boostcamp

오디오 언어모델의 경량 모델링 레서피 탐구 - NOTA

CV-16조(NOTA-03)

김동욱 T7108, 김재진 T7127, 박정욱 T7153, 이재건 T7227, 황은섭 T7268

boostcamp ai tech

INDEX

1. Intro

1. 팀원 소개
2. 프로젝트 방향
3. 프로젝트 타임라인

팀원 소개

황은섭

- 모델 학습 및 실험 관리
- BaseLine 코드 및 모델 분석
- Dataset 구축

이재건

- 개발 환경 구축 및 초기 세팅
- 모델 학습 및 실험 관리
- 경량화 기법 및 모델 서칭

김재진

- 개발 환경 구축 및 초기 세팅
- BaseLine 코드 및 모델 분석
- Dataset 구축

김동욱

- EDA
- 경량화 기법 및 모델 서칭
- BaseLine 코드 및 모델 분석

박정욱

- EDA
- 오디오 증강 리서치
- BaseLine 코드 및 모델 분석

프로젝트 방향

프로젝트 목표

- 온디바이스 AI 환경에서도 **고성능 모델을 운용할 수 있도록 경량화 및 최적화 수행**
- Audio-Understanding 벤치마크에서 **정확도를 유지하면서도 모델 크기를 줄이고 추론 속도를 향상**

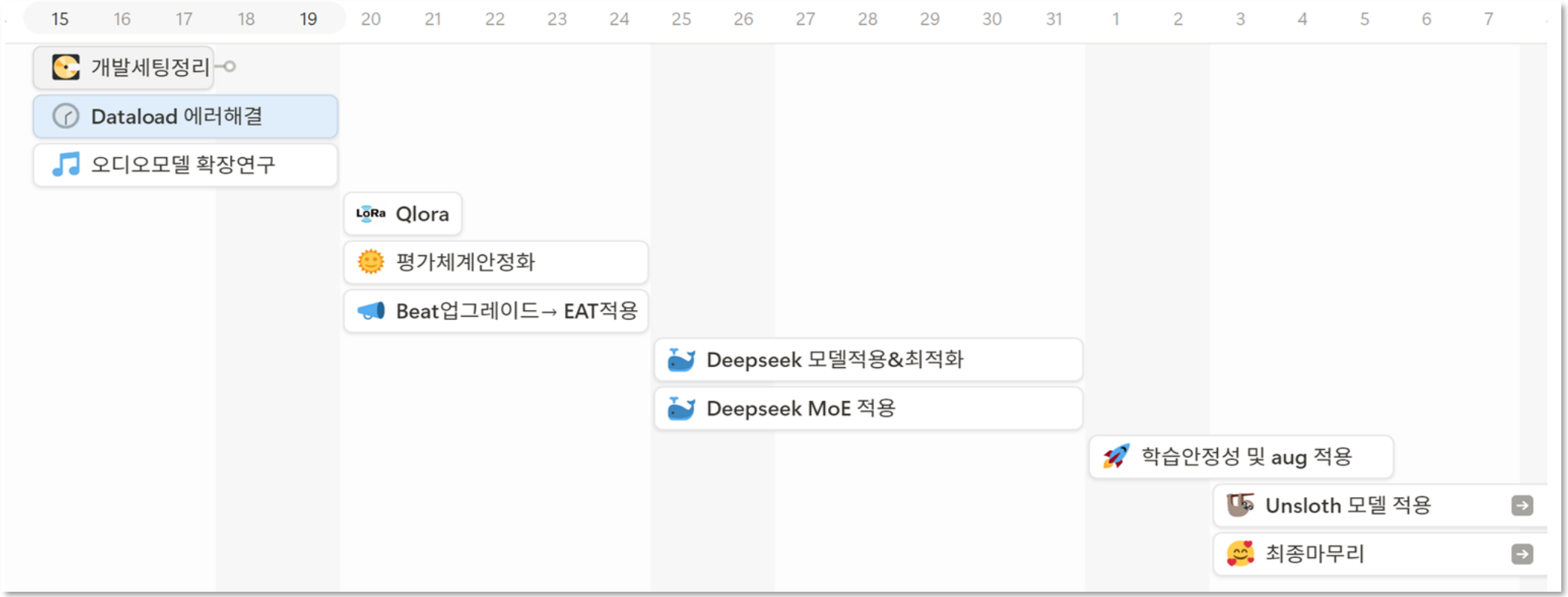
문제 해결 방안

- LoRA, Quantization, Knowledge Distillation 등의 **경량화 기법 적용**
- 기존의 모델보다 **빠르고 작은 모델 사용**
- 성능 유지를 위해 **Audio 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**

프로젝트 타임라인

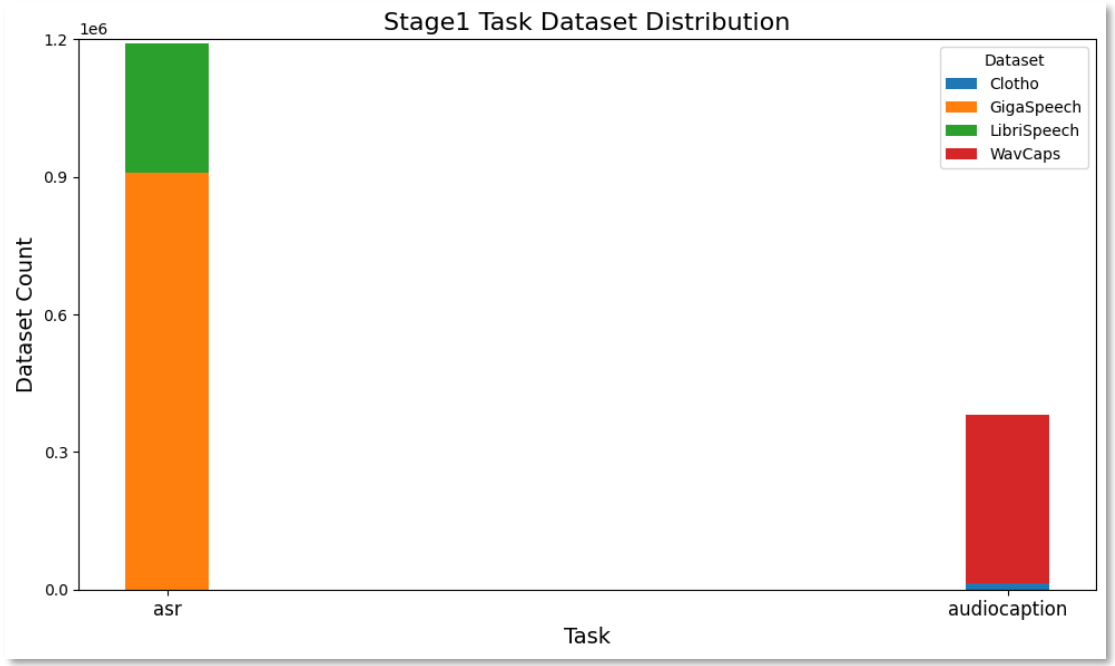


초기 세팅 (1/15 - 1/17) -> **모델 최적화** (1/18 - 1/26) -> **학습 안정화 및 성능 개선** (1/27 - 2/6) -> **최종 마무리** (2/7 - 2/10)

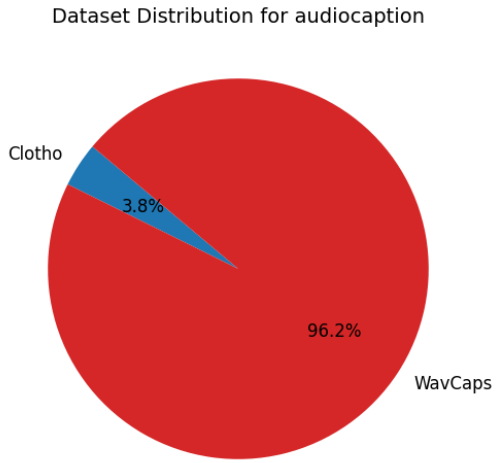
2. Model / Research

1. 데이터셋
2. 모델
3. 경량화 기법
4. 증강 기법
5. 최종 적용 모델

데이터셋

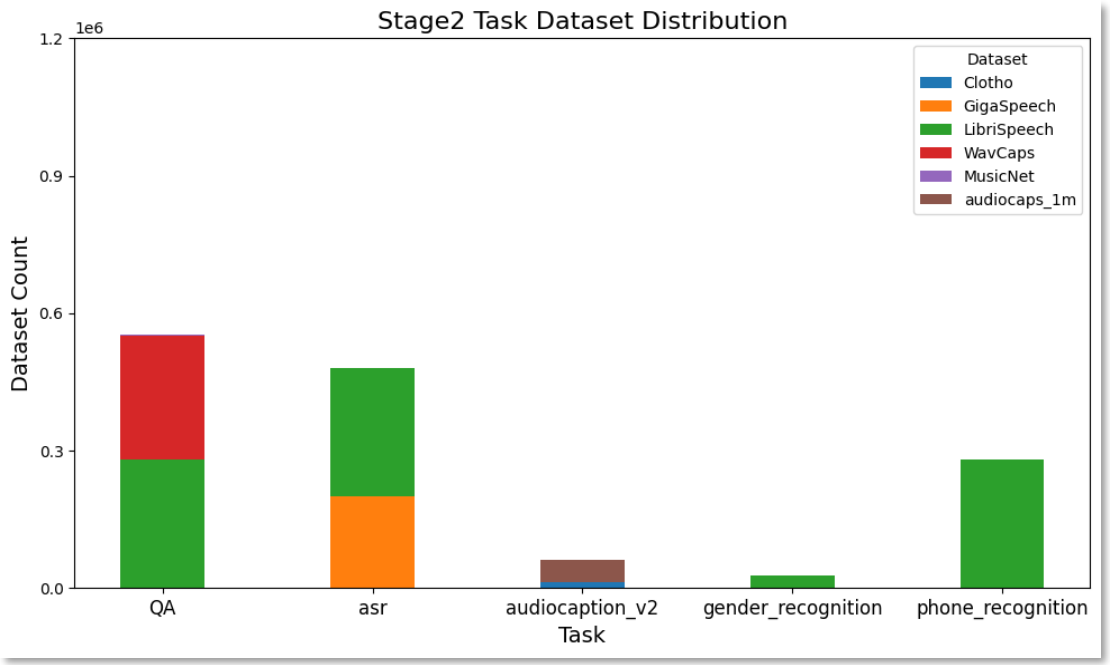


- Stage1 / task -

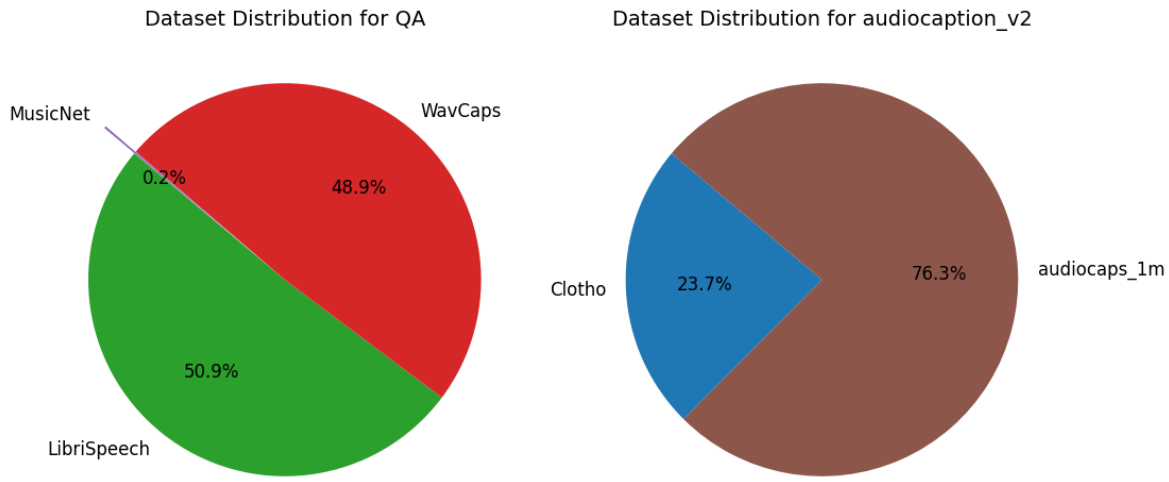


- Audiocaption task pie chart -

데이터셋

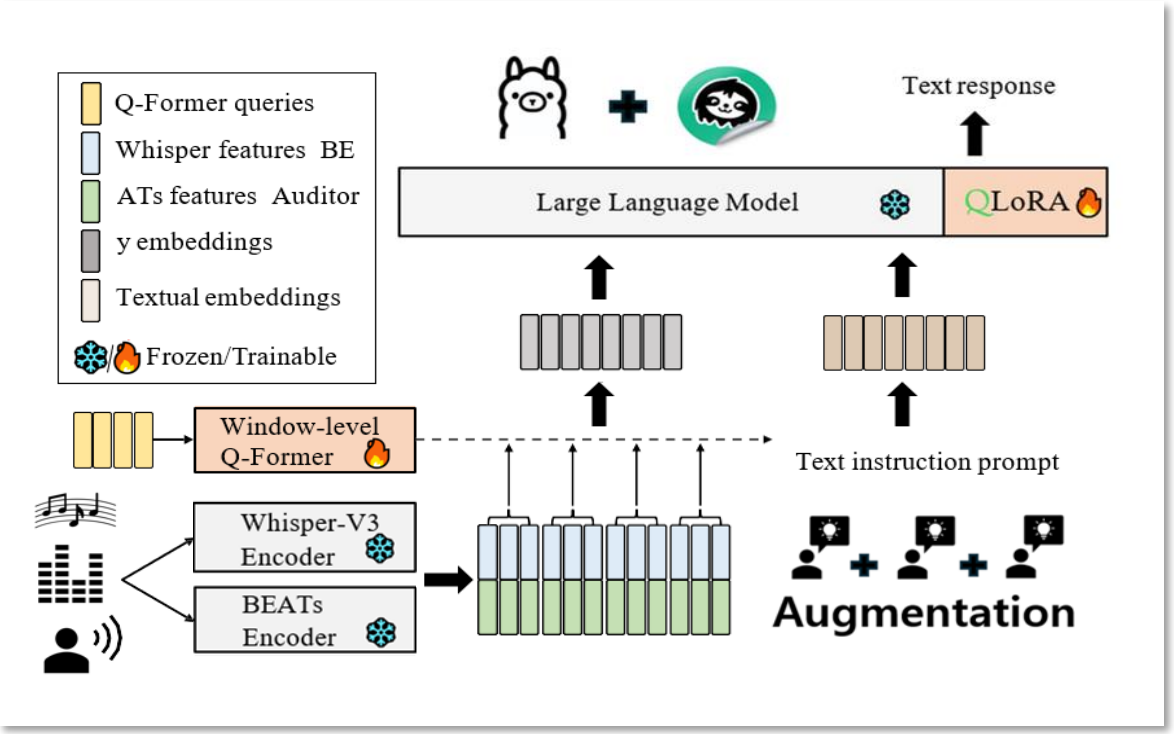


- Stage2 / task -



- QA, Audiocaption_v2 pie chart -

모델



- 아키텍처 -

Whisper
- 속도 측면에서 향상된 Whisper-large-v3 turbo로 실험

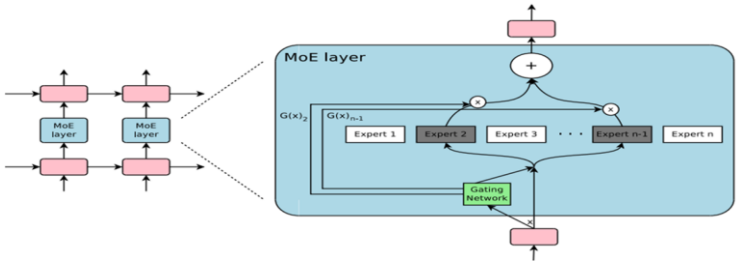
BEATs
- 경량화 및 속도 개선을 위해 BEATs제거 및 EAT로 교체하여 실험

LLM
- Unsloth의 llama 계열(1B, 3B, 8B)와 Qwen 2.5-0.5B로 실험

경량화 기법

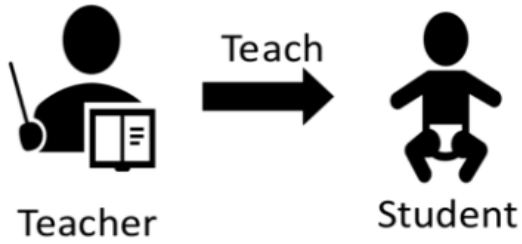
LoRA + Quantization

- 메모리 사용을 줄이기 위해 양자화가 추가된 LoRA 적용
- 메모리 효율성 측면에서 이점



Mixture of Experts

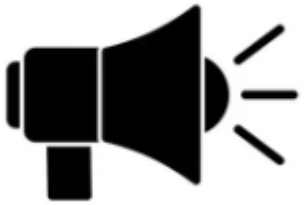
- Llama-MoE 모델 사용
- 몇 개의 Expert만 활성화하여 latency 감소와 메모리 감소를 기대



Distillation

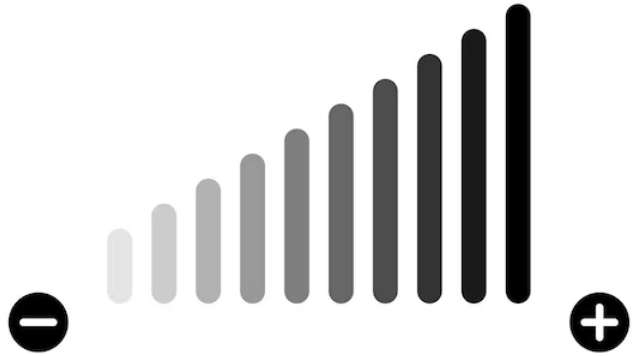
- DeepSeek R1을 Teacher모델로 학습한 llama 모델 실험
- BaseModel보다 적은 메모리 사용량으로 더 나은 성능을 기대

증강 기법



Add Noise

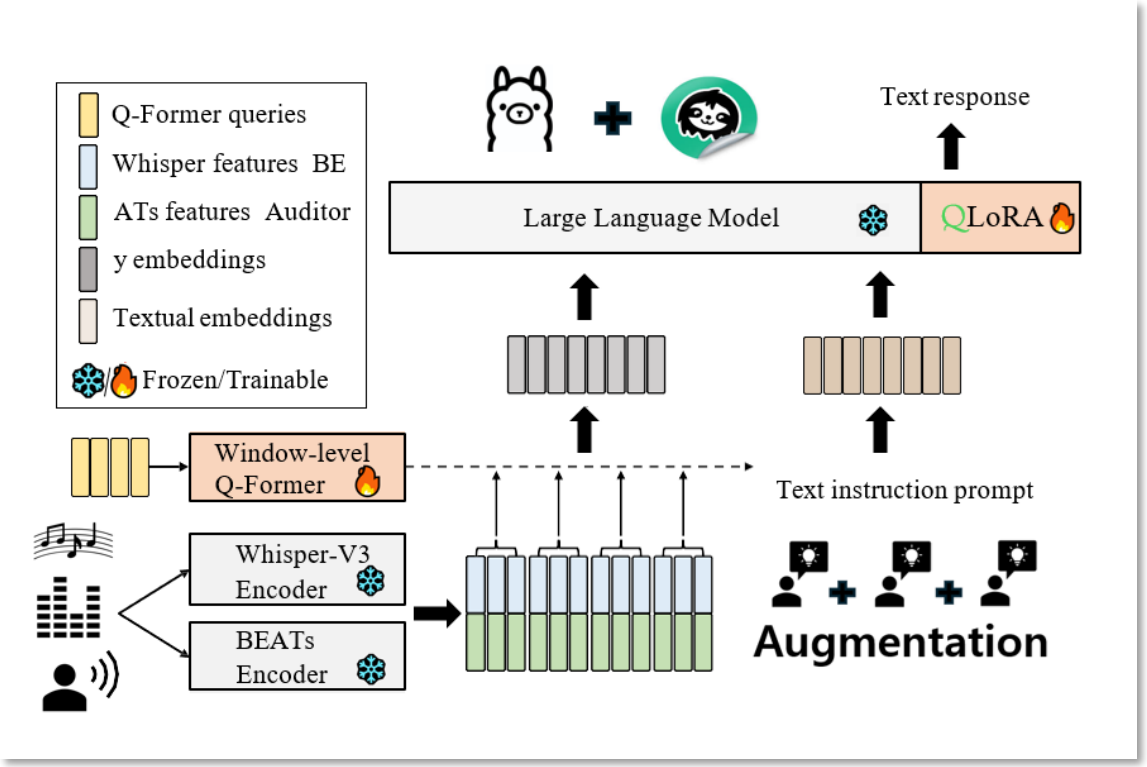
• 오디오에 노이즈를 적용해 현실적인 음성 패턴 추가



Gain Control

• 오디오의 볼륨을 다르게 적용해 다양한 음성 패턴 추가

최종 적용 기법



- 최종 아키텍처 -

최종 적용 모델

LLM: Unsloth Llama-3.2-1B-Instruct-bnb-4bit + QLoRA

Audio Encoder: Whisper-Large-v3 turbo, BEATs

Unsloth 모델 선택 이유

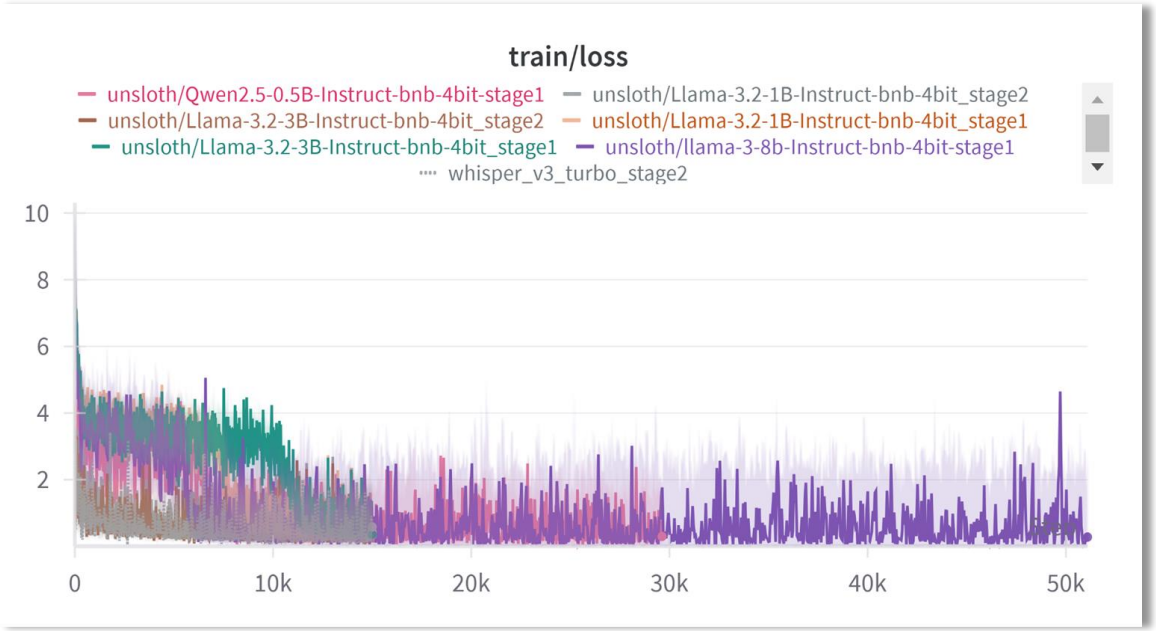
- MoE 구조 적용 → **경량화 & 효율성 극대화**
- 적은 파라미터로도 **높은 성능 유지**

1B 모델을 선택한 이유

- 빠른 추론 속도
- 성능유지
- 리소스 절감
- 적은 차이의 Train Loss

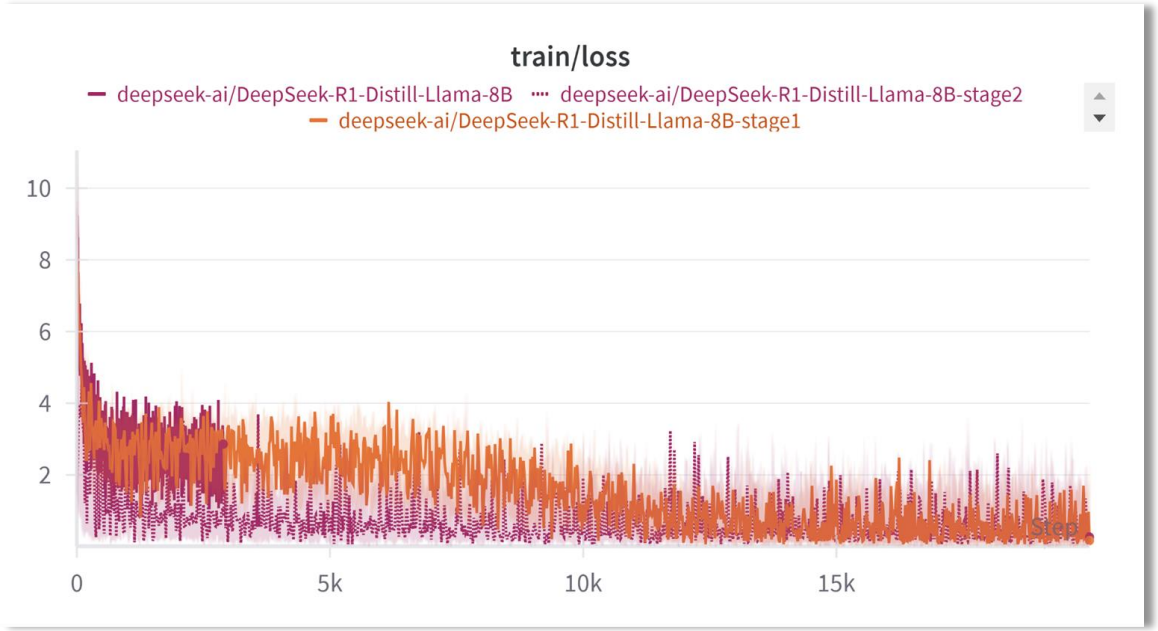
최종 적용 기법

unsloth



- Unsloth loss 비교 표-

deepseek

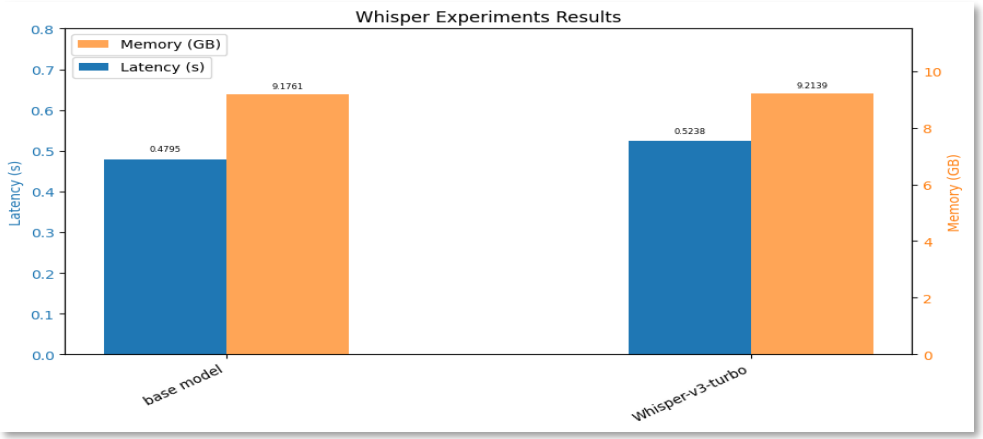


- Deepseek loss 비교 표-

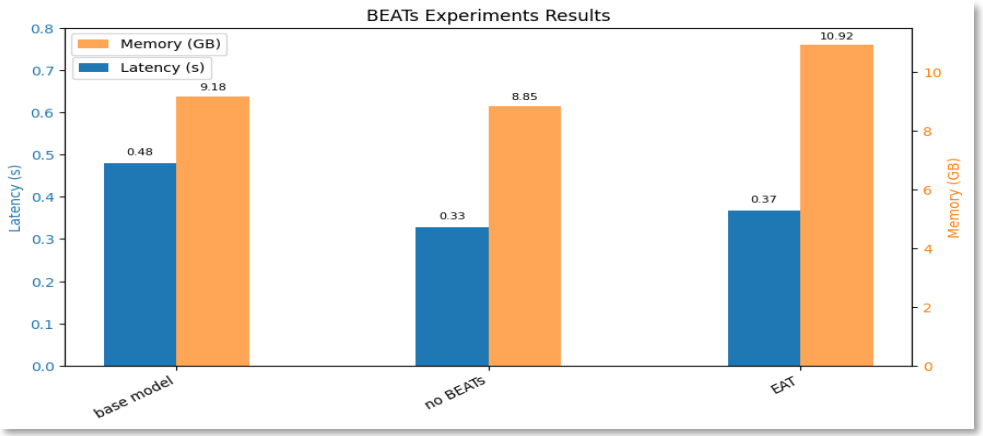
3. Result / Conclusion

1. 실험 결과
2. 학습 결과
3. 후속 개발 및 연구
4. 시연 영상

실험 결과



- whisper 성능비교 표-



- Beats 성능비교 표-

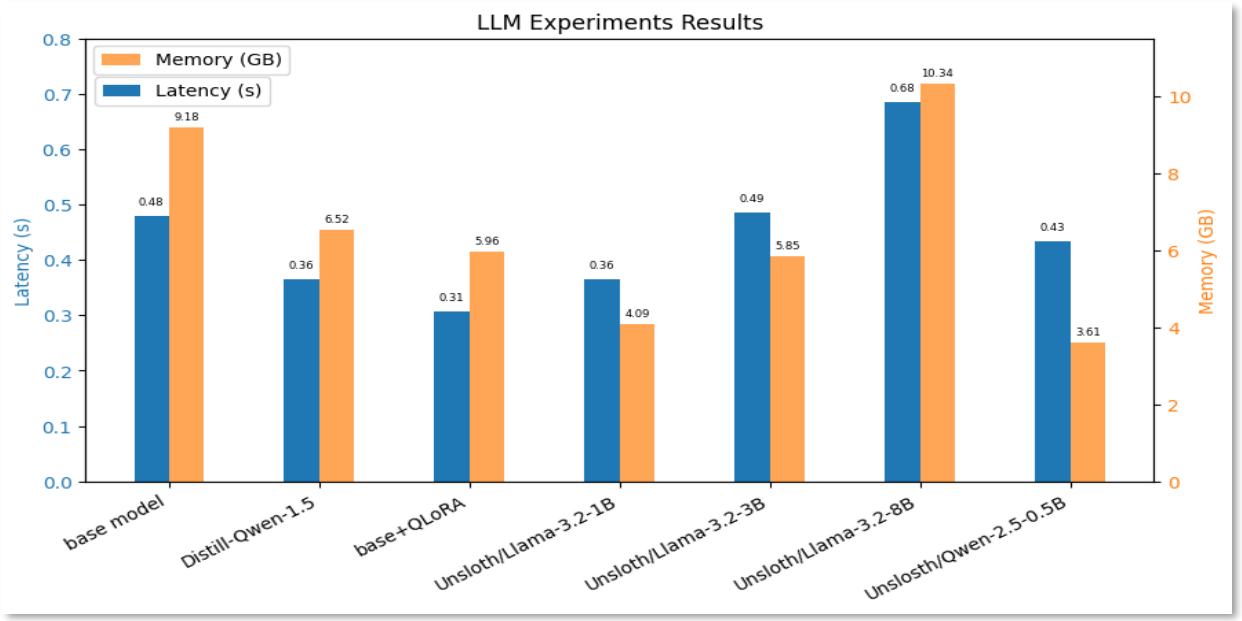
Whisper Experiments

Model	Latency	Memory
base model	0.4795	9.1761
Whisper-v3-turbo	0.5238	9.2139

BEATs Experiments

Model	Latency	Memory
base model	0.4795	9.1761
w/o BEATs	0.3285	8.8462
EAT	0.3680	10.9199

실험 결과

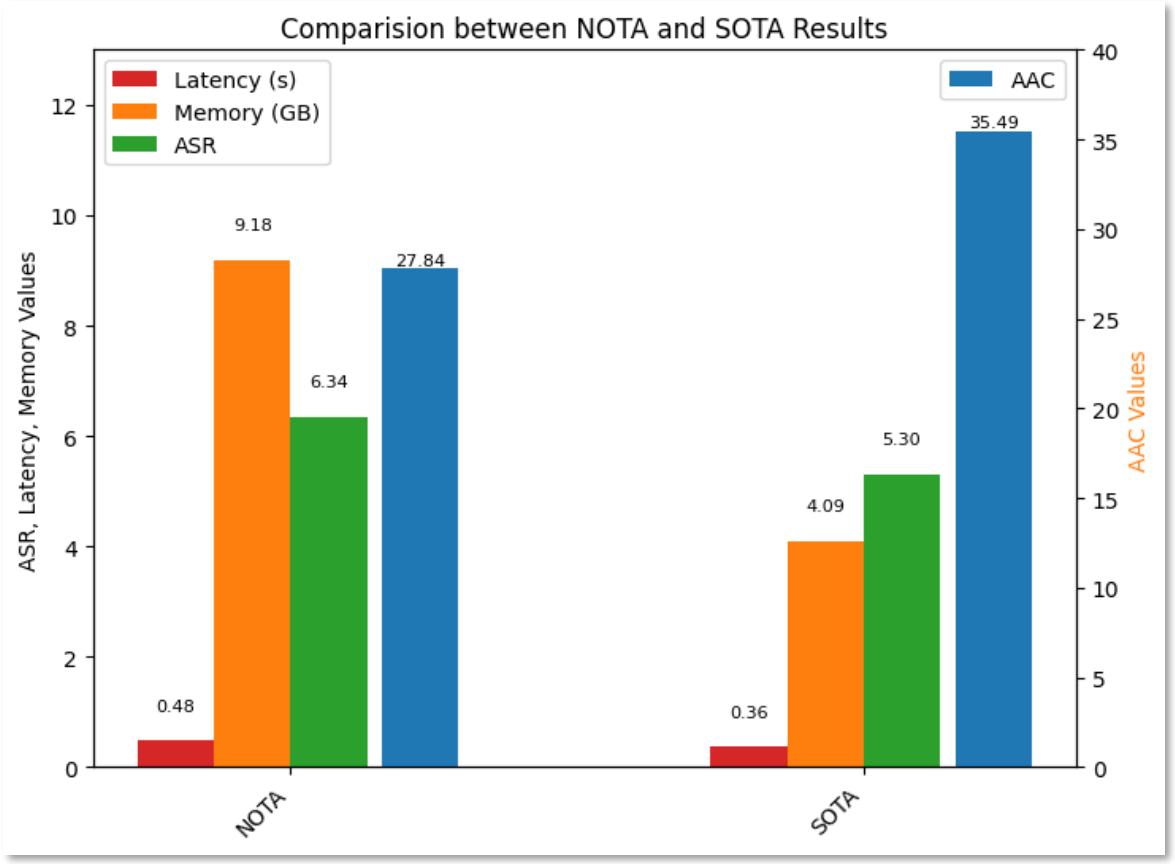


- LLM 성능비교 표-

LLM Experiments

Model	Latency	Memory
base model	0.4795	9.1761
Distill-Qwen-1.5	0.3646	6.5197
Base + QLoRA	0.3060	5.9613
Unsloth/ Llama-3.2-1B	0.3641	4.0938
Unsloth/ Llama-3.2-3B	0.4865	5.8511
Unsloth/ Llama-3.2-8B	0.6847	10.3366
Uslosth/ Qwen-2.5-0.5B	0.4339	3.6098

학습 결과



- 최종성능 비교 표-

Final Model Evaluation

Model	ASR	AAC	Memory	Latency
NOTA	6.34	27.84	9.17	0.47
SOTA	5.30	35.49	4.09	0.36

후속 개발 및 연구

1. 데이터 측면 개선

- 추가적인 **데이터 증강 + 도메인 특화 데이터** 확보
- 사용자 패턴 반영 **Fine-tuning**

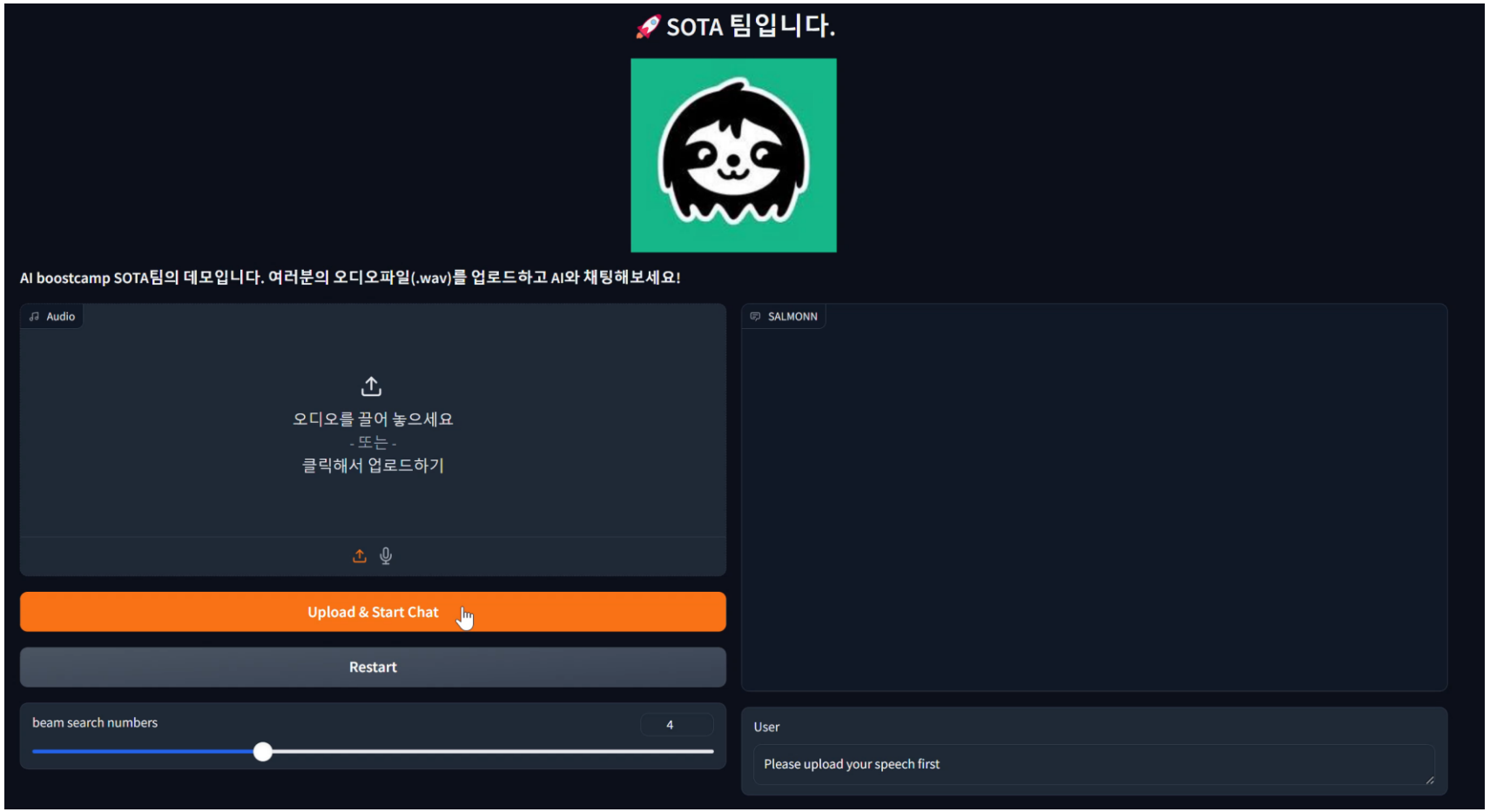
2. 학습/연산 효율 극대화

- **Mixed Precision / Dynamic Gradient Clipping**
- **Flash Attention** 등의 **연산 효율**을 높인 기법으로 학습

3. 추가 연구 아이디어

- **Airflow 자동화**
- **그래프·지식 기반 음성** 후처리
- **TTS 합성 데이터 & Transducer** 비교
- **LLM 디코딩 방식** 개선

시연 영상

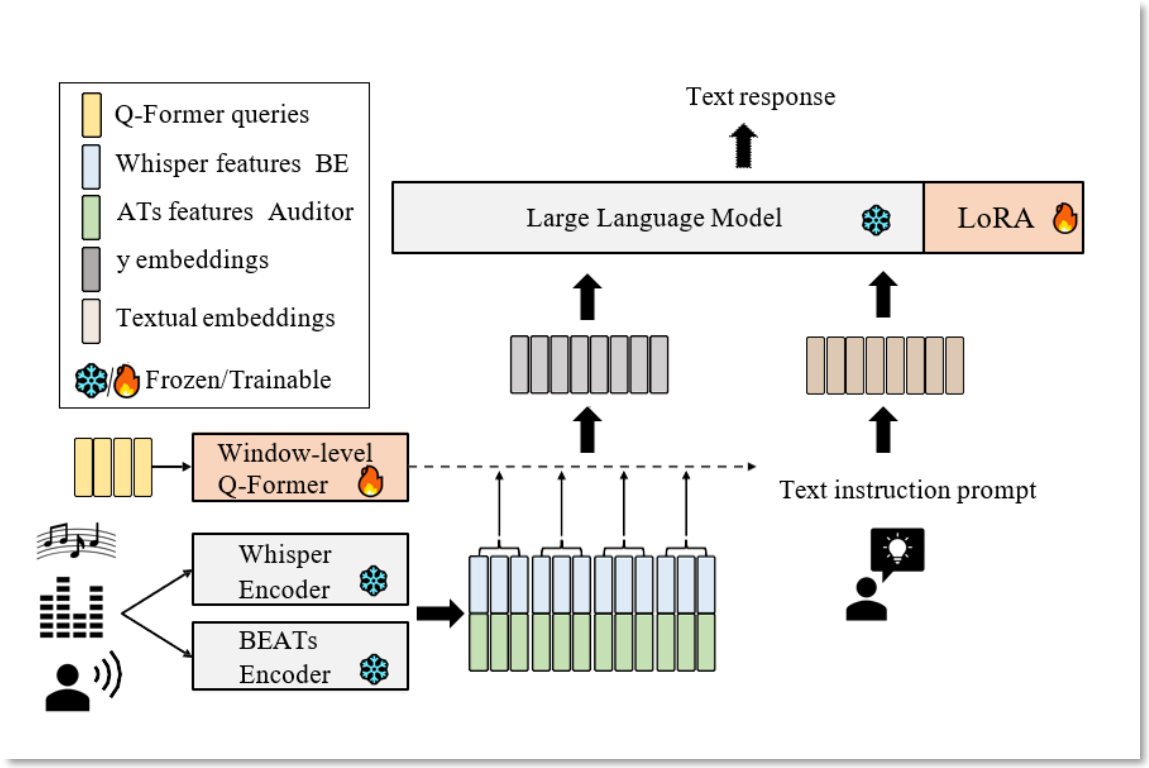


시연 링크: https://fa991a26f767fa9bd4.gradio.live

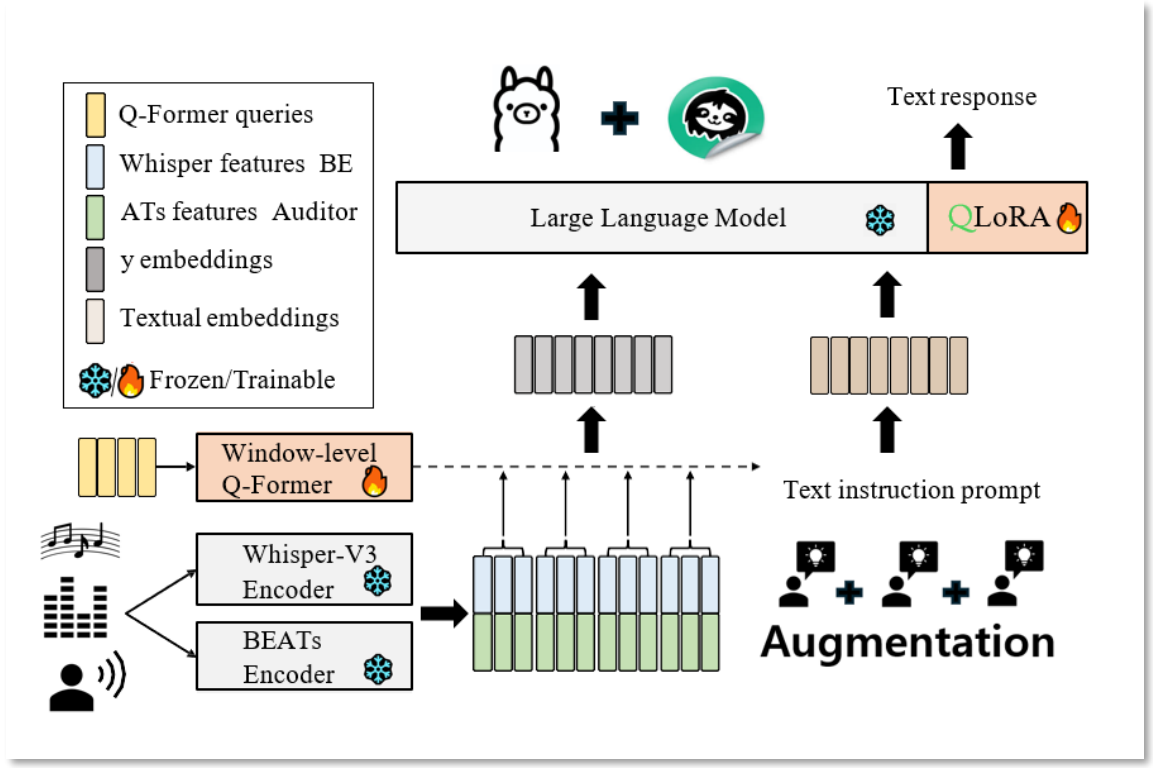
4. 별첨

1. 아키텍처

아키텍처



- 기본 아키텍처 -



- 최종 아키텍처 -

End of Document

Thank You.