Pusan National University
Computer Science and Engineering
Technical Report 2025-5

강의 음성 기반 실시간 AI 주석 시스템 및 지연 최소화를 위한 병렬 처리 통신 구조 설계

Real-Time AI Annotation System for Lecture Audio and Parallel Communication Architecture for Minimizing Latency

드레스코딩 팀

202255514 김동인

202155532 김예슬

202355581 정유성

지도교수 조준수

목차

1. 연구 배경 및 목표

1.1 연구 배경

최근 비대면 및 디지털 교육 환경의 확산과 함께, 강의 녹음 및 필기 보조 도구에 대한 수요가 빠르게 증가하고 있다. 특히 음성 인식 기술의 발전으로 인해, 강의나 회의 음성을 텍스트로 변환해주는 다양한 서비스가 등장하고 있지만, 기존 시스템은 대부분 단순히 음성을 텍스트로 변환하거나 녹음이 완료된 후 요약하는 수준에 그친다.

이러한 시스템들은 강의 내용을 실시간으로 정리하거나, 즉시 슬라이드와 연동해 주석화하는 기능이 부족하며, 사용자는 녹음 파일을 일일이 재청취하거나 변환된 텍스트를 수동으로 정리해야 하는 번거로움을 겪고 있다. 이는 학습 흐름의 단절을 초래할 뿐만 아니라, 오히려 수기 필기보다 더 높은 인지 부하와 반복 작업을 유발하는 문제가 있다.

이에 따라, 음성 데이터를 단순히 텍스트로 전환하는 것을 넘어, 실시간으로 정제되고 활용 가능한 형태로 제공하는 차세대 학습 지원 시스템의 필요성이 대두되고 있다.

1.2 개발 목적 및 필요성

본 프로젝트는 위와 같은 문제점을 해결하기 위해, 음성 인식(STT) 모델과 GPT 기반 자연어 처리(NLP) 모델을 결합한 실시간 강의 주석 시스템을 개발하고자 한다.

사용자는 강의 슬라이드(PDF)를 업로드한 뒤, 실시간으로 녹음되는 강의 음성을 정제된 텍스트 형태로 받아보고, 자동 처리된 내용을 사용자 관점에서 보완할 수 있도록 직접 수정한 후, 이를 슬라이드 위에 드래그 앤 드롭 방식으로 주석처럼 배치할 수 있다. 이러한 접근은 기존의 단순 녹음이나 후처리 기반 시스템과 달리, 강의 진행 중에도 실시간으로 정리 및 필기가 가능하다는 장점을 제공한다.

또한, 본 시스템은 단순한 인터페이스 구현을 넘어, 실시간 처리를 위한 전처리 흐름(STT → GPT → UI) 상의 병목 구간을 식별하고, 이를 비동기 처리 및 병렬 구조로 최적화하는 데에도 목적이 있다. 특히, 음성 인식 결과의 누락 및 지연, 사용자의 인터랙션 반영 속도 등에서 발생할 수 있는 지연 요소들을 분석하고, 이를 최소화하기 위한 구조적 개선 방안을 함께 제시한다.

WebSocket 기반의 실시간 통신 구조 설계, 멀티 스레딩 및 큐 기반 병렬 처리 모델 적용, 모듈 간 비동기 이벤트 전달 구조를 통해 전체 시스템의 응답성을 개선하고, 사용자 경험(UX)을 저해하지 않는 실시간성을 달성하는 것이 이 과제의 핵심 필요성과 목적이다.

1.3 과제 목표

본 프로젝트의 구체적인 목표는 다음과 같다.

1. 멀티모달 AI 기술(STT + GPT)을 활용하여 실시간 강의 음성 기반 주석 시스템 구축
2. PDF 슬라이드 뷰어와 연동하여, 실시간으로 정제된 텍스트를 수정 및 삽입 가능한 형태로 제공하는 기능 구현
3. 사용자가 텍스트를 슬라이드의 원하는 위치에 자유롭게 배치할 수 있는 인터랙티브 UI 설계 및 구현
4. 전체 파이프라인(STT → NLP → UI 렌더링)에 걸친 성능 병목 지점을 식별하고, 이에 대한 성능 개선 전략(큐 처리, 버퍼링, 캐시 사용 등)을 실험적으로 제시
5. 사용자가 주석 작업을 마친 슬라이드를 PDF 형태로 저장 및 다운로드할 수 있는 기능 제공
6. 교육 환경뿐 아니라 회의록, 실시간 보고서 등 다양한 응용 분야로의 확장 가능성 확보

이러한 목표를 통해 본 과제는 단순한 강의 보조 서비스 개발을 넘어, 실시간 AI 처리 시스템의 구조 최적화와 반응성 향상이라는 연구적 기여를 함께 달성하는 것을 지향한다.

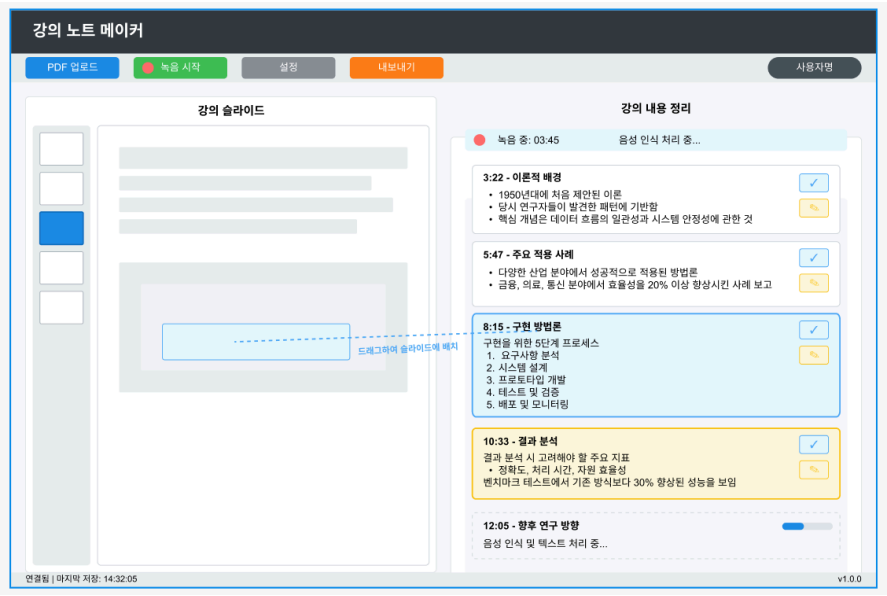
2. 요구사항 분석

2.1 입력 및 출력 데이터 정의

- **입력 데이터**
  - 사용자 업로드 PDF 슬라이드 파일
  - 사용자 마이크로부터 입력되는 실시간 강의 음성
- **출력 데이터**
  - GPT 로 정제된 주석 텍스트 목록 (시간순 정렬)
  - 사용자가 직접 배치한 슬라이드별 주석 포함 PDF 문서 (최종 저장 및 다운로드 가능)

2.2 기능 요구사항

1. **PDF 슬라이드 업로드 및 인터페이스**



  - 사용자가 강의 자료로 PDF 파일을 업로드할 수 있다.
  - 업로드된 PDF 는 웹사이트 내에서 페이지별로 탐색이 가능하며, 사이드바 등으로 편리하게 이동할 수 있다.

- PDF 는 수정되지 않고, 내용은 오직 오버레이(덮어쓰기) 방식으로 보여진다.

2. **실시간 음성 인식 및 정제**
    - 사용자가 녹음 버튼을 통해 강의 음성을 실시간으로 입력할 수 있다.
    - 실시간으로 음성인식을 수행하며 받아쓴 내용을 GPT API 를 통해 정제하여 문맥에 맞는 문장으로 변환한다.
3. **블록 생성 및 관리**
    - 정제된 문장은 오른쪽 클립보드 창에 블록 단위로 하나씩 생성된다.
    - 블록 안 내용은 사용자가 직접 수정할 수 있다.
4. **드래그 앤 드롭 주석 배치**
    - 사용자는 클립보드 블록을 드래그하여 PDF 의 원하는 위치에 배치할 수 있다.
    - 드래그한 블록은 PDF 위에 겹쳐서 표시되며, 실시간으로 시각화된다.
5. **최종 PDF 출력**
    - 사용자가 모든 내용을 정리한 후 'PDF 반환' 버튼을 누르면, 현재까지 배치된 블록이 포함된 PDF 문서가 새로 생성된다.
    - 사용자는 최종 PDF 파일을 다운로드할 수 있다.

2.3 비기능 요구사항

- **실시간성** : 음성 → 주석 변환까지의 평균 지연 $\leq$ 1.5 초
- **안정성**
    - GPT/STT API 응답 실패 시 자동 재시도 및 대기열 관리
    - 시스템 정상 처리율 $\geq$ 90%
- **통신 구조**
    - WebSocket 기반 양방향 통신 구현

- 음성 녹음 및 인식이 진행되는 동안 문장 분석 및 블록 생성 작업 비동기 처리

- **사용자 경험(UX)**
  - 슬라이드에 주석을 놓을 때 마우스 기반 직관적 인터랙션
  - 주석 미리보기, 삭제 버튼, 재정렬 기능 제공
  - 실시간으로 사용자 작업 데이터를 주기적으로 저장

2.4 핵심 기술 요약

- **실시간 STT 및 문장 정제**
  - Google Cloud Speech-to-Text(STT)를 활용한 스트리밍 기반 실시간 음성 인식
  - OpenAI GPT 또는 유사 모델을 이용해 인식된 문장을 문맥 기반으로 정제
- **슬라이드 매핑**
  - 드래그 앤 드롭 인터페이스를 통해 사용자가 텍스트 블록을 PDF 페이지의 특정 위치에 배치
- **주석 자동 생성 및 삽입**
  - 음성 입력으로부터 실시간으로 텍스트 블록이 자동 생성
- **최종 문서 출력**
  - 사용자가 배치한 모든 블록을 포함하는 새로운 PDF 파일을 서버 측에서 생성 및 다운로드 제공

기술 요소	설명
PDF.js 뷰어	사용자가 업로드한 슬라이드 PDF 렌더링
MediaRecorder	브라우저 기반 실시간 마이크 입력 처리
Google STT	음성을 텍스트로 변환
Drag & Drop UI	사용자 주석의 슬라이드 내 배치 지원
GPT (텍스트 정제)	STT 결과를 문장 단위로 요약 및 정제하여 문맥 유지
WebSocket	서버와 클라이언트 간 실시간 메시지 송수신
PDF 출력	주석이 포함된 최종 PDF를 생성 및 다운로드

3. 연구 방향

3.1 프론트엔드 구성

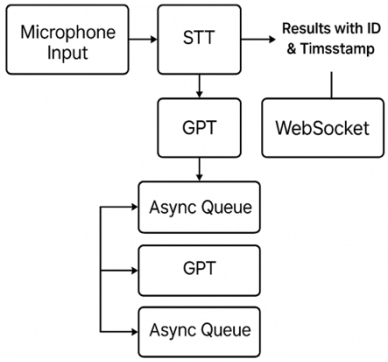
- PDF.js 를 활용하여 PDF 슬라이드를 웹 브라우저 상에서 렌더링
- MediaRecorder API 를 통해 사용자의 마이크 입력을 실시간으로 캡처
- 인식된 문장을 표시하는 텍스트 패널 UI 구성 (시간순 나열, 최신 내용 아래 배치)
- Drag & Drop 기능을 활용하여 텍스트 주석을 사용자가 슬라이드에 직접 배치할 수 있도록 인터페이스 구현
- 사용자 편의를 고려한 텍스트 수정, 삭제, 위치 이동 UI 컨트롤 제공

3.2 백엔드 구성

- Spring Boot 기반 REST API + WebSocket 서버 구현
- MediaRecorder 에서 전송된 음성 데이터를 수신하고 일정 단위로 분할
- 분할된 음성 데이터를 STT 엔진(Google STT)에 전송해 텍스트로 변환
- 텍스트를 GPT API 에 전달하여 문맥 기반 정제 및 요약 수행
- 처리된 주석 데이터를 실시간으로 프론트엔드에 전송하고, 저장 및 PDF 출력에 활용

3.3 실시간 오디오 → 텍스트 흐름 설계

- 마이크 입력을 슬라이딩 윈도우 기반으로 일정 시간 단위 분할하여 처리 효율 개선
- 각 오디오 조각별로 STT → GPT 순차 흐름 정의
- 처리 결과는 고유 ID 와 타임스탬프를 포함하여 클라이언트에 전달



본 과제에서는 강의 음성을 실시간으로 처리하기 위해, 오디오 입력을 일정 시간 간격으로 분할하는 슬라이딩 윈도우 기반 처리 구조를 도입한다. 이 구조는 전체 음성 스트림을 프레임 단위로 분해하여 지연 없는 병렬 처리와 파이프라인화를 가능하게 하며, 오디오 데이터의 누락 없이 정확한 타이밍으로 텍스트 변환을 수행하는 데 유리하다.

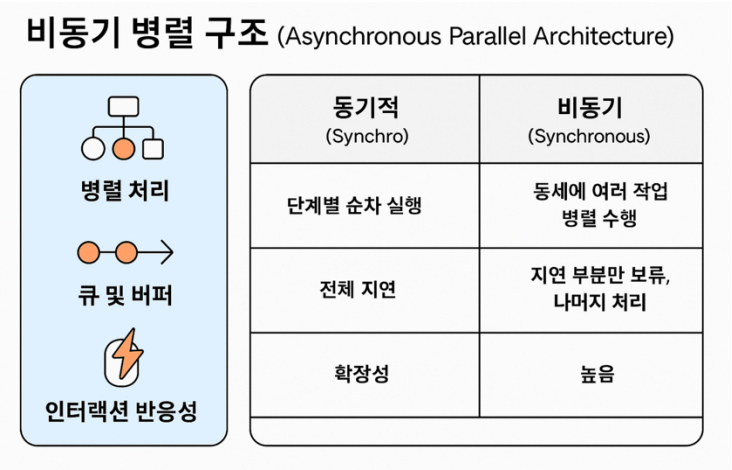
분할된 오디오 단위는 순차적으로 STT 모듈로 전달되며, 초기 텍스트가 생성된 후에는 곧바로 GPT 기반 자연어 처리 모델을 통해 정제 및 요약이 수행된다. 각 처리 단위는 고유 식별자(ID) 및 타임스탬프와 함께 프론트엔드로 전달되며, 주석화 시 해당 슬라이드와의 시각적 맥락을 제공하는 데 활용된다.

본 흐름에서 GPT 응답 지연은 사용자 경험에 직접적인 영향을 미칠 수 있으므로, 비동기 큐 구조 및 응답 지연 버퍼링 전략을 병행 도입할 예정이다. 해당 구조는 텍스트 수신 순서를 보장하며, 응답 순서가 바뀌는 현상에 대한 예외 처리를 포함한다.

연구 방향 측면에서, 다양한 분할 간격에 따른 처리 효율성, 인식 정확도, 사용자 반응성 등을 정량적으로 분석하고자 하며, 최적의 분할 정책을 실험적으로 도출할 예정이다.

3.4 병렬 처리 및 Task 분산 구조

- 멀티스레드 환경에서 오디오 수신, STT 요청, GPT 요청, UI 반영을 모듈별 병렬 분리
- 각 단계는 비동기 Task Queue 기반으로 처리되어, STT/GPT 응답 지연이 전체 흐름에 영향을 주지 않도록 설계
- 버퍼 및 캐시 메커니즘을 도입하여 순서 보장 및 임시 보관 기능 제공
- 사용자 인터랙션 우선순위 처리를 도입하여 시각적 응답성을 높임



본 시스템은 실시간성 확보를 핵심 요구사항으로 하며, 전체 파이프라인(STT, GPT 처리, UI 반영)에서 발생할 수 있는 지연을 최소화하기 위해 비동기 기반의 병렬 처리 아키텍처를 설계한다.

각 처리 단계는 개별 스레드 또는 비동기 작업 단위로 분리되어 실행되며, 오디오 수신 → STT 변환 → GPT 요청/응답 → 프론트엔드 전달까지의 모든 흐름은 Task Queue 기반의 처리 체계로 관리된다. 특히 다수의 사용자가 동시에 사용하는 상황에서도 안정성을 확보하기 위해, 멀티스레드 환경에서의 Task 충돌 방지 및 순서 보장을 위한 버퍼링 및 캐시 구조가 포함된다.

또한, 사용자가 슬라이드에 주석을 삽입하거나 수정하는 등의 인터랙션 동작은 별도의 고우선순위 큐에서 처리되어, 사용자 인터페이스 응답성(UX) 을 저하시키지 않도록 설계한다.

본 연구에서는 병렬 구조의 처리 속도, 지연 발생률, 큐 처리량 등을 측정하고, 처리 모듈 간 최적 분산 구조를 실험적으로 도출한다. 이를 통해 실시간 시스템에서의 병목 지점 특성과 개선 효과를 정량적으로 분석하고자 한다.

4. 개발 및 일정 계획

4.1 개발 환경

- **프론트엔드**
  - React.js + PDF.js
  - TailwindCSS
  - Web Audio API, MediaRecorder API
  - Fabric.js 또는 Konva.js
- **백엔드**
  - Spring Boot
  - Python
  - OpenAI GPT API 연동
  - MongoDB
- **서버 환경**
  - Docker, Amazon Kubernetes

4.2 역할 분담

이름	역할
예슬	슬라이드 뷰어 및 주석 UI 개발, 주석 저장 API 연동, 서버 및 배포 인프라 구성
동인	실시간 녹음 처리 및 WebSocket 통신 전체 흐름 설계, 기획 총괄 및 일정 관리 (PM)
유성	마이페이지 및 사용자 기능 구현, STT/GPT 처리 흐름 설계 및 큐 구조 실험
공통	실시간성 및 처리 성능 실험, 사용자 반응성 개선 실험

4.3 전체 개발 일정표 (Gantt Chart)

단계	5월 1주차	5월 2주차	5월 3주차	5월 4주차	6월 1주차	6월 2주차	6월 3주차	6월 4주차	7월 1주차	7월 2주차	7월 3주차	7월 4주차	8월 1주차	8월 2주차	8월 3주차	8월 4주차	9월 1주차
요구사항 분석 및 시스템 설계																	
개발 환경 세팅 및 프레임워크 구성																	
PDF 업로드/뷰어 기능 개발																	
실시간 음성 인식 기능 개발																	
GPT API 연동 및 정제 기능 개발																	
클립보드 블록 생성 및 편집 기능																	
드래그 앤 드롭 인터페이스 구현																	
실시간 세션 저장 기능																	
최종 PDF 출력 기능																	
전체 통합 테스트 및 디버깅																	
중간보고서 작성 및 발표 준비																	
기능 고도화 및 최적화																	
최종 안정화 및 결과 보고서 작성																	

5. 추후 확장 방안

5.1 교육 현장에서의 활용 가능성

본 시스템은 대학 강의뿐만 아니라 학원, 세미나 등 다양한 교육 현장에서 활용될 수 있다. 특히 교수자 또는 강연자가 제공하는 PDF 슬라이드와 동기화하여 실시간으로 주석을 추가할 수 있으므로, 청중들은 학습 내용을 보다 손쉽게 정리하고 복습할 수 있다. 향후에는 다수의 사용자가 동시에 같은 슬라이드를 주석하는 협업 기능, 강의 영상과의 동기화 기능 등으로 확장할 수 있다.

5.2 내용 자동 요약 기능

실시간 강의 내용을 문장별로 정제하는 것에 더해, 강의 종료 후 전체 내용을 요약하여 주요 핵심 포인트만 추출하는 기능도 구현할 예정이다. 이를 위해 GPT 또는 Bart 기반 요약 알고리즘을 추가로 연동하며, 사용자가 클릭 한 번으로 요약본을 받을 수 있도록 편의성을 높인다. 향후에는 챕터별 요약, 키워드 기반 요약 기능으로 고도화할 수 있다.