과제 개요

- 발화에 대응하는 립리딩 영상과 음성을 pair로 구축하여 Audio와 Video를 기반으로 Text를 추정하는 모델 개발(Audio-Visual Speech Recognition)
- 문제 정의
  - 화자에 대한 Video + Audio 데이터를 기반으로 화자의 얼굴 혹은 입술 특징과 음성 특징을 융합하여 Text를 도출하는 문제
- 추진 배경
  - 다양한 유형의 노이즈가 있는 실제 환경에서 견고성을 달성하기 위해 음향 잡음에 의해 왜곡되지 않는 영상 정보를 활용하여 음향적으로 불리한 환경에서 중요한 역할 기대
  - 대화형 인공지능 등 대화체 커뮤니케이션 증가, 다수 화자의 비정형 회의 데이터에 대한 음성인식 시스템 등 다양한 분야에서 응용 가능
- 활용 가능 사례
  - 음성인식 개선
  - 청각장애인을 위한 서비스
  - 화자 식별
  - 자동 립싱크 생성

제한사항

- 문제 진행 시 규정
  - 사용 가능한 프레임워크는 pytorch로 제한함
  - Pretrained Weights의 사용을 허용하지 않음
  - 모델 학습 파라미터 50M 이하
  - 추론 시간 최대 30분 --> 추론 서버의 자원에 따라서 달라질 수 있음
- 데이터 사용 규정
  - 참가자는 문제 해결에 있어 주최 측이 제공한 데이터 외의 별도의 외부 데이터를 사용할 수 없음
- 재현 검증 절차 관련 규정
  - 참가자는 학습과 추론 결과를 재현할 수 있도록 별도의 수정 없이 재현 가능한 코드 각 1부(.py 형식), 학습된 가중치 파일 1부, 라이브러리 리스트, 설명 자료(README)를 제출하여야 함
  - 무작위 난수 등을 사용할 경우 필히 seed를 설정

평가 지표

- 단어오류율(WER), 편집거리 Levenshtein distance.

상금

- 상금 :
- 상금 설명 :

데이터 구성

- Train (13 GB)
    - Video : 10000개 mp4 파일
        - 설명 : 한국어 발화를 촬영한 파일, 문장단위로 구성
        - 파일명 포맷 : {FileName}.mp4
    - Audio : 10000개 wav 파일
        - 설명 : 한국어 발화를 녹음한 파일, 문장단위로 구성
        - 파일명 포멧 : {FileName}.wav
    - Label : 10000개 txt 파일
        - 파일명 포멧 : {FileName}.txt

- Test (0.1 GB)
    - Video : 100개 mp4 파일
        - 설명 : 한국어 발화를 촬영한 파일, 문장단위로 구성
        - 파일명 포맷 : {FileName}.mp4
    - Audio : 100개 wav 파일
        - 설명 : 한국어 발화를 녹음한 파일, 문장단위로 구성
        - 파일명 포멧 : {FileName}.wav
    - Label : 100개 txt 파일
        - 파일명 포멧 : {FileName}.txt

추가 데이터 구성

- Noise
    - 설명 : 학습에 사용될 Clean Audio에 믹싱할 여러 가지 Noise 파일
    - 파일명 포멧 : {FileName}.wav

코드 사용법

- 목적 : 베이스라인 코드를 실행하기위해 전처리, 학습, 추론 과정을 설명
- 환경 세팅
    - pytorch 설치
        - 실행 환경에 맞는 1.9 버전 pytorch 설치 (pytorch 설치 페이지)
        - 예) pip install torch==1.9.0
- 작업 폴더 구성

```
working_folder
├── base_builder
│   └── model_builder.py
├── base_model
│   └── base_model.py
├── checkpoint
│   └── checkpoint.py
├── dataloader
│   ├── augment.py
│   ├── data_loader.py
│   ├── feature.py
│   └── vocabulary.py
├── metric
│   ├── metric.py
│   └── wer_utils.py
├── dataset
│   ├── labels.csv
│   ├── Train.txt
│   └── Test.txt
├── preprocessing.py
├── vid2npy.py
├── train.py
├── train.yaml
├── requirements.txt
└── data_folder
```

전처리

- 목적 : 학습에 사용될 Video 데이터를 numpy 형태로 변환
- 데이터 구성

```
data_folder
├── Train
│   ├── Video
│   │   ├── file1.mp4
│   │   ├── file2.mp4
│   │   ├── ...
│   │   └── fileN.mp4
│   ├── Audio
│   │   ├── file1.wav
│   │   ├── file2.wav
│   │   ├── ...
│   │   └── fileN.wav
│   └── Label
│       ├── file1.txt
│       ├── file2.txt
│       ├── ...
│       └── fileN.txt
├── Test
│   ├── Video
│   │   ├── file1.mp4
│   │   ├── file2.mp4
│   │   ├── ...
│   │   └── fileK.mp4
│   ├── Audio
│   │   ├── file1.wav
│   │   ├── file2.wav
│   │   ├── ...
│   │   └── fileK.wav
│   └── Label
│       ├── file1.txt
│       ├── file2.txt
│       ├── ...
│       └── fileK.txt
├── Noise
    ├── file1.wav
    ├── file2.wav
    ├── ...
    └── fileM.wav
```

- 다음을 실행시켜 전처리 코드 실행

  python vid2npy.py --data_folder data_folder ₩

- 목적 : 학습에 사용될 Video, Audio, Text, Token 데이터를 짝을 맞추어 하나의 txt 파일로 각 데이터 주소를 저장

- 다음을 실행시켜 전처리 코드 실행

  python preprocessing.py --data_folder data_folder ₩
  --mode Train(or Test)

학습

- 목적 : 음성 기반 행동 생성 모델 학습
- 다음을 실행

  python train.py
- train.yaml에서 환경 설정