Wrapup_report

📌CV-16

📌SOTA

김동욱T7108

김재진T7127

박정욱T7153

이재건T7227

황은섭T7268

📌intro:

1️⃣ 기획 :

주제 선정 배경 요약

- 최근 온디바이스 AI가 주목받음에 따라, 한정된 VRAM 및 하드웨어 자원을 지닌 일반적인 디바이스 환경에서도 고성능 모델을 운용하기 위해서는 모델의 경량화와 최적화가 필수적입니다. 이에 따라, Audio understanding 벤치마크에서 베이스라인 모델의 정확도를 적절히 유지하면서도 모델 크기를 줄이고 추론 속도를 높이는 '작고 빠른 모델'에 대한 레서피를 설계하는 것이 주된 목표입니다.

문제 해결 방안

- 모델의 **효율적인 파라미터 구성**을 통해 **성능을 유지하면서도 연산 비용을 절감**하는 전략을 수립
- **경량화 기법 적용**: LoRA(Low-Rank Adaptation), 양자화(Quantization), 지식 distillation 등 다양한 기법을 활용하여 모델 최적화
- SALMONN에서 대부분의 파라미터를 차지하는 LLM 모델을 더 작고 빠르도록 설계된 LLM으로 교체.

- 적은 파라미터를 차지하는 Whisper v2는 조금 더 무겁지만 성능이 좋고 빠른 Whisper v3 turbo로 교체
- 기존 성능 유지를 위한 증강 적용

기대 효과

- 기존 모델 대비 연산 효율성이 향상되어 **실시간 서비스 적용 가능성 증가**
- 모델 경량화를 통한 **배포 비용 절감 및 다양한 환경에서의 활용 가능성 확대**

2️⃣ 팀원 역할 :

프로젝트를 진행하면서 개개인 별로 주도적으로 참여한 부분을 성과와 역량 중심으로 작성

▼ 이재건

모델 경량화 및 최적화를 중심으로 프로젝트에 기여했으며, **환경 세팅, 모델 업그레이드, MoE 적용, QLoRA 및 Unsloth 활용** 등 다양한 최적화 작업을 수행하였습니다.

1. **개발 환경 구축 및 초기 세팅**
    - **Docker 환경 세팅 및 초기 구성**
    - **SHM 메모리 문제 해결** → 모델 학습 시 메모리 부족 문제 해결
2. **모델 학습 및 최적화 문제 해결**
    - **Train 과정에서 발생한 문제 해결**
    - **BEATs 모델 개선 및 업그레이드(EAT 적용)**
    - **Deepseek 모델 분석 및 경량화 방향 연구**
3. **최신 경량화 기법 연구 및 적용**
    - **MoE(Mixture of Experts) 모델 연구 및 코드 적용**
    - **QLoRA 적용 및 경량화 실험**
    - **Unsloth 프레임워크 도입 및 최적화 적용**

핵심 성과 및 역량

- **경량화된 최신 AI 모델 적용** (BEATs, Deepseek, MoE, QLoRA, Unsloth)
- **최적화 기법 연구 및 실험** (QLoRA, 4bit 모델 적용)
- **환경 구축 및 성능 개선** (Docker 세팅, 메모리 문제 해결)

- **논문 및 최신 연구 탐색** (MoE, Deepseek 등)

프로젝트에서 **최신 모델 경량화 기법을 연구하고, 실제로 적용하여 최적화 성능을 개선하는 데 핵심적인 역할**을 수행하였습니다.

▼ 황은섭

모델 학습 및 코드 분석

- 모델을 학습하고 추론하는 과정에서, 불편하거나 추가하면 좋을 부분을 수정했습니다.
  - valid셋 나누기
  - 코드 관리
  - validation에서 asr, aac 측정하기
- 실험 관리

병목점 분석

- Beats에서 시간이 많이 걸리는 것을 확인하고, Beats를 EAT으로 바꾸려고 시도했습니다.
  - EAT의 구현이 잘못되었는지 성능이 낮았음.
  -

▼ 김동욱

1. **Base Model 분석**
   - Nota 측에서 제공된 베이스 모델 중 Qwen2 Audio에 대해 논문을 읽고 분석하였습니다.
   - 다른 베이스 모델인 Salmonn에서 각 부분에 대해 파라미터 수를 분석하여 latency와 메모리에 대해 어느 부분이 영향을 많이 줄지에 대해 분석하였습니다.
2. **경량화 기법 Search 및 적용**
   - 기존의 LoRA에 양자화까지 추가된 QLoRA를 기존의 모델에 적용시켰습니다.
   - 또한 모델의 latency를 줄이고자 salmonn에 적용된 llama3.2-3B에 MoE를 적용시키려 했습니다.
3. **Model Searching**

- salmonn에 적용된 Whisper-large v2대신에 성능이 더 좋아지고, 속도가 빨라진 Whisper-large-v3 turbo를 찾고 적용시켰습니다.

4. **EDA**
    - Base Model인 Salmonn의 파라미터 수를 분석하여 어느 부분을 우선적으로 경량화를 할지 분석하였습니다.

▼ 박정욱

1. **LLM 분석**
    - 텍스트 생성 시 토큰도 생성되는 문제 원인 분석
    - 텍스트를 생성하는 메커니즘 분석
    - 텍스트 생성과 관련된 Config 파라미터 분석
2. **데이터 전처리**
    - 데이터 구성 요소, task 분포를 파악하기 위해 EDA 수행
    - 데이터의 다양성을 위한 Audio Augmentation 리서치
3. **모델 최적화**
    - Whisper 아키텍쳐 분석
    - BEATs 없이 학습을 통한 경량화 시도

▼ 김재진

1. Baseline code 구동
    a. 제공된 서버에서 초기 구동을 위한 설정, 데이터셋 수정
2. Baseline Model 분석
    a. salmonn, beats의 아키텍쳐 분석
3. 테스트용 학습의 시간 절약을 위한 데이터셋 구축
    a. 2task만 학습시켜 적은 학습량으로도 새로운 기법이 적용된 모델의 테스트

3️⃣ 프로젝트 타임라인 :

주제 선정, 데이터 수집 및 전처리 등 해당 내용을 수행한 세부적인 기간과 활동 내용

프로젝트는 1월 중순부터 본격적으로 환경 세팅 및 데이터 전처리를 시작하여,오디오 입력 데이터 처리 → 모델 확장(BEATs, Deepseek, Whisper, EAT) → 경량화 기법(QLoRA,

MoE, Unsloth) 적용 →**최종 모델 검증 및 인퍼런스 문제 해결**의 순서로 진행되었습니다.

▼ 세부적인 진행절차(작업기록을 중심적으로)

1월 15일 ~ 1월 17일: 프로젝트 초기 세팅 및 데이터 전처리 준비

- **01.15**: 데이터셋 정리
- **01.17**:
  - `os.environ["RANK"]` 설정 이슈 해결
  - Docker를 활용한 **개발 환경 세팅 및 초기환경** 구성
  - **EDA**(Exploratory Data Analysis) 진행
  - 파일 경로(`path`) 문제 및 Requirements 용량 이슈 해결
  - 서버 시간 동기화 작업(Server time 맞추기)
  - SPIDEr 관련 및 평가지표 확인

1월 18일 ~ 1월 19일: 모델 학습 환경 문제 해결 및 검증

- **01.18**:
  - `salmonn` 모듈 파라미터 수 분석
  - 학습(train) 과정 중 발생하는 문제점들 해결
- **01.19**:
  - 데이터 로드 오류(Failed to load [경로]) 처리
  - **validation** 단계 점검

1월 20일: 오디오 입력 데이터 처리 및 QLoRA 적용 시도

- 오디오 input 데이터 처리 과정 정리
- **QLoRA 적용** 시도
- Stage1 학습 그래프 분석
- Train 과정 정리
- 미니 데이터셋 구성

1월 21일: 오디오 모델 확장 연구 (BEATs, Deepseek, Whisper)

- **BEATs 발전 과정** 리서치

- **Deepseek** 정보 수집
- 도커 사용 과정에서 `shm` 메모리 문제 해결
- Whisper 발전 과정 및 최신 버전 확인

1월 22일: 평가 프로세스 점검

- Evaluate 과정에서 발생하는 에러 및 평가 명령어 확인
- Task 인자 문제, QLoRA 에러 해결

1월 23일: 모델 업그레이드, MoE, QLoRA 문제 해결

- **BEATs** 업그레이드 모델(EAT) 적용
- Bottleneck 파악
- Evaluate 시 Text 문제 관련 에러 해결
- **MoE(Mixture of Experts)** 관련 논문 및 코드 찾기
- QLoRA 적용 문제, device 이슈 해결
- Test 과정 시간 분석
- Whisper에 LoRA 적용 검토
- 노타 1차 멘토링 정리

1월 24일 ~ 1월 29일: LLM Input 확인, DeepSeek·EAT 모델 적용

- **01.24**: LLM input 형식 확인
- **01.26**:
  - BEATs 인코더 선택 옵션 정리
  - **DeepSeek R1 모델** 적용
  - **MoE 모델** 적용
- **01.27**: LLM Loss 분석
- **01.28**: DeepSeek 논문 및 **경량화(lightweighting)** 아이디어 연구
- **01.29**:
  - EAT 적용 과정 정리
  - QLoRA TensorMismatch 문제 해결

2월 1일 ~ 2월 3일: 학습 세부 조정 및 오디오 증강 기법

- **02.01**:
  - EAT와 `accum_grad` 학습 이슈
  - 학습률(epoch) 초기화 문제 해결
  - 학습 돌리는 명령어 정리
- **02.02**: DeepSeek 코드 분석
- **02.03**:
  - **Audio augmentation** 기법 검토
  - 빠른 테스트용 데이터셋 및 task별 데이터셋 구성

2월 4일: Preprocessing, GPT 기반 증강 추천, Unsloth 적용

- BEATs, Whisper **preprocessing** 과정 정리
- GPT를 이용한 증강 기법 추천
- **Llama-3.2-3B-Instruct에 MoE 적용** 시도
- **Unsloth 코드** 적용 시작

2월 5일: 최적화 및 검증

- **LLaMa MoE** 추론 시간 측정
- **Unsloth 적용** 과정 정리
- validation에서 asr, aac 측정 진행
- 인퍼런스 시 `</s>` 반복 출력 문제 해결
- 증강 실험 진행
- 증강 적용 시 Loss NaN 문제 해결

2월 6일: Generate 파라미터·증강 문제 해결

- Generate 함수 파라미터 고찰
- Test에서 `stopping_criteria` 추가 검토
- Tmux 접속 환경 정리
- 증강 적용 시 Loss 발산 문제 분석
- 패딩 토큰을 eos 토큰으로 사용하는 이유 확인

2월 7일: Unsloth를 활용한 Qwen2.5-0.5B-Instruct 4bit 모델 적용

- **Unsloth Qwen2.5-0.5B-Instruct-bnb-4bit** 학습 및 정리

타임라인정리

Aa 이름	날짜	담당자	상태
개발세팅정리	@2025년 1월 15일 → 2025년 1월 17일		시작 전
Dataload 에러해결	@2025년 1월 15일 → 2025년 1월 19일		시작 전
오디오모델 확장연구	@2025년 1월 15일 → 2025년 1월 19일		시작 전
Qlora	@2025년 1월 20일 → 2025년 1월 21일		시작 전
평가체계안정화	@2025년 1월 20일 → 2025년 1월 24일		시작 전
Beat업그레이드→ EAT적용	@2025년 1월 20일 → 2025년 1월 24일		시작 전
Deepseek 모델적용&최적화	@2025년 1월 25일 → 2025년 1월 31일		시작 전
Deepseek MoE 적용	@2025년 1월 25일 → 2025년 1월 31일		시작 전
학습안정성 및 aug 적용	@2025년 2월 1일 → 2025년 2월 5일		시작 전
Unsloth 모델 적용	@2025년 2월 3일 → 2025년 2월 9일		시작 전
최종마무리	@2025년 2월 3일 → 2025년 2월 9일		시작 전

📌model/ research:

데이터셋:

▼ 데이터셋

- **Librispeech: CC-by-4.0**

  "LibriSpeech: an ASR corpus based on public domain audio books", Vassil Panayotov, Guoguo Chen, Daniel Povey and Sanjeev Khudanpur, ICASSP 2015 (pdf)

- **MusicNet: CC-by-4.0**

  "MusicNet: Compact Convolutional Neural Network for Real-time Background Music Detection", Reddy, Chandan KA, Vishak Gopa,

Harishchandra Dubey, Sergiy Matusevych, Ross Cutler, and Robert Aichner, 2021 (pdf)

- **Clotho: CC-by-4.0**

  "Clotho: An Audio Captioning Dataset", Konstantinos Drossos, Samuel Lipping, Tuomas Virtanen, ICASSP 2020 (pdf)

- **WavCaps: CC-by-4.0**

  "WavCaps: A ChatGPT-Assisted Weakly-Labelled Audio Captioning Dataset for Audio-Language Multimodal Research", Xinhao Mei, Chutong Meng, Haohe Liu, Qiuqiang Kong, Tom Ko, Chengqi Zhao, Mark D. Plumbley, Yuexian Zou, Wenwu Wang, IEEE 2024 (pdf)

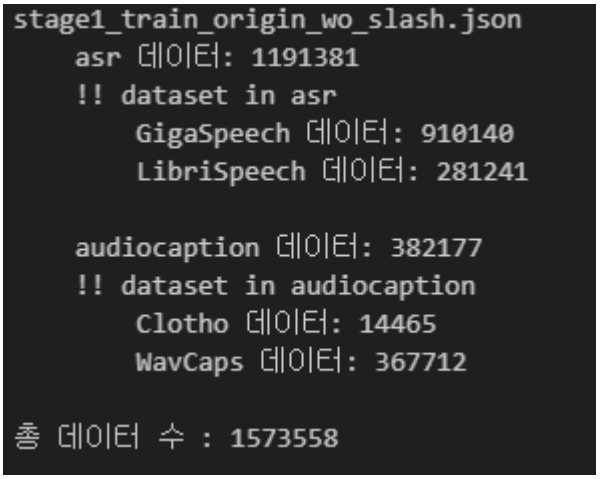
- **GigaSpeech: Apache-2.0**

  "GigaSpeech: An Evolving, Multi-domain ASR Corpus with 10,000 Hours of Transcribed Audio", Guoguo Chen, Shuzhou Chai, Guanbo Wang, Jiayu Du, Wei-Qiang Zhang, Chao Weng, Dan Su, Daniel Povey, Jan Trmal, Junbo Zhang, Mingjie Jin, Sanjeev Khudanpur, Shinji Watanabe, Shuaijiang Zhao, Wei Zou, Xiangang Li, Xuchen Yao, Yongqing Wang, Yujun Wang, Zhao You, Zhiyong Yan, 2021 (pdf)

- **AudioCaps: MIT**

  "AudioCaps: Generating Captions for Audios in The Wild", Chris Dongjoo Kim, Byeongchang Kim, Hyunmin Lee, Gunhee Kim, NAACL 2019 (pdf)

▼ stage, task별 데이터 이미지



```
stage2_train_origin_wo_slash.json
    asr 데이터: 481241
    !! dataset in asr
        GigaSpeech 데이터: 200000
        LibriSpeech 데이터: 281241

    phone_recognition 데이터: 281239
    !! dataset in phone_recognition
        LibriSpeech 데이터: 281239

    QA 데이터: 551759
    !! dataset in QA
        MusicNet 데이터: 320
        LibriSpeech 데이터: 281037
        WavCaps 데이터: 270402

    audiocaption_v2 데이터: 61112
    !! dataset in audiocaption_v2
        audiocaps_1m 데이터: 46647
        Clotho 데이터: 14465

    gender_recognition 데이터: 28539
    !! dataset in gender_recognition
        LibriSpeech 데이터: 28539

총 데이터 수 : 1403890
```

1️⃣ Model

- **Whisper**
  - 기존에 사용했던 Whisper-large v2 대신에 Whisper-Large v3 turbo를 사용했습니다. Whisper-large v2보다 성능과 속도 측면에서 이점이 있고, 실험에서도 이점을 보여주어 채택했습니다.
- **Beats**
  - 속도 개선을 목적으로 기존의 사용하던 Beats를 제외하고 실험해보았습니다. 이를 통해 Beats가 정말 모델 성능에 유의미한지 파악하고 메모리 사용량과 latency를 감소시키려 했습니다.
  - 병목점을 확인했을 때 Beats가 시간을 제일 많이 차지했습니다. 따라서 Beats보다 성능과 속도가 빠른 EAT라는 모델을 사용하여 실험을 해보았습니다.

    - 하지만 Beats가 시간을 제일 많이 차지하는 이유는 데이터를 전처리하는 시간까지 포함되기 때문이었고, EAT은 효과가 뛰어나지 않아 Beats를 그대로 사용하였습니다.

- **LLM**
    - 가장 파라미터 수가 많은 부분이였기에 중점적인 경량화할 부분이였습니다.
    - 우선 DeepSeek-MoE 모델로 LLM을 사용하여 실험을 진행해보려 했지만, 모델이 GPU에 다 올라가지 않아 실험을 진행해보지는 못했습니다.
    - DeepSeek을 Teacher모델로 지식 증류한 Qwen모델인 DeepSeek-R1-Distill-Qwen-1.5B을 사용하여 실험했습니다.
    - 마지막으로는 unsloth에 제공된 Llama계열(1B, 3B, 8B)와 Qwen2.5 0.5B를 실험했습니다. 기존의 모델에 비해 LoRA, Quantization 등 여러 경량화 기법이 적용되어 속도와 메모리 측면에서 이점이 있을거라 생각하여 실험했습니다.

    ▼ (LLM 실험한거 정리해봤습니다)

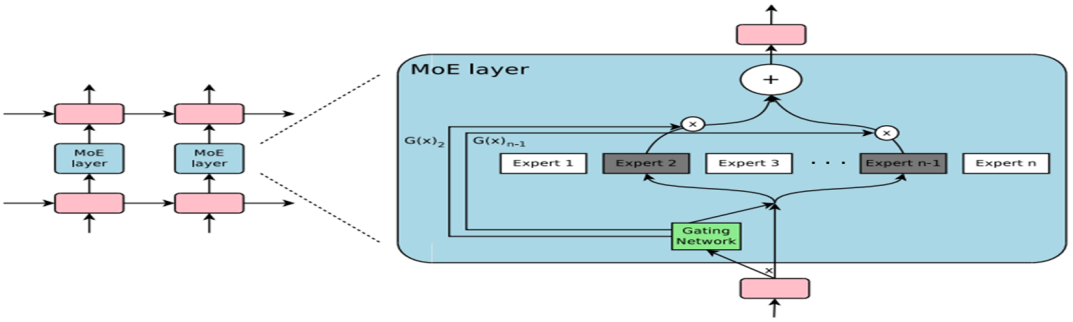
        - 성능 지표, 메모리 모두 측정한 모델
            - Llama-3.2-3B-Instruct: 베이스라인 모델
            - Llama-3.2-3B-Instruct_qlora: 베이스라인 모델에 QLORA 적용
            - DeepSeek-R1-Distill-Qwen-1.5B: DeepSeek R1 모델로 교체
            - Llama-3.2-3B-Instruct-bnb-4bit: Unsloth 모델로 교체
        - 둘 중 하나만 측정한 모델
            - Llama-3.2-1B-Instruct_qlora: 베이스라인의 3B → 1B로 교체 후 QLORA 적용(메모리 X)
            - Llama-3.2-1B-Instruct-bnb-4bit: Unsloth 모델로 교체(성능 지표X)
            - Qwen-2.5-0.5B-Instruct: Unsloth 모델로 교체(성능 지표X)
        - 측정 실패한 모델
            - deepseek-moe-16b-base: DeepSeek MoE 모델로 교체하려했으나 실패

2️⃣ 경량화 기법

- **QLoRA**

- 기존에 제공된 모델에 있는 LoRA대신에 double Quantization까지 적용된 QLoRA를 사용하였습니다. 기존의 float16자료형을 int4로 저장하기에 메모리 효율성 측면에서 이점을 보았습니다.

- **MoE**



  - LLM부분에 여러 전문가를 두어 모델의 latency를 줄이고자 Llama-MoE 모델을 사용하여 실험을 진행했습니다.
  - 하지만 기존에 제공되어 있는 Llama-MoE모델은 기존의 Llama-3B보다 컸기에 기존의 Llama-3B에 직접 MoE를 적용해보는 시도를 했습니다.

- **Knowledge Distillation**
  - LLM을 좀 더 작은 모델로 사용하고, 성능은 유지하기 위해 DeepSeek을 Teacher 모델로 Knowledge Distillation을 진행한 Qwen 모델로 교체를 하여 실험했습니다.
  - 하지만 학습 결과 기존의 모델에 비해 성능 하락이 심했습니다.

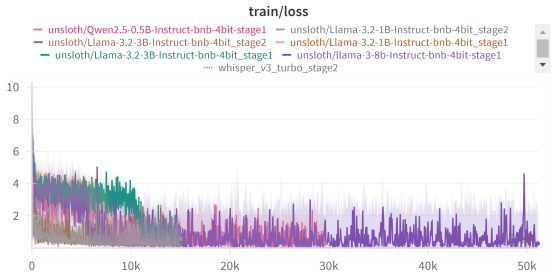
3️⃣ 최종적용모델 - Unsloth Llama-3.2-1B-Instruct-bnb-4bit

적용이유

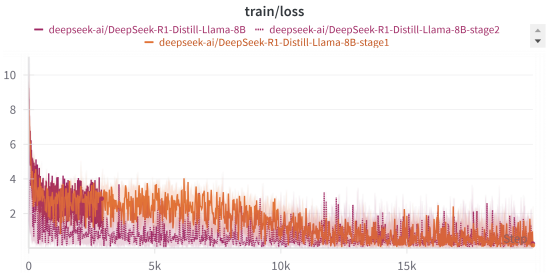
1. **Unsloth 모델 자체의 우수성**
   - Deepseek에서 직접 커스텀하여 개발한 모델로, 기존의 3계층 Dense 구조를 MoE(Mixture of Experts) 구조로 변경해 **경량화와 효율성을 극대화**했다고 알려져 있습니다.
   - 이러한 아키텍처 덕분에 상대적으로 적은 파라미터로도 만족스러운 성능을 낼 수 있어 Unsloth 모델을 최종적으로 선택했습니다.
2. **여러 버전 중 1B 모델 선정**

- Unsloth 모델은 0.5B부터 14B까지 다양한 크기의 모델을 제공하지만, 우리 팀은 **1B 모델**을 최종적으로 선택하였습니다. 그 이유는 다음과 같습니다:
    1. **추론 속도**
        - Wandb를 통해 비교했을 때, 1B 모델이 **가장 빠른 추론 속도**를 보였습니다.
    2. **성능 유지**
        - 1B라는 상대적으로 작은 모델임에도 불구하고, **성능이 큰 모델에 비해 크게 떨어지지 않았습니다.**
    3. **리소스 절감**
        - 모델 크기가 작으므로 **GPU 메모리나 서버 비용 부담이 적어**, 실제 서비스 환경에서 **더 쉽고 경제적으로 운영**할 수 있습니다.

위와 같은 이유로 Unsloth **Llama-3.2-1B-Instruct-bnb-4bit** 모델을 최종적으로 채택하였습니다.



unsloth-train/loss



deepseek_train_loss

→ unsloth가 deepseek에 비해 70% 경량화된 모델임에도 불구하고 train/loss 값은 비슷하게 나옵니다.

📌result/ conclusion:

1️⃣ 후속 개발 및 연구방향 제시.

1. **모델 크기 축소 및 성능 유지**
    - 지식 증류(KD), Quantization, Pruning 활용

- 핵심 모듈만 로딩하는 Modular 방식 도입

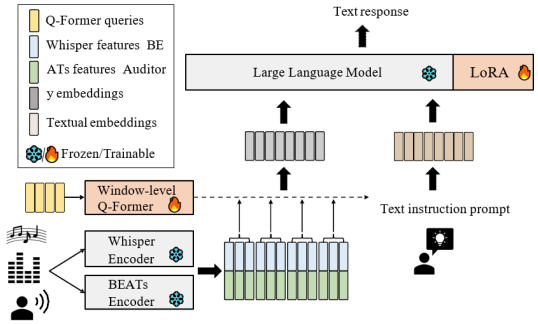
2. **모델 아키텍처 커스터마이징**
   - Prefix Tuning 적용
   - 토큰화 방식 개선 및 레이어 구조 최적화
3. **데이터 측면 개선**
   - 데이터 보강(Augmentation) 및 도메인 특화 데이터 확보
   - 사용자 입력 패턴 반영한 미세 조정(Fine-tuning)
4. **학습 파이프라인 최적화**
   - Mixed Precision, Dynamic Gradient Clipping 적용
   - 멀티태스킹 학습으로 AAC/ASR 성능 동시 개선
5. **데이터 전처리 함수 통일**
   - Beats과 Whisper가 입력으로 받는 spectrogram을 하나의 함수로 통일
6. **추가 연구 아이디어**
   - 온라인 학습 및 지속적 학습(Continual Learning)
   - 그래프·지식 기반 음성 후처리 연구
   - TTS 기반 합성 데이터 활용 및 Transducer 기반 접근 비교

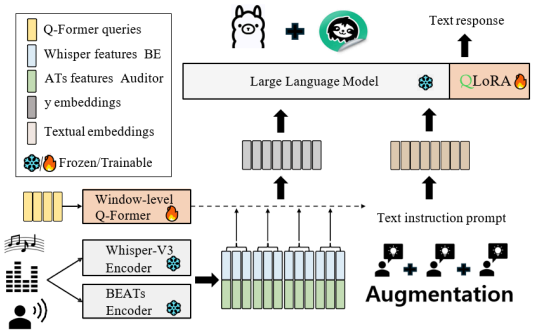
목표: 모델 경량화 + 성능 유지 + 도메인 특화 최적화 🚀

📌(필수 작성) 별첨.

서비스 아키텍쳐

기본아키텍쳐 변경아키텍쳐





📌개인회고

프로젝트 달성도와 완성도에 대한 자체적인 평가 의견과 느낀점, 추후 어떻게 발전시키면 좋을 지에 대한 의견

▼ 이재건 - 자체 평가 의견 및 느낀점

자체 평가 의견

- **환경 세팅 및 모델 발전 과정**에서 많은 시간을 투자하고, 각종 에러와 디버깅을 해결하며 프로젝트 전반에 기여하였습니다.
- 다양한 모델(Deepseek, BEATs, EAT, MoE, QLoRA 등)을 파악하고 적용하려는 시도에 적극적이었지만, 개인 사정으로 인해 모든 부분을 완벽하게 마무리하지 못한 점이 아쉽습니다.
- 그럼에도 프로젝트 진행 과정에서 **성과를 도출**하고, 꾸준한 시도와 시행착오를 통해 **전진**했다는 점에서 긍정적인 평가를 내리고 싶습니다.

느낀 점

- **도메인 이해의 중요성**: 단순 코딩 실력보다 AI 모델의 작동 원리와 파라미터(bos_token_id, eos_token 등)에 대한 심층적인 이해가 더 중요하다는 것을 깨달았습니다. 특히 CV 트랙 위주로 경험을 쌓다 보니, NLP 관련 개념을 추가로 학습해야 함을 절감했습니다.
- **GPU 리소스 부족 문제 체감**: GPU가 2개뿐이라서 학습 및 실험에 많은 시간이 소모되었습니다. GPU 자원이 많을수록 학습 시간 단축뿐 아니라 **디버깅 속도**에도 큰 도움이 된다는 사실을 직접 체감했습니다.
- **기본 모델(Baseline)의 중요성**: 기본 모델을 깊이 이해하면, 추후 다양한 문제 상황에서 **유연하게 대처**할 수 있음을 실감했습니다.

추후 발전 방향

1. **도메인 지식 확장**: NLP·오디오·비전 등 다방면의 AI 분야를 꾸준히 학습하여, 모델 내부 구조와 파라미터를 보다 정확히 이해하고 활용할 수 있는 역량을 갖추고자 합니다.
2. **하드웨어 자원 확보 및 효율적 활용**: GPU 리소스가 제한적인 상황에서 효율적 경량화·분산 학습 기법을 적극적으로 도입하고, 추후 가능하다면 **GPU 서버 확장**이나 클라우드 리소스 활용 방안을 모색해보겠습니다.
3. **기본 모델(Baseline) 심화 학습**: 다양한 모델을 접목하기 전에, 프로젝트에서 사용되는 **베이스라인 모델**을 심도 있게 분석·최적화하여 향후 문제 발생 시 빠르게 해결하는 역량을 키
4. **협업 및 커뮤니티 참여**: 공개된 모델이나 라이브러리, 연구 커뮤니티 등을 적극적으로 활용하고 공유함으로써, 최신 트렌드를 빠르게 파악하고 **지식 교류**를 활성화합니다.

▼ 김동욱 - 자체 평가 의견 및 느낀점

자체 평가

- 기존의 경혐했었던 프로젝트와 많이 달라 어려움을 겪었다. 기존에는 데이터도 주로 이미지만 다루었고, 단순히 성능을 높이면 되는 프로젝트였다. 하지만 이번 프로젝트는 성능도 중요하지만 경량화가 주제였기에 경량화와 성능의 tradeoff를 다루기 힘들었던 것 같다. 또한 이전과 달리 새로운 도메인 데이터를 다루었기에 많이 해매었다.
- 데이터 다운로드부터 여러 문제가 있어 일정이 많이 미뤄지고 많은 실험을 해보지 못한 점이 굉장히 아쉬운 것 같다.
- 그리고 경량화 기법으로 여러 가지를 생각하였지만 여러 이유로 인해 모든 것을 실험해보지 못한 점이 아쉬운 부분이였다.

느낀점

- 이번 프로젝트를 통해 특정 데이터나 도메인만 공부하는 것이 아니라 다른 도메인에 대해서도 기본적으로라도 공부를 해야할 필요성을 느꼈다.
- 이전과 달리 서버 1개와 GPU 2개를 제공받아 사용했다. 또한 학습에도 굉장히 많은 시간이 소요되었기에 자원의 중요성을 다시 한번 느꼈다.

▼ 황은섭

달성도, 완성도에 대한 평가 의견

- 모델 구조에 직접적으로 관여하여 효과를 내려는 시도가 실패해 아쉬웠습니다.

- 실험적으로는 속도와 메모리가 매우 낮아졌기에 목적은 80%정도 달성했다고 생각합니다.
- 노타가 원하던 기존의 성능을 유지하는 부분은 아직 최종 학습이 끝나지 않아 판단하기 어렵습니다.

느낀 점

- LLM와 음성 모델에 대한 지식이 부족해 어려움을 많이 겪었습니다. 주의했어야할 부분을 신경쓰지 못해 뒤늦게 해결하게 되어 아쉬움이 많았는데, 이런 경험으로 도메인 지식이 왜 중요한지 느낄 수 있었습니다. 이론과 베이스 코드의 이해가 바탕이 되어야 함을 알았습니다.
- 다양하고 많은 시도를 하지 못한 것 같아서 아쉬움이 남습니다.
- 프로젝트를 하면서 다른 도메인에서도 경험을 쌓아 좋았습니다. 경량화 및 최적화, 음성 모델, 언어 모델, Multimodal에 대해 작지만 소중한 지식을 얻을 수 있었습니다.

발전 의견

- Beats를 더 발전된 모델인 EAT으로 성공적으로 교체한다면 성능 개선이 이루어지리라 생각합니다.
- raw데이터가 각각 다른 함수로 만들어진 spectrogram이 되어 whisper와 Beats에게 다르게 들어가는데, 같은 spectrogram을 인식할 수 있도록 두 인코더를 수정할 수 있다면 더 빠른 속도를 기대할 수 있을 것입니다.

▼ 박정욱

자체 평가 의견

- 오디오, LLM 도메인에 대한 지식이 부족하다보니 어디서부터, 어떻게, 무엇을 해야할지 바로 아이디어를 내지 못했고, 이전 프로젝트와 비교해서 빠르게 진행되지 못했습니다.
- 빠르게 진행되지 못하다보니, 아이디어는 많이 나왔지만 시간이 부족해 다양한 실험을 하지 못 했던 것이 아쉽습니다.
- 하지만 이런 상황에서도 LLM 교체, 경량화 기법, 증강, 최적화 관련 라이브러리 등 역할 분배를 통해 다양한 방법들을 조사하고, 모두 종합하여 최종 모델을 선택할 수 있었다는 점에서, 높은 완성도를 달성했다고 생각합니다.

느낀점

- 현재는 CV, NLP, RecSys 구분 없이 모두 LLM이 사용되기 때문에 언젠가는 LLM을 공부해야한다는 생각을 하고 있었고, 이번 프로젝트에서 LLM을 실험할 수 있는 기회를 얻게 되어 뜻깊었습니다.
- 프로젝트를 통해 LLM의 주요 개념(토큰화, 텍스트 디코딩, Loss)을 공부하면서 한 층 더 성장했다는 느낌을 받았습니다.
- 팀원들과 협업을 하면서, 혼자서 진행했다면 절대 이 정도 분량의 개념을 알지 못하고 또한 성과를 내지 못 했을 것입니다. 협업의 중요성을 다시 한 번 생각할 수 있었고, 부스트캠프를 하기 전과 비교했을 때 인식이 많이 바뀌게 되었습니다.

발전할 점

- 처음 보는 도메인에 적응하고 대처하면서 지금 돌이켜봤을 때 아쉬웠던 점이 많았습니다. 모델의 학습 파이프라인에 대한 전반적인 이해, 도메인에 대한 이해를 바탕으로 접근한다면, 앞으로 새로운 도메인의 실험을 하더라도 빠르게 진행될 수 있을 거라 생각합니다.

▼ 김재진

- 자체 평가 의견
  - LLM 계열 도메인 지식의 부족으로 오는 문제점들을 마주했습니다. 간단한것에 지연되는 시간, 같은 범주의 문제임에도 개별적으로 확인해 시간이 더 지연되는 것 같았습니다.
  - 경량화 관련, 많은 기법들에 대한 아이디어와 실험이 있었지만, 최종적으로 그 아이디어의 실현을 직접적으로 한것이 아닌게 아쉬웠습니다.
  - 맡은 역활에 대해 미흡한 모습이 많았다고 생각이 듦니다.
- 느낌점
  - 새로운 도메인에서 작업을 할 때 해당 도메인의 기본 지식을 최대한 미리 학습해야 함을 느꼈습니다. 경우에 따라 다르지만 각 문제 발생시 지연되는 시간을 고려했을 때 미리 학습하는것이 나았을거란 의견입니다.
  - 아이디어와 구현, 직접적인 적용과 배포된 모델 사이의 시간적인 trade-off로 직접적인 구현을 많이 하지 못하는 경우가 많았습니다. 추후 프로젝트를 진행함에 있어 초반이라도 밀도있게 계획하고 일정을 더 세밀하게 계획해야하는걸 다시금 깨달았습니다.
  - 최신 경향에 대한 파악이 미진해, 최신 트렌드인 아이디어들을 찾는데 생각보다 시간이 걸렸다고 생각합니다. 어떤것들이 hot한지 꾸준히 들여다 보는 습관을 들여야 나중에라도 편할것 같습니다.

    - 프로젝트를 진행하면서, 존재하는 role들의 상호작용들을 생각해보고 이해해서, 어떤 역을 맡던 프로젝트 진행에 효율적인 도움이 되도록 강구해야겠습니다.
- 추후 발전 방향
    - 이번 프로젝트에선 경량화를 위해 상대적으로 적은 파라미터 수로 비슷한 성능을 내어 연산수를 줄임으로써, 메모리, latency를 줄이는 방향으로 적용해 나아갔습니다. 이러한 방식 이외에도 커널이나 디바이스 최적화, Flash-Attention과 같이 연산차제의 최적화 기법들을 적용하여 더 이상적인 최적화가 가능할것으로 보입니다.