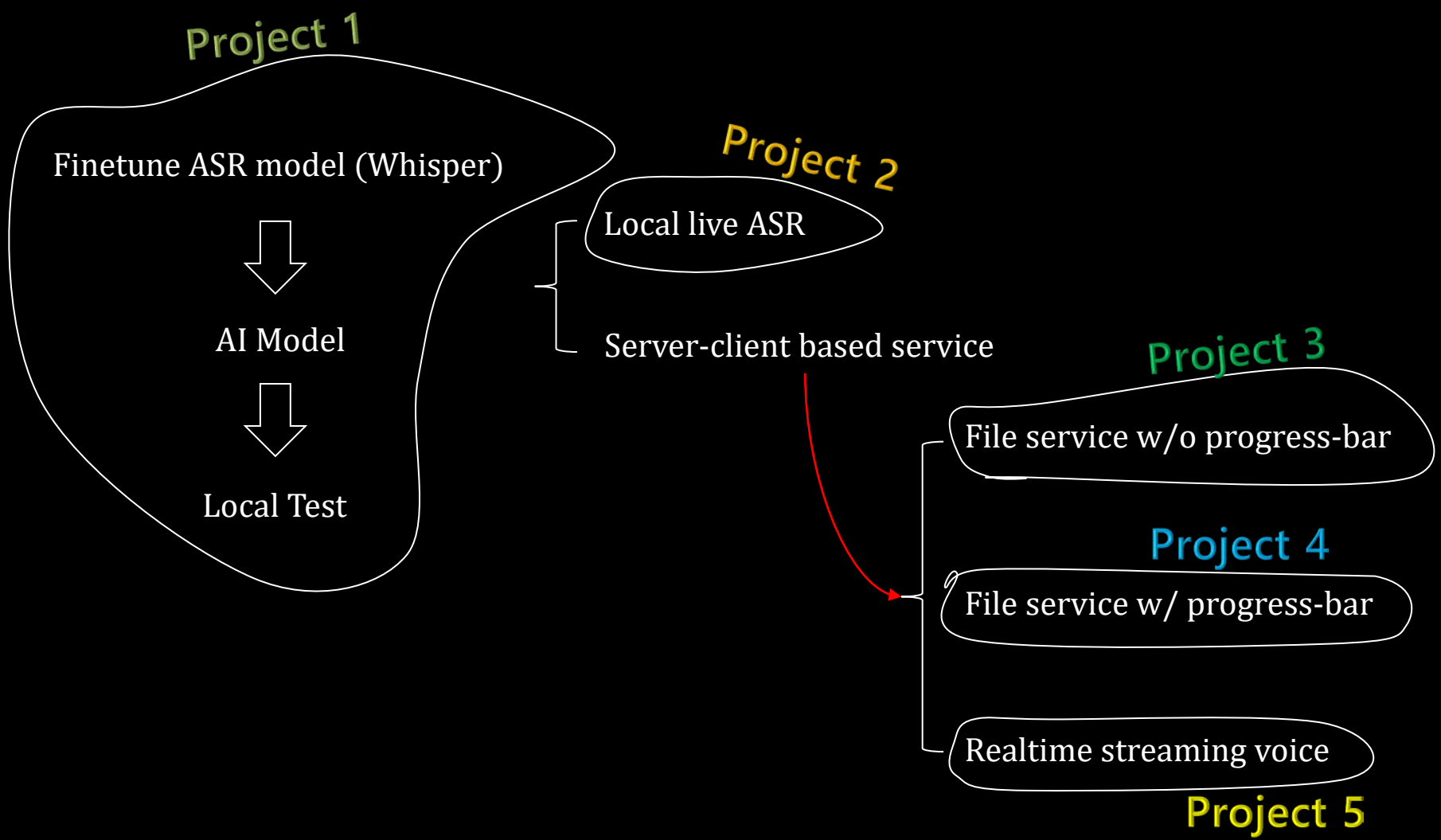
OpenAI - Whisper Fine-tuning
(Master Plan of Attack!)

소프트웨어 꼰대 강의

노기섭 교수

(kafa46@cju.ac.kr)

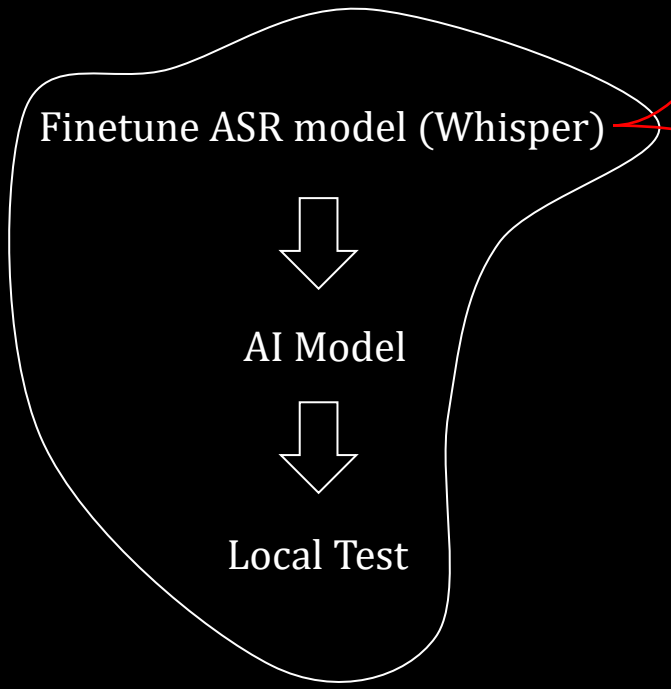
Big Picture


Project 1
Finetune ASR model (Whisper)
AI Model
Local Test
Project 2
Local live ASR
Server-client based service
Project 3
File service w/o progress-bar
Project 4
File service w/ progress-bar
Realtime streaming voice
Project 5

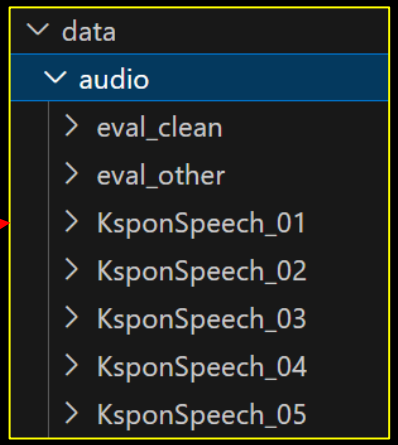
지금까지 우리가 해온 작업은?

실제로 한 일은....

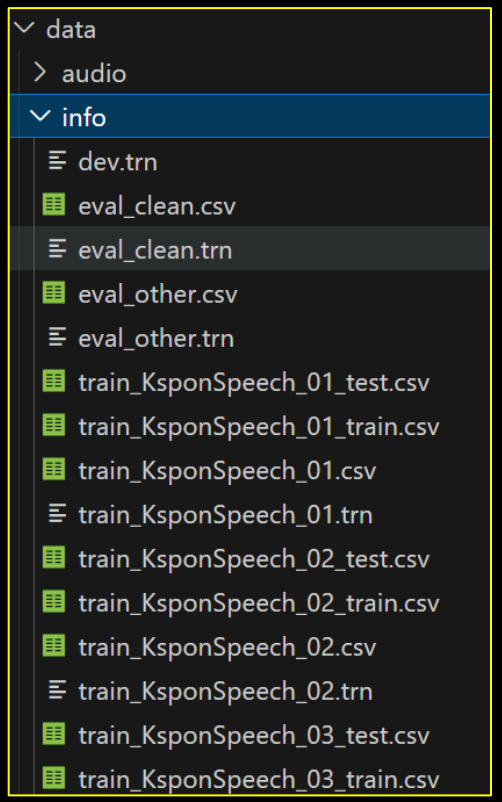
데이터 전처리 작업 ㅠ



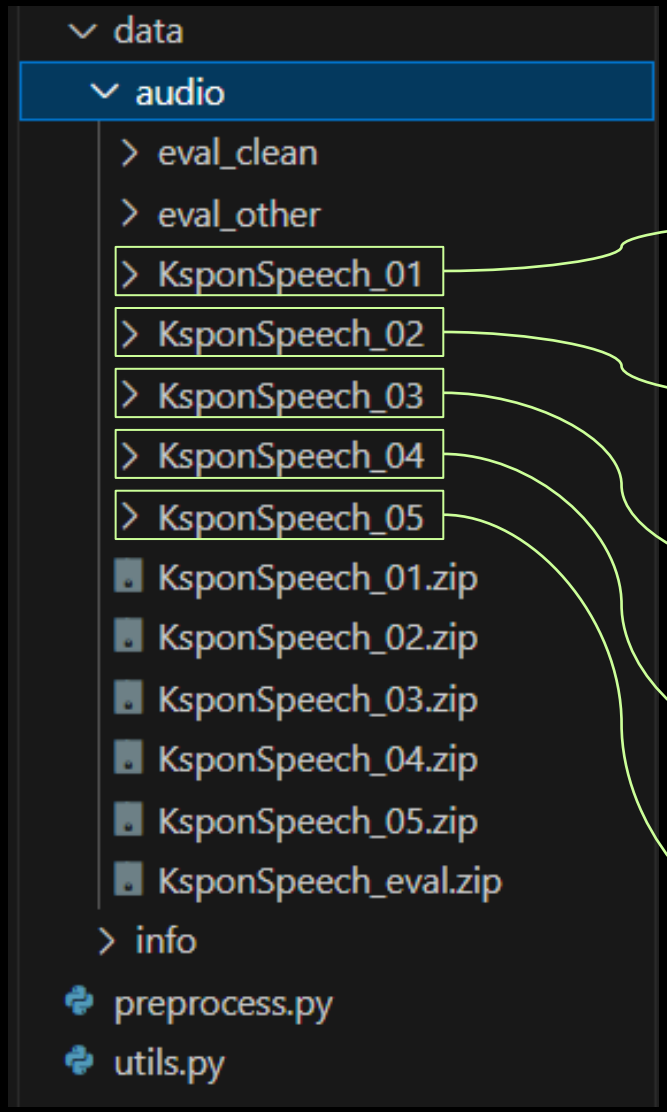
음성파일
전처리



텍스트 파일
(Transcript)
전처리



Finetuning 전략



Pre-trained model
→ "openai/whisper-small", "openai/whisper-medium", etc.

finetuning #1 → Updated pre-trained model #1

finetuning #2 → Updated pre-trained model #2

finetuning #3 → Updated pre-trained model #3

finetuning #4 → Updated pre-trained model #4

finetuning #5 → Updated pre-trained model #5

Final Model for us ^^

Main References

https://huggingface.co/blog/fine-tune-whisper

가장 집중해서 참고할 문서 (∵Huggingface official recommendation)

https://velog.io/@mino0121/NLP-OpenAI-Whisper-Fine-tuning-for-Korean-ASR-with-HuggingFace-Transformers

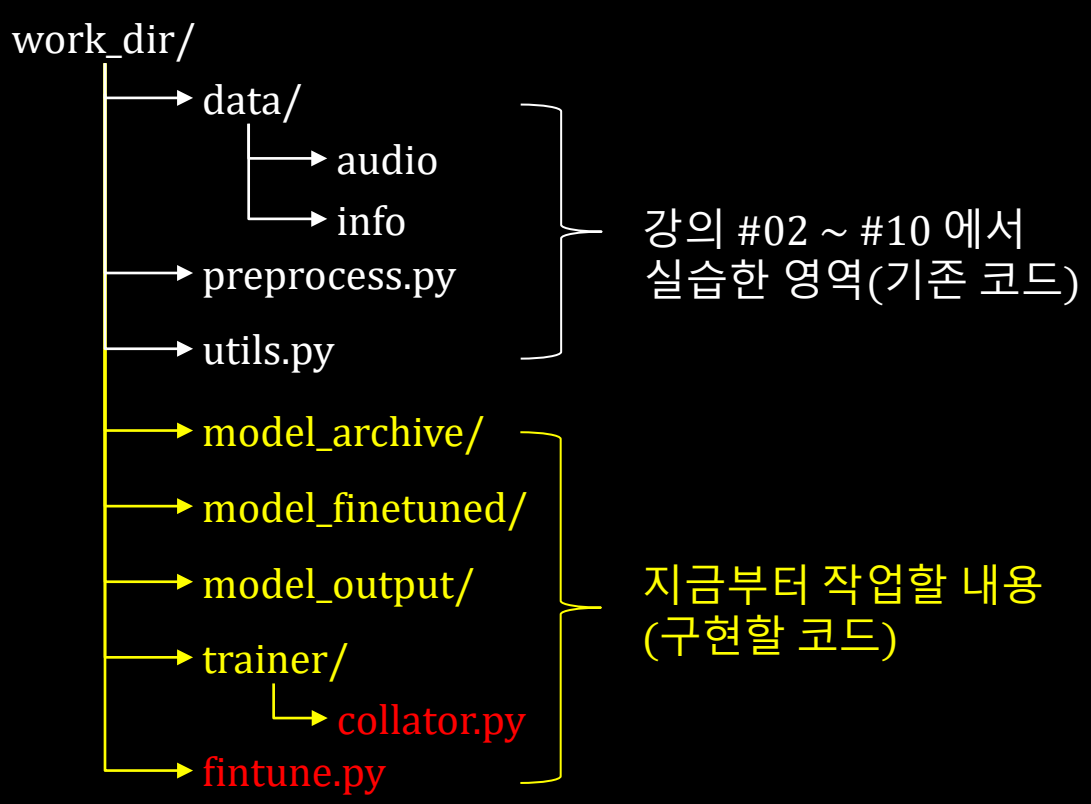
Implementation Approach

클래스로 구현

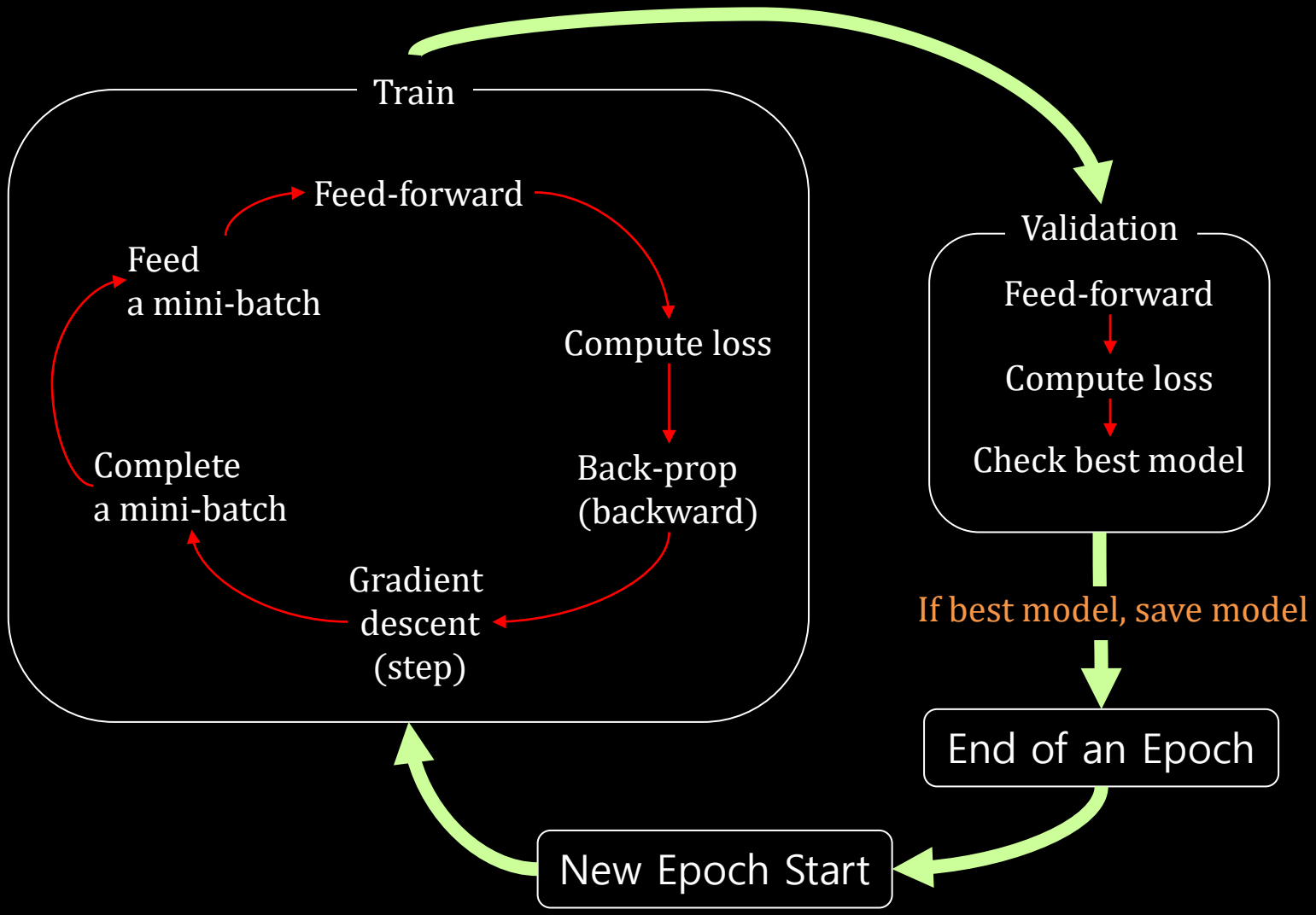
Huggingface에는 class 형태로 처리하지 않음.

서버에서 객체를 생성해서 항상 들고 있어야 해서... ㅠ

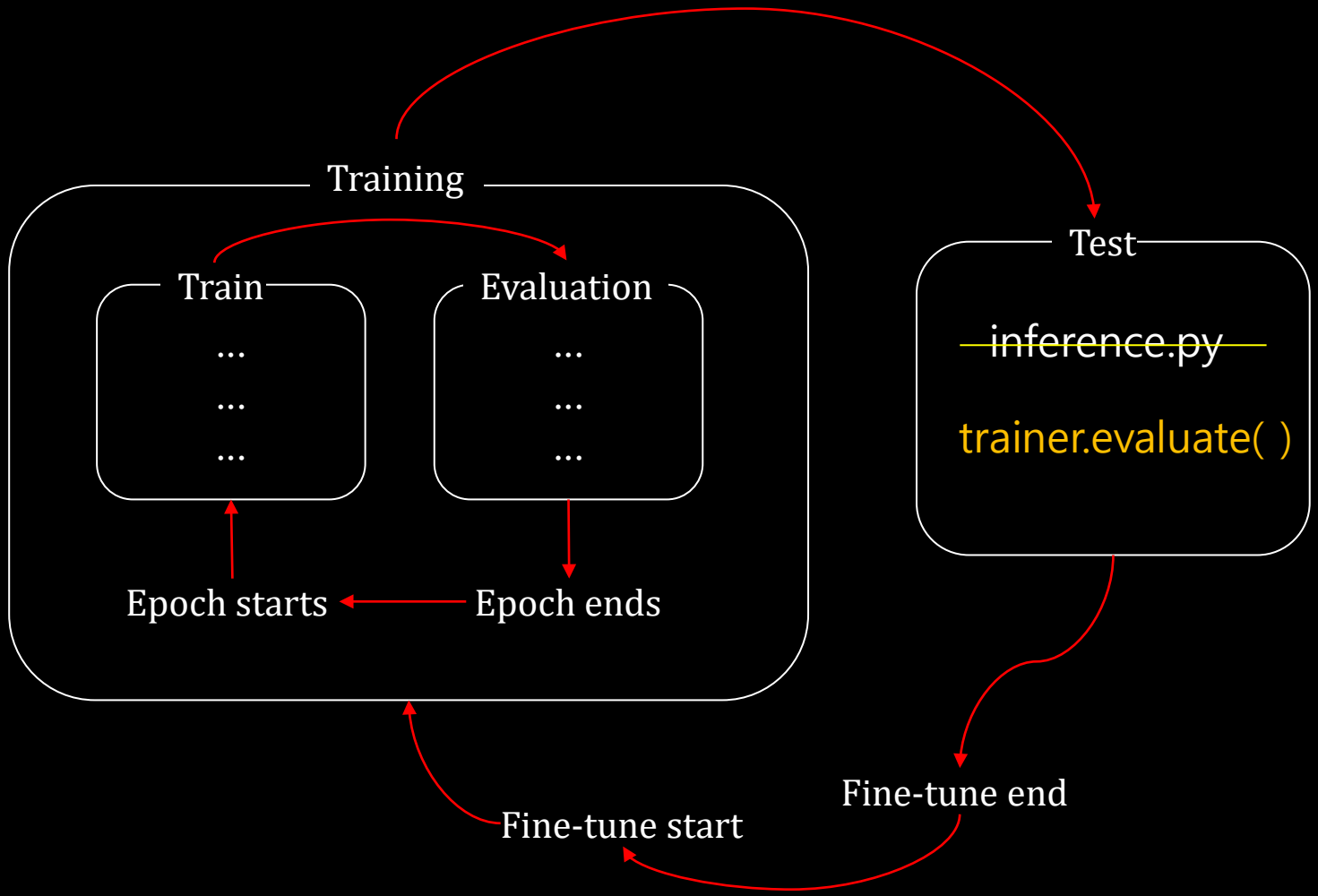
Workspace Structure



Structure for Train / Evaluation / Test



Fine-tuning 1 회 수행 프로세스



Dataset: Train / Evaluation / Test

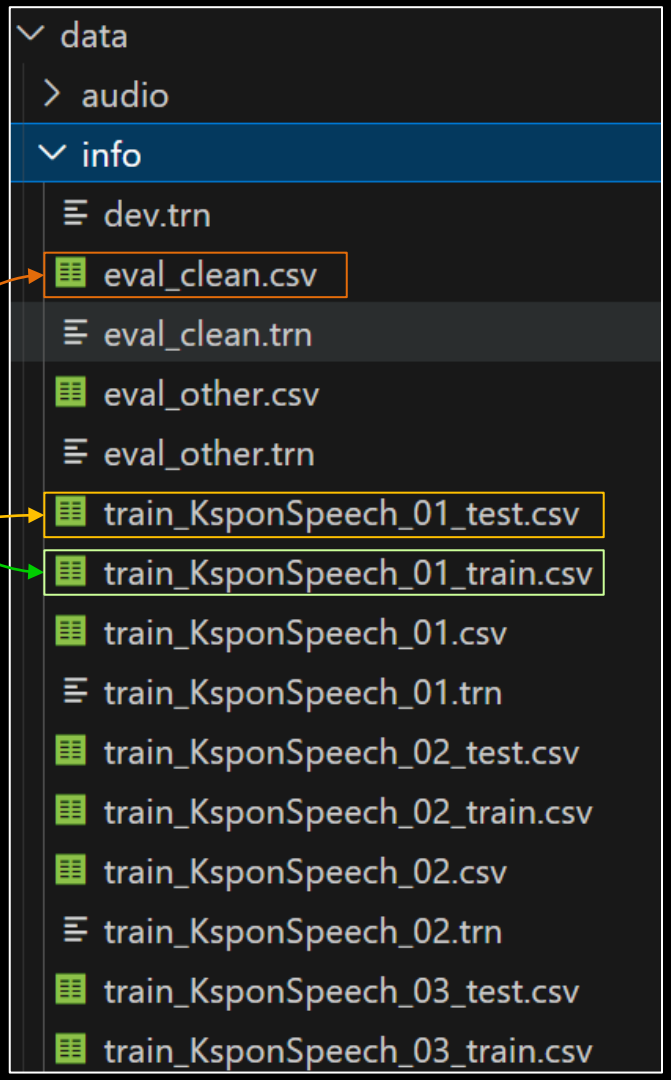
For each fine-tuning,

Train set
- train_KsponSpeech_0X_train.csv

Validation set
- train_KsponSpeech_0X_test.csv

Test set
- eval_clean.csv

Fixed dataset



finetune.py

```
def get_config():
```

```
class Trainer:

    def __init__(self, config) -> None:

    def load_dataset(self) -> DatasetDict:
        '''Build dataset containing train/valid/test sets'''

    def compute_metrics(self, pred) -> dict:
        '''Prepare evaluation matric (wer, cer, etc.)'''

    def prepare_dataset(self, batch):
        '''Get input_features with numpy array and sentence labels'''

    def process_dataset(self, dataset) -> tuple:
        ''' Process loaded dataset applying prepare_dataset'''

    def enforce_fine_tune_lang(self) -> None:
        '''Enforce fine-tune language'''

    def create_trainer(self, train, valid) -> None:
        '''Create seq2seq trainer '''

    def run(self) -> None:
        '''Run trainer'''
```

수고하셨습니다 ..^^..