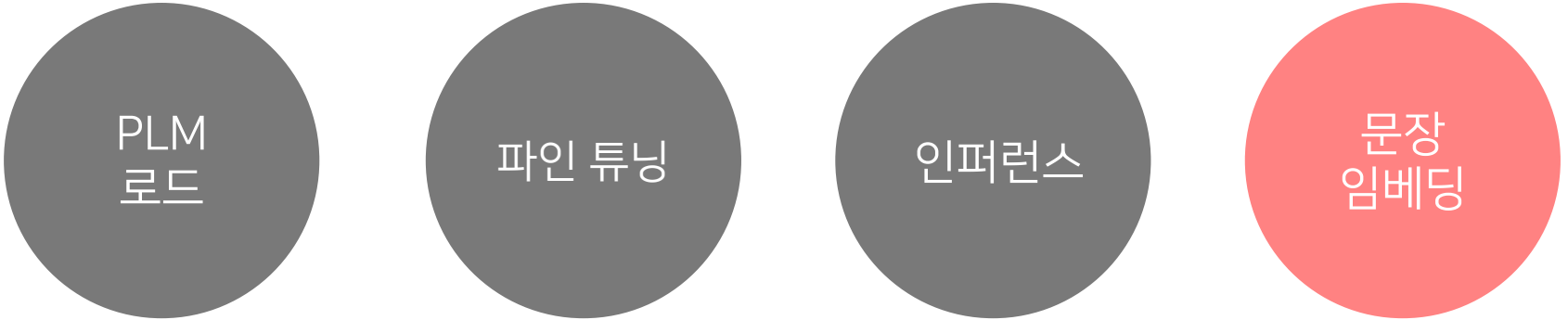
모델 학습 이후

모델 학습 이후 필요한 기능?



모델 학습 이후

모델 학습 이후 필요한 기능?



모델 학습 이후

무엇을 고려해야 할까?

- PLM 및 사내 데이터셋 배포: 기존 통일된 방법 부재
- 확장성있는 튜닝 예시
  - 데이터에 따라 전처리 등 모델러가 직접 코드 개발
  - HyperCLOVA 이후 새로운 방법론

https://unsplash.com/photos/1K9T5YiZ2WU

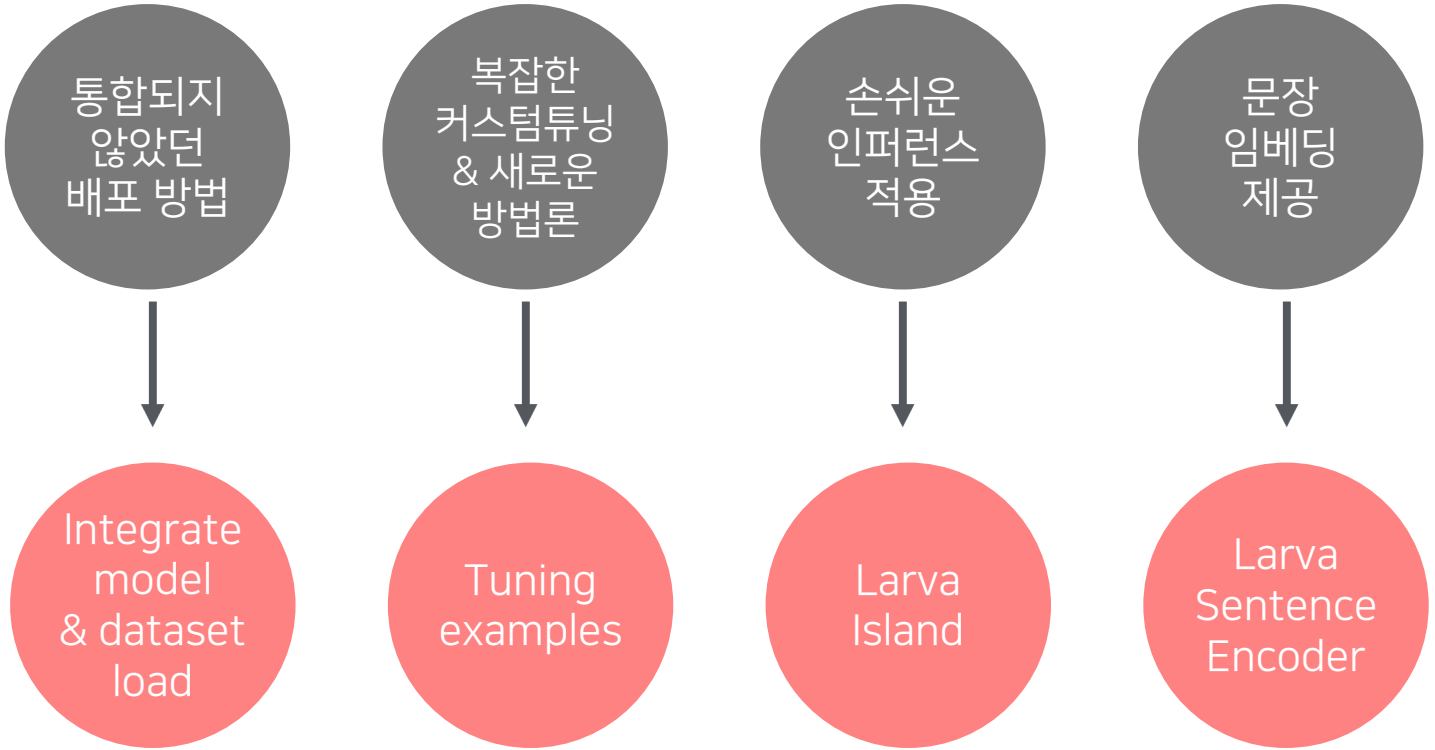
모델 학습 이후

DEVIEW 2021

무엇을 고려해야 할까?

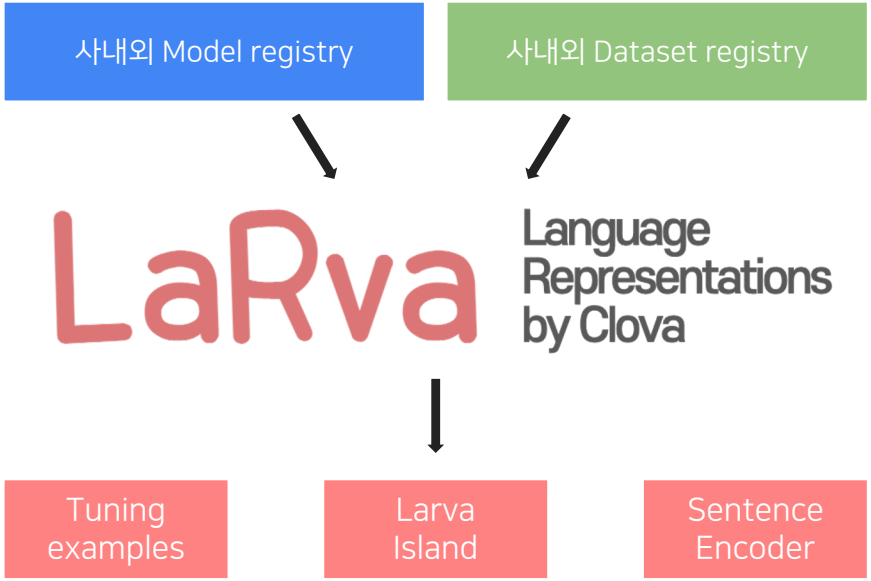
- 손쉬운 인퍼런스
  - 인퍼런스만 손쉽게 하고 싶은 사용자를 위한 프레임워크
- 문장 임베딩 제공
  - 다양한 모델 학습 방법 존재
  - 서비스에 적용 가능하도록 다양한 기능 제공

LaRva 라이브러리


통합되지 않았던 배포 방법
복잡한 커스텀튜닝 & 새로운 방법론
손쉬운 인퍼런스 적용
문장 임베딩 제공
Integrate model & dataset load
Tuning examples
Larva Island
Larva Sentence Encoder

LaRva 라이브러리

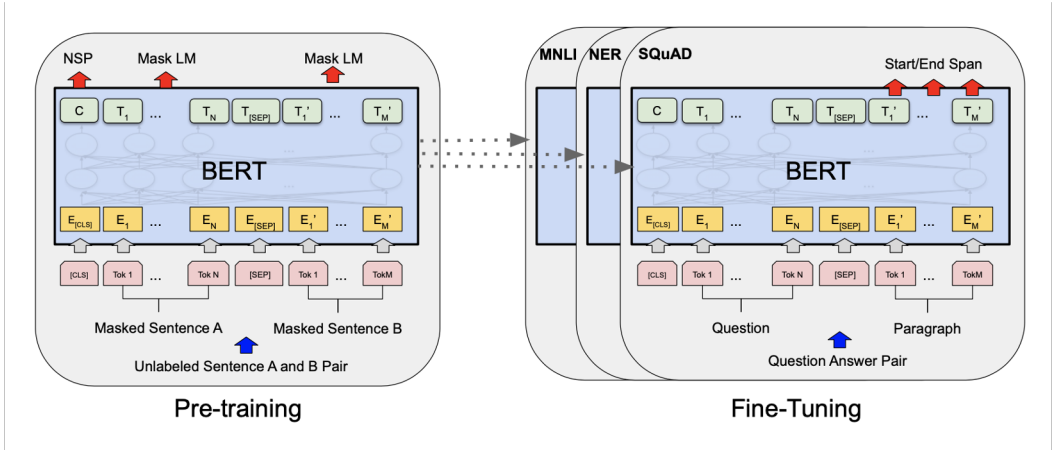
사내외 PLM 및 데이터셋 로딩 방식 통합



Tuning before HyperCLOVA

HyperCLOVA 이전의 모델을 활용한 fine-tuning

- Pre-training 단계에서 배운 지식의 “transfer”
- 모델의 파라미터 미세 조정



DEVLIN, Jacob, et al. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

Tuning before HyperCLOVA

확장성있는 fine-tuning 예시

- 데이터마다 서로 다른 포맷 & 전처리 🥲
- Base 클래스 통해 기본적인 Preprocess, Metric 등 정의
- 최소한의 코드만 모델러가 수정하도록 확장성 고려

Tuning after HyperCLOVA

모델 사이즈 증가에 따른 fine-tuning 방법론의 한계점

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

Tuning after HyperCLOVA

DEVIEW 2021

In-context Learning

- Few-shot
- Prompt engineering

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

```
1  Translate English to French:        ← task description
2  sea otter => loutre de mer          ← examples
3  peppermint => menthe poivrée
4  plush girafe => girafe peluche
5  cheese =>                           ← prompt
```

Traditional fine-tuning (not used for GPT-3)

Fine-tuning

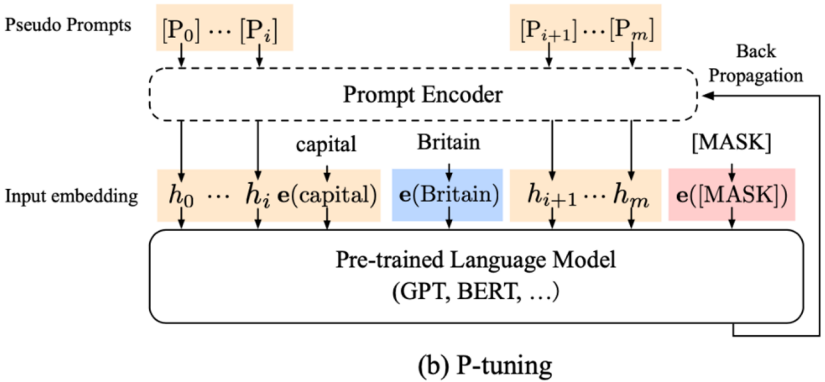
The model is trained via repeated gradient updates using a large corpus of example tasks.

```
1  sea otter => loutre de mer          ← example #1
   ↓
   gradient update
   ↓
1  peppermint => menthe poivrée        ← example #2
   ↓
   gradient update
   ↓
   • • •
   ↓
1  plush giraffe => girafe peluche     ← example #N
   ↓
   gradient update
   ↓
1  cheese =>                           ← prompt
```

BROWN, Tom B., et al. Language models are few-shot learners. arXiv preprint arXiv:2005.14165, 2020.

Tuning after HyperCLOVA

Prompt based tuning (P-tuning)

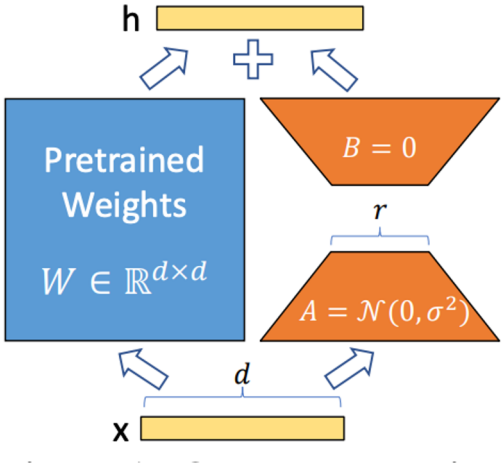


(b) P-tuning

LIU, Xiao, et al. GPT Understands, Too. arXiv preprint arXiv:2103.10385, 2021.

Tuning after HyperCLOVA

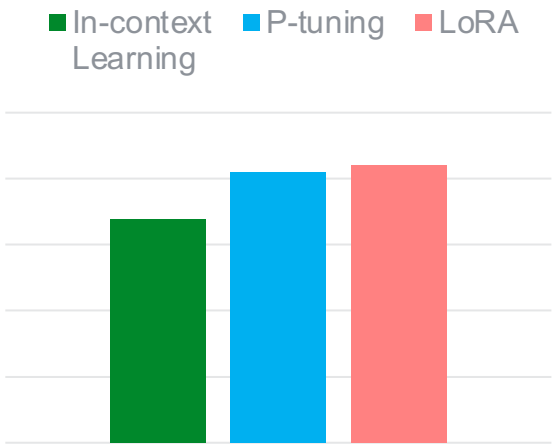
LoRA: Low-Rank Adaption



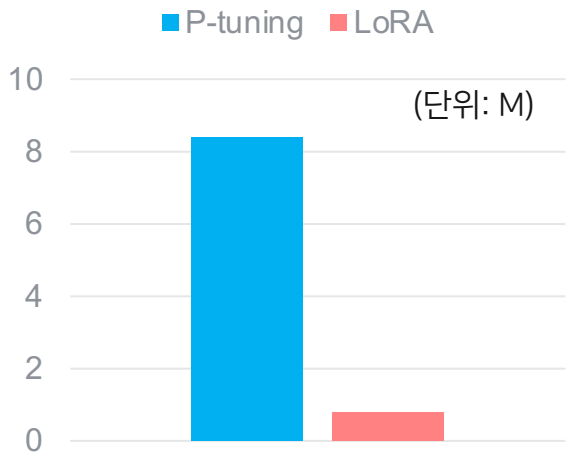
HU, Edward J., et al. LoRA: Low-Rank Adaptation of Large Language Models. arXiv preprint arXiv:2106.09685, 2021.

Tuning after HyperCLOVA

HyperCLOVA 1.3B 실험 결과



(a) NSMC



(b) 추가된 params

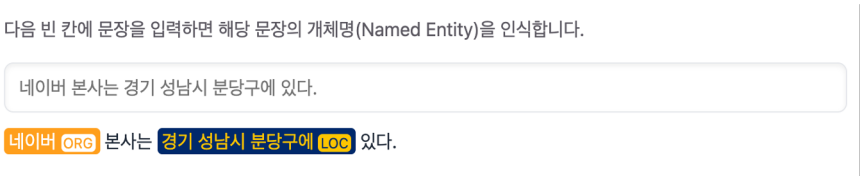
KIM, Boseop, et al. What Changes Can Large-scale Language Models Bring? Intensive Study on HyperCLOVA: Billions-scale Korean Generative Pretrained Transformers. arXiv preprint arXiv:2109.04650, 2021.

Larva Island

튜닝 이후 인퍼런스

- Larva Island: 손쉬운 인퍼런스를 위한 프레임워크
- 튜닝 모델 변경만으로 인퍼런스 가능: 손쉬운 연구/서비스 적용
- 웹데모 및 API 사용 가능
- 배치 단위 처리 가능

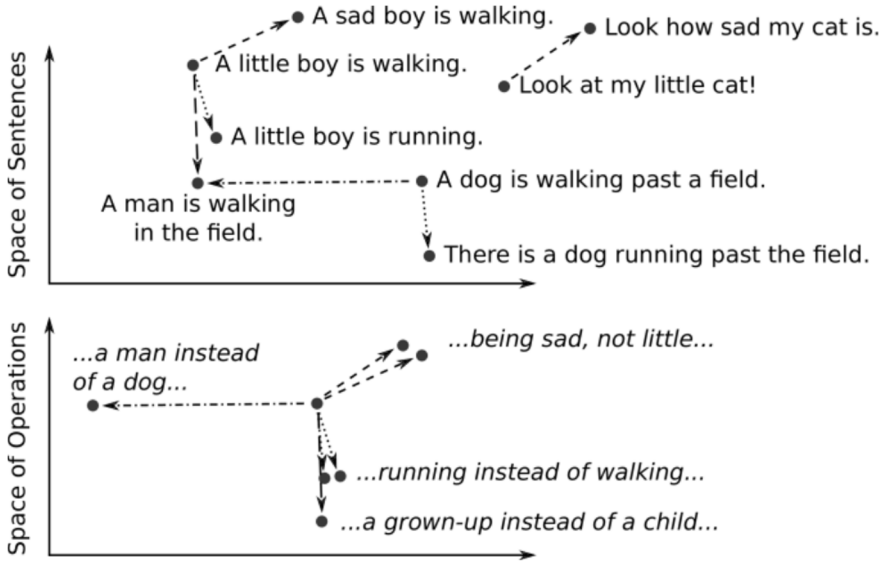
```
from larva import LarvaIsland

island = LarvaIsland("sentiment-analysis", model="model_code")

island("오늘 데뷰 발표 모두 재미있었어요!")
```



Sentence Encoder

문장 임베딩의 필요성: 다양한 사용 가능성

- Retrieval
- Clustering
- Classification



BARANČÍKOVÁ, Petra; BOJAR, Ondřej. In search for linear relations in sentence embedding spaces. arXiv preprint arXiv:1910.03375, 2019.

Sentence Encoder

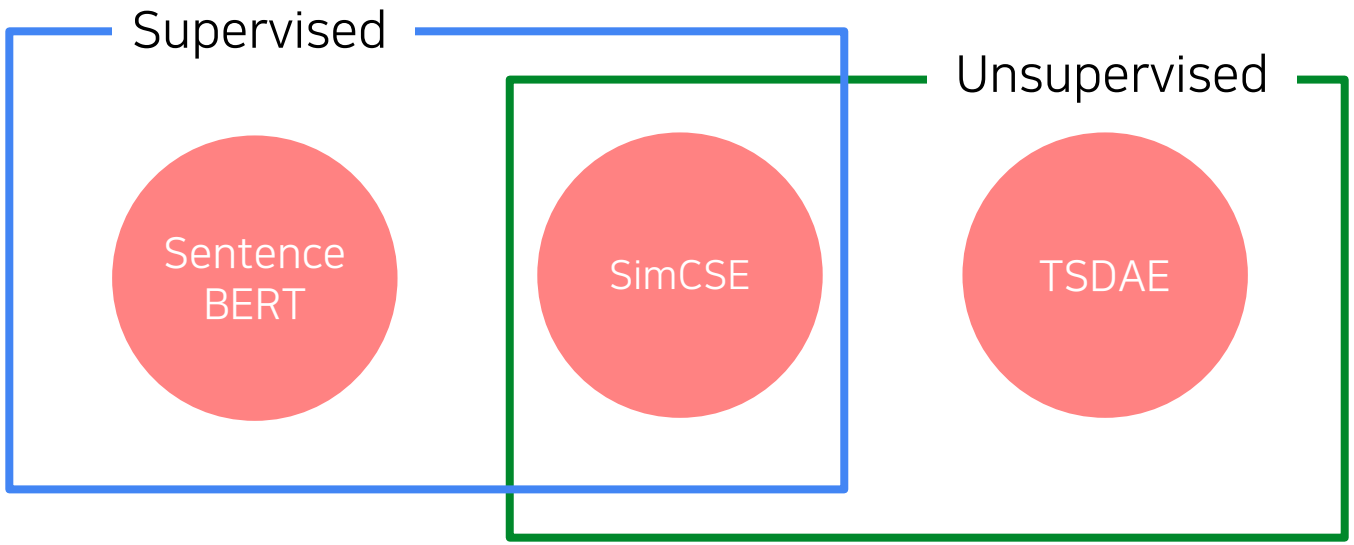
DEVIEW 2021

사용성을 위한 인터페이스

- 손쉬운 사용을 위해 허깅페이스🤗 구조와 동일하게 맞춤
- 문장 임베딩 생성, Similarity search 등의 기능 제공
- Faiss, 리소스 선택 등의 옵션 추가 제공

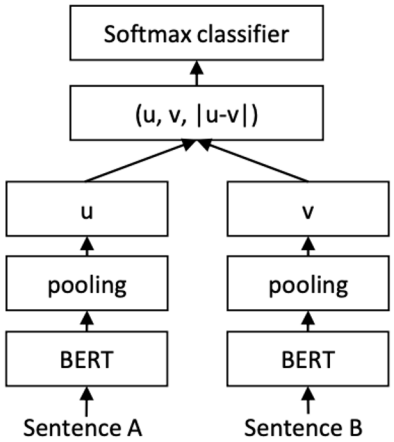
```python
from larva import LarvaSentenceEncoder

encoder = LarvaSentenceEncoder.from_pretrained("model_code", use_faiss=False)

# generate embedding
encoder.encode(["강아지가 산책을 하고 있다.", "고양이 한 마리가 낮잠을 자고 있다."])
```

Sentence Encoder

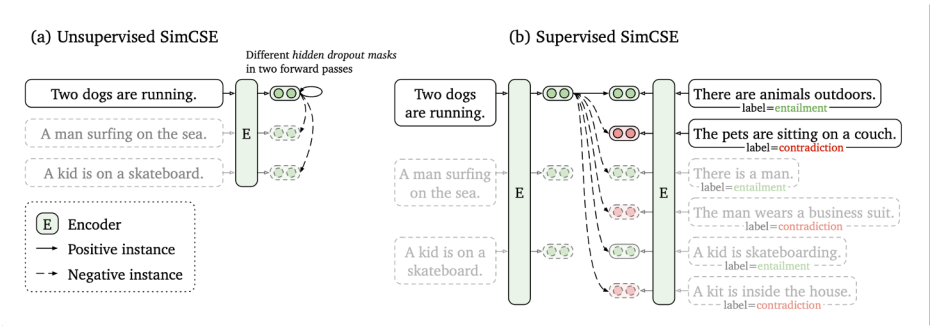


Sentence Encoder

Sentence-BERT



SimCSE

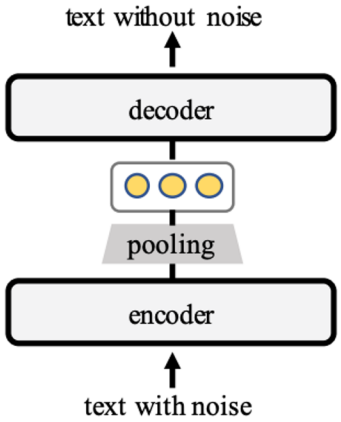


REIMERS, Nils; GUREVYCH, Iryna. Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084, 2019.
GAO, Tianyu; YAO, Xingcheng; CHEN, Danqi. SimCSE: Simple Contrastive Learning of Sentence Embeddings. arXiv preprint arXiv:2104.08821, 2021.

Sentence Encoder

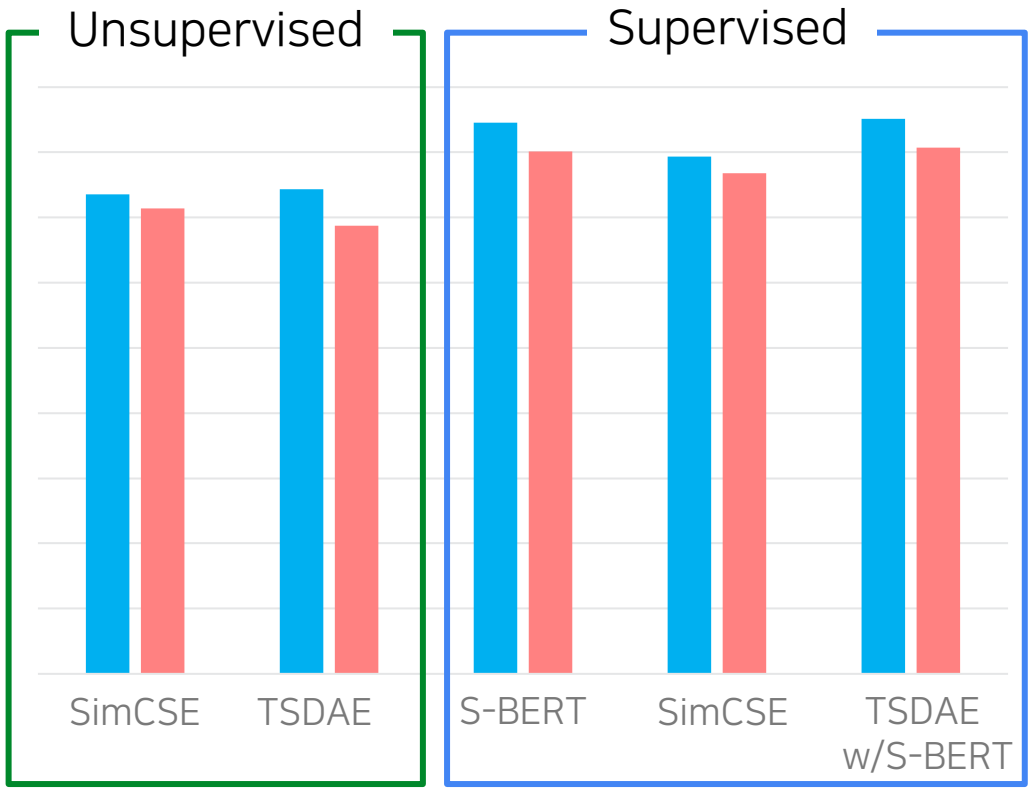
TSDAE

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



WANG, Kexin; REIMERS, Nils; GUREVYCH, Iryna. TSDAE: Using Transformer-based Sequential Denoising Auto-Encoder for Unsupervised Sentence Embedding Learning. arXiv preprint arXiv:2104.06979, 2021.

Sentence Encoder



4. Future Work

Challenges

앞으로 넘어야 할 문제들

- 실제 서비스 적용? 큰 사이즈의 모델은 아직 어려움이 있음
  - 모델 Parallelism
  - 모델 경량화 – Distill, Quantization

https://unsplash.com/photos/VYTQNnaboUA

Challenges

다양한 모델 제공

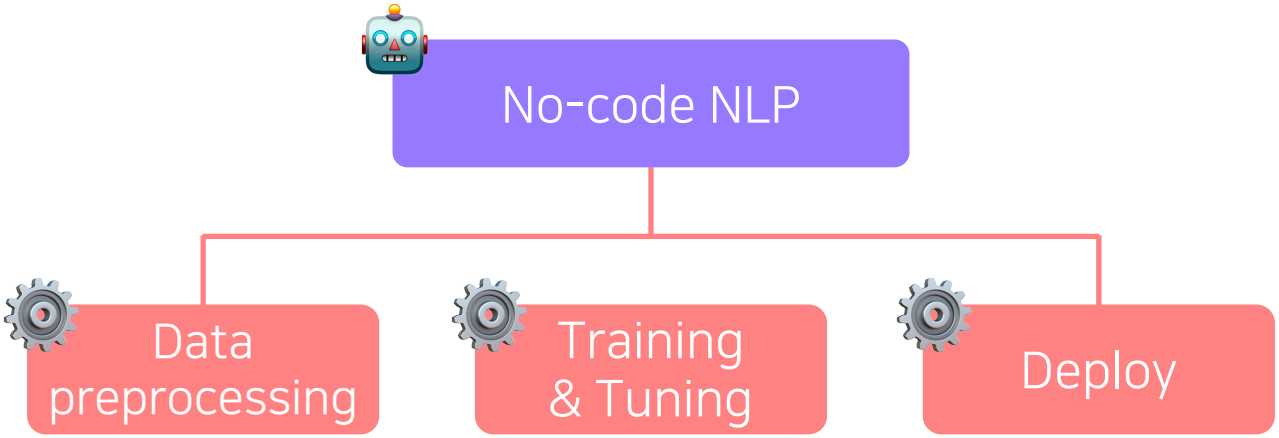
- 다국어 Multi-lingual 모델 제공
- 이미지 등의 Multi-modal 모델 제공

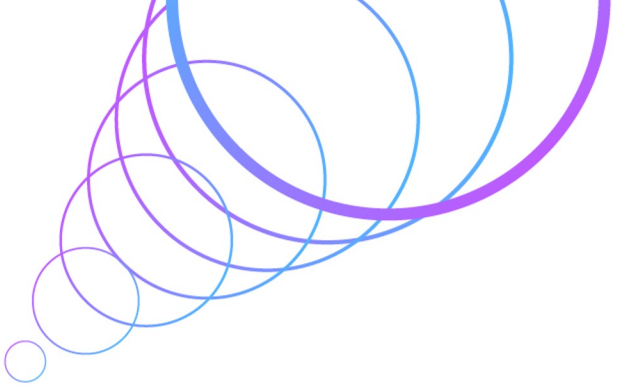
https://towardsdatascience.com/multilingual-transformers-ae917b36034d

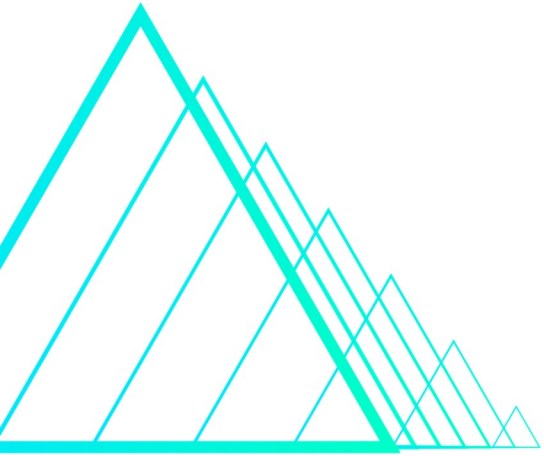
No-code NLP

손쉬운 No-code NLP

- 데이터 전처리, 학습, 배포 등의 자동화: No-code
- 작은 스케일의 LM부터 Large scale의 LM까지 포괄







Thank You