

# 파이보(Pibo) 로봇 기반 실시간 토론 지원 시스템

상세 정리 문서

## 1. 프로젝트 목표

파이보 로봇에 탑재할 실시간 토론/회의 지원 AI 시스템 개발. 여러 사용자가 대화하는 것을 AI가 듣고, 실시간으로 문서화하며, 필요시 사회자로서 중재까지 수행.

## 2. 핵심 기술 스택

### 2.1 STT (Speech-to-Text)

- 실시간/저지연(Low-Latency) 음성 인식 필수
- 사용자들의 대화를 끊김 없이 텍스트로 변환
- 로컬 모델 사용 권장 (API 최소화, 오프라인 환경 고려)

### 2.2 TTS (Text-to-Speech)

- AI 가 중재나 응답을 할 때 음성으로 출력
- 마찬가지로 저지연 필요

### 2.3 실시간 문서 생성

- STT로 변환된 텍스트를 즉시 문서로 기록
- 이 단계에서는 LLM을 직접 사용하지 않음 (단순 기록)
- 문서가 일정 크기가 되면 Vector DB로 이관

### 2.4 Vector DB (RAG)

- 대화량이 많아지면 컨텍스트 윈도우 초과 → Vector DB 필요
- 문서를 청크 단위로 분할하여 저장
- 나중에 특정 정보 검색/참조 시 활용
- 단, 무조건 필요한 건 아님: 컨텍스트 내에서 처리 가능하면 원본 문서 직접 사용도 가능

### 2.5 Function Calling / MCP

- 웹 검색 같은 기능은 **직접 구현할 필요 없음**
- MCP(Model Context Protocol)로 이미 잘 만들어진 도구들이 있음
- 각각의 평선을 잘 설계하는 것은 의미 있지만, 웹 검색 평선 자체를 만드는 건 불필요

## 3. 시스템 아키텍처 상세

### 3.1 전체 흐름

[사용자들 대화]



[STT - 실시간 음성 인식] (저지연, 로컬 모델 권장)



[실시간 문서 생성 - 버퍼] LLM 미사용, 단순 텍스트 기록



일정 크기 도달 시 → [Vector DB 저장] + 새 문서 버퍼 시작



[백그라운드 AI 판단 루프] (10~30초 주기)

### 3.2 백그라운드 AI 판단 로직

질문: "지금 사회자로서 중재가 필요한가?"

- **중재 불필요 시:** 대기 유지, 계속 모니터링
- **중재 필요 시:** 개입 실행
  - "잠시만요" 액션 발동
  - 중재 메시지 생성
  - TTS로 음성 출력

## 4. 화자별 쟁점 도출 프로세스

### 4.1 트리거 조건

- 사용자 요청: "지금 논지 정리해줘"
- 주기적 자동 호출: 백그라운드에서 AI가 스스로 리콜

### 4.2 처리 방식

Case A: 컨텍스트 내 처리 가능 (대화량 적음)

원본 문서 전체 → LLM → 화자별 논지 요약

## Case B: Vector DB 필요 (대화량 많음)

Vector DB에서 "화자 A 발언" 쿼리 → 화자 A 관련 내용만 추출

Vector DB에서 "화자 B 발언" 쿼리 → 화자 B 관련 내용만 추출

추출된 내용들 → LLM → 화자별 논지 요약

## 4.3 출력 형식 예시

### [화자 A]

- 주장 1: ...
- 주장 2: ...

### [화자 B]

- 주장 1: ...
- 반론: ...

## 5. 웹 앱 UI 구상

화면을 좌우 2분할하여 실시간 정보를 표시하는 웹 애플리케이션

왼쪽 패널	오른쪽 패널
<p><b>실시간 대화 기록</b></p> <p>10:01 화자A: 저는 이렇게 생각합니다...</p> <p>10:02 화자B: 그건 좀 다른 관점에서...</p> <p>10:03 화자A: 네, 하지만 제 의견은...</p> <p>10:04 화자C: 두 분 의견 모두 일리가...</p> <p>[실시간 업데이트 중...]</p>	<p><b>화자별 논지 요약</b></p> <p>▶ <b>화자 A</b></p> <ul style="list-style-type: none"><li>핵심주장 1</li><li>핵심주장 2</li></ul> <p>▶ <b>화자 B</b></p> <ul style="list-style-type: none"><li>핵심주장 1</li><li>화자A에 대한 반론</li></ul> <p>[자동 요약 업데이트 중...]</p>

[AI 중재] 잠시만요. 현재 논점을 정리하면...

## 6. 개발 원칙 및 방향

### 6.1 오프라인 우선

- 인터넷 없는 환경에서 작동 가능하도록 설계
- API 호출 최소화, 로컬 모델 다운로드하여 사용

### 6.2 단계별 개발 (일주일 단위)

주차	목표	비고
1주차	STT 실시간 구현	완벽 추구 X, 적정선에서 마무리
2주차	실시간 문서화	버퍼 → DB 이관 로직
3주차	Vector DB 연동	청크 분할, 검색 기능
4주차	화자별 요약	LLM 연동
5주차	중재 판단 로직	백그라운드 루프
6주차	웹앱 UI	프론트엔드
7주차+	통합 및 테스트	파이보 탑재

### 6.3 중요 마인드셋

"끝까지 가려면 끝이 없어요. 어느 정도 선에서 끝내고 넘어가야 해요. 계속 찾다 보면 더 좋은 게 나올 수 있지만, 그러면 나머지는 아예 못 가요."

## 7. 진로 관련 조언 요약

항목	내용
정출연 연구직	박사 학위 거의 필수
석사 졸업 후	주로 개발 업무 담당, 연구 리딩은 어려움
조언	다양한 박사님/선배들과 상담 권장
핵심 질문	"롱텀 목표가 뭔가?" → 그에 맞는 경로 설계 필요

## 8. 추후 확인 필요 사항

- STT 모델 선정 (Whisper? 다른 로컬 모델?)
- Vector DB 종류 (Chroma? FAISS? Milvus?)
- 화자 분리(Speaker Diarization) 구현 방법
- 중재 판단 기준/프롬프트 설계
- 파이보 로봇 하드웨어 스펙 및 제약사항