

테디노트 LAB RAG팀 - AI Research Engineer 지원 자료

- 요약
- 채용 요구사항별 역량 매칭
 - 자격 요건 충족도
 - 1. 팀원 간 의사소통 및 협업 능력
 - 2. CS 지식 및 Python Engineering 역량
 - 3. RAG & Agent 관련 라이브러리 활용 능력
 - 4. 다양한 도메인의 RAG & Agent 프로젝트 경험
 - 5. 다양한 Vector Database 활용 경험
 - 6. RAG 성능 고도화를 위한 데이터 분석적 사고역량
 - 우대 사항 충족도
 - 1. Model (LLM, Embedding, Re-ranking) 기본 이해
 - 2. 최신 트렌드 파악
 - 3. 성능 고도화 인사이트 도출
 - 4. Production Level End-to-End RAG 파이프라인
 - 5. 다양한 벤치마크 및 방법론 응용
 - 6. 클라우드 플랫폼 사용 경험
- 핵심 강점 및 차별화 포인트
 - 1. Production Level RAG 경험 (총 2년 2개월+)
 - 2. 멀티 에이전트 시스템 설계 역량
 - 3. 하이브리드 검색 및 성능 고도화
 - 4. 팀 리딩 및 프로젝트 관리
 - 5. 오픈소스 기여 및 협업
 - 6. 다양한 도메인 경험
 - 7. 대규모 팀 협업 및 멀티 팀 조율
- 주요 프로젝트 요약
 - 1. SK PharmaAIX MR Assistant (2024.07 ~ 현재, 6개월+)
 - 2. 한국자동차연구원 AI 에이전트 (2024.03 ~ 2025.11, 1년 8개월)
 - 2. LangChain Open Tutorial 오픈소스 기여 (2025.01 ~ 2025.02, 2개월)
 - 3. Podly - AI 기반 맞춤형 음성 뉴스 (2024.08 ~ 2024.09, 2개월)
 - 4. Modular RAG 연구 (2024.10 ~ 2024.12)
- 기술 역량 상세
 - AI/ML

- Backend
- Frontend
- Database
- DevOps & Infrastructure
- 기타
- [소프트 스킬](#)
 - 프로젝트 관리
 - 이해관계자 관리 및 조율
 - 아키텍처 설계
 - 문제 해결
 - 오픈소스 기여 및 학습
 - 꾸준한 개발 습관 및 기술 공유
- [테디노트 RAG팀에 기여할 수 있는 부분](#)
 - 1. 다양한 도메인의 RAG 프로젝트 수행
 - 2. 성능 고도화를 위한 평가지표 수립
 - 3. RAG & Agent 선행리서치
 - 4. SLM Training (추가 학습 의지)
- [관련 링크](#)
- [희망 연봉](#)
- [지원 동기 및 포부](#)

요약

신승엽 | AI Research Engineer 지원자

- 1년 8개월간 자동차 분야 AI Agent 프로젝트 PL 경험 (1인 → 3명 팀 리딩)
- LangChain/LangGraph 기반 멀티 에이전트 시스템 설계 및 구축 경험
- RAG 시스템 설계, 구현, 성능 고도화 전문
- Production Level End-to-End RAG 파이프라인 운영 경험
- Qdrant, PostgreSQL(pgvector), Neo4j 등 다양한 Vector DB 활용 경험

온라인 프로필:

- GitHub: <https://github.com/syshin0116> (25+ repositories, 꾸준한 일일 커밋 습관)
- 기술 블로그: <https://syshin0116.github.io> (Quartz 기반 개인 기술 블로그, 지속적 포스팅)

채용 요구사항별 역량 매칭

자격 요건 충족도

1. 팀원 간 의사소통 및 협업 능력

- 한국자동차연구원 프로젝트 (1년 8개월)
 - 1인 개발 → 3명 팀 PL로 성장
 - 요구사항 분석, 일정 관리, 팀원 코드 리뷰 진행
 - 팀원 업무 분담 및 프로젝트 리딩 경험
- LangChain Open Tutorial 오픈소스 기여 (2개월)
 - 2,000+ 커밋 대규모 프로젝트 참여
 - Peer Review 세션 및 협업 경험
 - 3개 튜토리얼 작성 및 GitHub/GitBook 게시

2. CS 지식 및 Python Engineering 역량

- 기술 스택:
 - Backend: FastAPI, Flask, Python
 - 비동기 처리: Celery, Redis
 - ORM: SQLAlchemy, Kysely, Drizzle ORM
 - 자료구조 및 알고리즘: 코딩테스트 다수 풀이 경험
- 코드 품질 관리:
 - 200+ 테스트 케이스 작성 경험
 - 체계적인 프로젝트 구조 설계 및 모듈화

3. RAG & Agent 관련 라이브러리 활용 능력

- LangChain/LangGraph 전문가:
 - 멀티 에이전트 시스템 설계 (Research Agent, Report Agent, Supervisor Agent)
 - StateGraph 기반 복잡한 워크플로우 구현
 - LangGraph Checkpoints 활용 상태 저장 및 롤백 기능
- FastAPI 전문가:
 - 한국자동차연구원: Flask → FastAPI 마이그레이션 주도
 - RESTful API 설계 및 문서화
 - SSE(Server-Sent Events) 스트리밍 구현

4. 다양한 도메인의 RAG & Agent 프로젝트 경험

프로젝트 1: SK PharmaAIX MR Assistant (2024.07 ~ 현재, 6개월+)

- **역할:** AI 파트 리드 / 랩큐 팀 PL (BCG RA 15명 + 랩큐 팀원 3명 리딩)
- **기여:**
 - **Multi-Agent Supervisor** 구조 설계 및 구현 (Chain → Agentic 전환)
 - **Vector DB 마이그레이션** (LanceDB → Qdrant) 및 하이브리드 검색 구현
 - **응답 속도 83% 개선** (3분 → 30초)
 - **답변 커버리지 233% 향상** (30% → 100%)
 - **답변 품질 99% 만족도** 달성 (1,000개 질문 기반 정성 평가)
 - 제약 도메인 특화 RAG 시스템 구축 (약 350명 MR 사용자)
 - 대규모 멀티 팀 협업 (약 10개 팀과 소통 및 조율)

프로젝트 2: 한국자동차연구원 AI 에이전트 (2024.03 ~ 2025.11, 1년 8개월)

- **역할:** 1~2차 1인 풀스택 개발 → 3~4차 메인 PL (팀 3명)
- **기여:**
 - LangGraph 멀티 에이전트 시스템 설계 및 구현 (Research → Report → Supervisor)
 - 하이브리드 검색 (Dense + Sparse) 도입으로 검색 정확도 향상
 - 4개 파일 파서 통합 (Upstage, LlamaParse, Docling, Unstructured) 및 성능 비교
 - 파싱 결과 시각화 (PDF ↔ Markdown Interactive Highlight)
 - 실시간 스트리밍 응답 시스템 구축 (SSE)
 - End-to-End Production 환경 운영 (한국자동차연구원 내부 서버)

프로젝트 2: Podly - AI 기반 맞춤형 음성 뉴스 (2024.08 ~ 2024.09, 2개월)

- **역할:** 팀장 / AI 및 백엔드 개발 (팀 4명)
- **기여:**
 - RAG 기반 뉴스 큐레이션 시스템 구현
 - 다양한 LLM 모델 테스트 (GPT, Upstage Solar, Gemma2, Phi 3.5, Llama 3.1)
 - OpenAI Whisper 음성 인식 및 ElevenLabs TTS 통합

프로젝트 3: LangChain Open Tutorial 오픈소스 기여 (2025.01 ~ 2025.02, 2개월)

- **기여:**
 - LlamaParse 튜토리얼 작성 및 번역
 - Conversation Memory Management System 튜토리얼 (단기/장기 메모리)
 - CoT-Based Smart Web Search 튜토리얼 (Multi-Query 생성)

프로젝트 4: Modular RAG 연구 (2024.10 ~ 2024.12)

- **연구 주제:**
 - 모듈화된 RAG 시스템 아키텍처 연구

- Chat History 관리 (컨텍스트 윈도우 최적화)
- RDBMS vs NoSQL 비교 연구
- Short-term & Long-term Memory 구조 설계

5. 다양한 **Vector Database** 활용 경험

- **Qdrant:**
 - SK PharmaAIX: LanceDB → Qdrant 마이그레이션, 하이브리드 검색 구현
 - 한국자동차연구원: 하이브리드 검색 (Dense + Sparse) 구현
- **PostgreSQL (pgvector):**
 - Kysely ORM 기반 데이터 모델링
 - 파싱된 파일 메타데이터 저장 및 벡터 검색
- **Neo4j (Graph DB):**
 - AppHub 프로젝트: 관계 기반 검색 시스템 설계
- **기타 경험:**
 - Chroma (학습 및 테스트)
 - Milvus, Elasticsearch (개념 이해 및 비교 연구)

6. RAG 성능 고도화를 위한 데이터 분석적 사고역량

- **SK PharmaAIX 성능 고도화:**
 - 응답 속도 83% 개선 (3분 → 30초)
 - 답변 커버리지 233% 향상 (30% → 100%)
 - 답변 품질 99% 만족도 달성 (1,000개 질문 정성 평가)
- **하이브리드 검색 도입:**
 - Dense Retrieval (벡터 검색) + Sparse Retrieval (키워드 검색) 결합
 - 검색 정확도 향상
- **멀티 파일 파서 성능 비교:**
 - 4개 파서(Upstage, LlamaParse, Docling, Unstructured) 비교 분석
 - 파서별 지원 파일 형식, 비용, 성능 분석 자료 작성
 - Interactive Highlight 기능으로 파싱 성능 실시간 검증
- **데이터 전처리 경험:**
 - 고용노동 공공데이터: Pandas 기반 결측치, 이상치 처리
 - SK Chemical Copoly COOH 예측: 시계열 데이터 전처리, 피쳐 엔지니어링

우대 사항 총족도

1. Model (LLM, Embedding, Re-ranking) 기본 이해

- **LLM 통합 경험:**
 - OpenAI API (GPT-3.5/4/4o), Claude, Private Model
 - Upstage Solar, Google Gemma2, Microsoft Phi 3.5, Meta Llama 3.1
- **Embedding:**
 - OpenAI Embeddings, Upstage Embeddings 사용
 - 임베딩 파이프라인 구축 및 벡터화
- **Re-ranking:**
 - 개념 이해 (Fine-Tuning 경험은 없으나 학습 의지 있음)

2. 최신 트렌드 파악

- **LangChain Open Tutorial** 컨트리뷰터:
 - RAG, LangGraph, Memory Management 최신 트렌드 학습
 - Peer Review 세션 통해 기술 인사이트 확장
- **Modular RAG** 연구:
 - RAG 시스템 아키텍처 연구
 - RDBMS vs NoSQL 비교, Memory 최적화 전략 연구
- **오픈소스 및 커뮤니티 활동:**
 - 2,000+ 커밋 대규모 프로젝트 참여 경험
 - GitHub, GitBook 통한 지식 공유

3. 성능 고도화 인사이트 도출

- **하이브리드 검색 도입:**
 - Dense + Sparse 검색 결합으로 정확도 향상
 - 빠른 실험 및 결과 도출 (SSE 스트리밍으로 실시간 피드백)
- **멀티 파서 통합 및 비교:**
 - 4개 파서 성능 비교 분석 자료 작성
 - 비용 대비 성능 최적화 (기본값: Upstage)
- **Interactive Highlight** 기능:
 - PDF ↔ Markdown 호버 시 하이라이트
 - 파싱 성능 실시간 확인 가능

4. Production Level End-to-End RAG 파이프라인

- **SK PharmaAIX (6개월+ 운영 중):**

- Azure 환경 배포 및 운영
- 약 350명 MR 사용자 대상 Production 서비스
- Multi-Agent Supervisor 구조 기반 RAG 시스템
- MariaDB + Qdrant + Redshift 통합 조회
- 1,000개 질문 기반 정성 평가 및 99% 만족도 달성
- 대규모 멀티 팀 협업 (약 10개 팀)
- **한국자동차연구원 (1년 8개월 운영):**
 - 내부 서버 (10.10.0.31) 배포 및 운영
 - Docker Compose 기반 멀티 컨테이너 환경
 - Keycloak SSO 인증, LangSmith/DataDog 모니터링
 - 파일 업로드 → 파싱 → 임베딩 → 벡터 저장 → 검색 → 응답 생성 전체 파이프라인 구축
 - 200+ 테스트 케이스 작성 및 품질 관리
 - 실시간 스트리밍 응답 (SSE)

5. 다양한 벤치마크 및 방법론 응용

- **멀티 에이전트 시스템 설계:**
 - Research Agent: 웹 검색 및 문서 분석
 - Report Agent: 검색 결과 기반 보고서 생성
 - Supervisor Agent: 워크플로우 조율 및 최종 응답 생성
- **Multi-Query 생성:**
 - 1개 질문 → 3~5개 검색 쿼리 생성
 - 비동기 실행으로 효율성 향상
- **Plan-and-Execute QA:**
 - CoT(Chain-of-Thought) 기반 복잡한 쿼리 다단계 분해
 - 검색 → 추출 → 추론 → 응답 생성 파이프라인

6. 클라우드 플랫폼 사용 경험

- **AWS:** Men-in-Black 프로젝트 (NVIDIA GPU 환경)
- **GCP:** Podly 프로젝트 배포
- **Azure:** 학습 경험 (SeSAC Azure ML Classic)
- **Vercel, Railway, Fly.io:** AppHub 프로젝트 배포 계획

핵심 강점 및 차별화 포인트

1. Production Level RAG 경험 (총 2년 2개월+)

- **SK PharmaAIX** (6개월+ 운영 중): 약 350명 MR 사용자, 99% 만족도
- **한국자동차연구원** (1년 8개월 운영): End-to-End RAG 파이프라인
- 실무 환경에서의 문제 해결 능력 (엡지 케이스, 성능 최적화, 보안)
- **정량적 성과:**
 - SK PharmaAIX: 응답속도 83% 개선 (3분 → 30초), 답변 커버리지 233% 향상 (30% → 100%)
 - 한국자동차연구원: 보고서 작성 시간 98% 단축 (1시간 → 1분)

2. 멀티 에이전트 시스템 설계 역량

- **SK PharmaAIX**: Multi-Agent Supervisor + Chain 하이브리드 구조
- **한국자동차연구원**: Research → Report → Supervisor 3단계 설계
- LangGraph 기반 복잡한 워크플로우 구현
- StateGraph, Checkpoints 활용한 상태 관리 및 롤백

3. 하이브리드 검색 및 성능 고도화

- Dense + Sparse 검색 결합으로 검색 정확도 향상
- Multi-Query 생성 및 비동기 실행으로 효율성 개선
- 멀티 파서 통합 및 성능 비교 분석

4. 팀 리딩 및 프로젝트 관리

- **대규모 팀 리딩**: BCG RA 15명 + 랩큐 팀원 3명 (SK PharmaAIX)
- **멀티 팀 협업**: 약 10개 팀과 소통 및 조율 (추진팀, 인프라팀, 보안팀, QA팀 등)
- 1인 개발 → 3명 팀 PL로 성장 (한국자동차연구원)
- 아키텍처 재설계 경험: Flask → FastAPI, Chain → Multi-Agent
- 200+ 테스트 케이스 작성으로 체계적인 품질 관리

5. 오픈소스 기여 및 협업

- LangChain Open Tutorial 대규모 프로젝트 기여
- 3개 튜토리얼 작성 (LlamaParse, Memory Management, Smart Web Search)
- Peer Review 및 기술 공유 경험

6. 다양한 도메인 경험

- 제약: SK PharmaAIX MR Assistant (MR 영업 지원)
- 자동차 분야: 한국자동차연구원 AI 에이전트
- 뉴스/음성: Podly AI 기반 음성 뉴스
- 데이터 분석: 고용노동 공공데이터 시각화
- 제조: SK Chemical Copoly COOH 예측
- Computer Vision: Men-in-Black 차량 감지 시스템

7. 대규모 팀 협업 및 멀티 팀 조율

- SK PharmaAIX: BCG RA 15명 + 랩큐 팀원 3명 리딩
- 약 10개 팀(추진팀, 인프라팀, 보안팀, QA팀, 백엔드, 프론트 등)과 협업

주요 프로젝트 요약

1. SK PharmaAIX MR Assistant (2024.07 ~ 현재, 6개월+)

역할: AI 파트 리드 / 랩큐 팀 PL (BCG RA 15명 + 랩큐 팀원 3명 리딩)

주요 성과:

- 응답 속도 **83% 개선**: 3분 → 30초
- 답변 커버리지 **233% 향상**: 30% → 100%
- 답변 품질 **99% 만족도** 달성
- Multi-Agent Supervisor 구조 설계 및 구현
- Vector DB 마이그레이션 (LanceDB → Qdrant)
- 대규모 멀티 팀 협업 (약 10개 팀)

기술스택:

- Backend: FastAPI, Python
- Database: MariaDB, Qdrant (Vector DB), AWS Redshift, Redis
- AI/ML: LangChain, LangGraph, Azure OpenAI API
- Infrastructure: Azure, Docker

담당 업무:

1. AI 파트 리딩 및 멀티 팀 협업

- BCG PM 다음으로 AI 파트 전체 리더
- BCG RA 5~6명 (3개월 로테이션, 총 15명) + 랩큐 팀원 3명 리딩
- 추진팀, 인프라팀, 보안팀, QA팀, 백엔드, 프론트 등 약 10개 팀과 협업

2. 서비스 아키텍처 재설계

- Chain 기반 → Multi-Agent Supervisor 구조로 전환
- Chain + Agentic 하이브리드 구조 설계
- Tool Agent 설계 (MariaDB, Qdrant, Redshift 조회)

3. Vector Database 교체 및 성능 고도화

- LanceDB → Qdrant 마이그레이션
- 하이브리드 검색 (Dense + Sparse) 구현
- 응답 속도 83% 개선, 답변 커버리지 233% 향상

4. 제약 도메인 특화 RAG 시스템

- 약 350명 MR 사용자 대상
- 영업활동 기록, 제품 정보 검색, 고성과자 노하우 공유

2. 한국자동차연구원 AI 에이전트 (2024.03 ~ 2025.11, 1년 8개월)

역할: 1~2차 1인 풀스택 개발 → 3~4차 메인 PL (팀 3명)

주요 성과:

- 보고서 작성 시간 98% 단축: 1시간 → 1분 (프롭 프로젝트 원리 이식)
- 전체 시스템 아키텍처 재설계: Flask → FastAPI, HTML/CSS/JS → Next.js + React, ChromaDB → Qdrant
- LangGraph 멀티 에이전트 시스템 설계 및 구현
- 하이브리드 검색 (Dense + Sparse) 도입
- 4개 파일 파서 통합 및 성능 비교
- 실시간 스트리밍 응답 (SSE) 구축
- Production 환경 운영 (한국자동차연구원 내부 서버)

기술스택:

- Backend: FastAPI, Celery, Redis
- Frontend: Next.js, React, TypeScript
- Database: PostgreSQL (Kysely ORM), Qdrant (Vector DB)
- AI/ML: LangChain, LangGraph, OpenAI API, Claude
- Infrastructure: Docker, Keycloak (SSO), LangSmith, DataDog

담당 업무:

1. 프로젝트 리딩 및 아키텍처 설계

- 전체 시스템 아키텍처 재설계 (Flask → FastAPI, HTML/CSS/JS → Next.js + React, ChromaDB → Qdrant)
- API 설계 및 데이터베이스 스키마 설계
- 프롭 프로젝트 원리 이식으로 보고서 작성 시간 98% 단축 (1시간 → 1분)

2. LangGraph 멀티 에이전트 시스템

- Research Agent: DuckDuckGo 검색, 문서 분석
- Report Agent: 검색 결과 기반 보고서 생성
- Supervisor Agent: 워크플로우 조율 및 최종 응답

3. 하이브리드 검색 (Dense + Sparse)

- Dense Retrieval: 벡터 검색 (Qdrant)
- Sparse Retrieval: 키워드 검색 (BM25)
- 검색 정확도 향상

4. 멀티 파일 파서 통합 (4차 - RAG용 임베딩 SW 모듈)

- Upstage, LlamaParse, Docling, Unstructured 4개 파서 통합
- 파서별 성능 비교 및 비용 분석 자료 작성
- 파싱 결과 시각화 (PDF ↔ Markdown Interactive Highlight)

5. 실시간 스트리밍 응답 (SSE)

- Server-Sent Events 기반 스트리밍 구현
- 사용자 경험 개선 (실시간 피드백)

2. LangChain Open Tutorial 오픈소스 기여 (2025.01 ~ 2025.02, 2개월)

역할: 오픈소스 컨트리뷰터 (번역 및 신규 튜토리얼 작성)

주요 성과:

- 2,000+ 커밋 대규모 프로젝트 참여
- 3개 튜토리얼 작성 및 GitHub/GitBook 게시
- Peer Review 세션 협업 경험

담당 업무:

1. LlamaParse 튜토리얼

- PDF, Word, PPT, Excel 파싱 구현

- Multimodal 모델(GPT-4o) 기반 문서 분석

2. Conversation Memory Management System

- LangGraph 기반 단기/장기 메모리 구현
- StateGraph 활용 대화 플로우 자동화

3. CoT-Based Smart Web Search

- Plan-and-Execute QA 시스템 구현
- Multi-Query 생성 (1개 질문 → 3~5개 쿼리)

3. Podly - AI 기반 맞춤형 음성 뉴스 (2024.08 ~ 2024.09, 2개월)

역할: 팀장 / AI 및 백엔드 개발 (팀 4명)

주요 성과:

- RAG 기반 뉴스 큐레이션 시스템 구축
- 다양한 LLM 모델 테스트 및 최적 모델 선택

기술스택:

- Backend: FastAPI
- AI/ML: LangChain, RAG, OpenAI API, Upstage Solar, Gemma2, Phi 3.5, Llama 3.1
- Speech: OpenAI Whisper, ElevenLabs TTS

4. Modular RAG 연구 (2024.10 ~ 2024.12)

역할: 1인 개발

연구 주제:

1. Modular RAG 구조 설계
2. Chat History 관리 (컨텍스트 윈도우 최적화)
3. RDBMS vs NoSQL 비교 연구
4. Short-term & Long-term Memory 구조 설계

기술 역량 상세

AI/ML

- **LLM 통합:** OpenAI API (GPT-3.5/4/4o), Claude, Private Model, Upstage Solar, Gemma2, Phi 3.5, Llama 3.1
- **LLM 프레임워크:** LangChain, LangGraph (멀티 에이전트 시스템)
- **RAG:** 하이브리드 검색 (Dense + Sparse), Multi-Query 생성
- **ML 라이브러리:** PyTorch, LightGBM, XGBoost, CatBoost, scikit-learn
- **Computer Vision:** YOLOv8, SORT, EasyOCR, ZoeDepth, MiDaS, OpenCV
- **Speech:** OpenAI Whisper, ElevenLabs TTS, TypeCast TTS

Backend

- 언어: Python
- 프레임워크: FastAPI, Flask
- 비동기 처리: Celery, Redis
- **ORM:** SQLAlchemy, Kysely (TypeScript), Drizzle ORM
- 시계열 분석: statsmodels, NumPy, Pandas

Frontend

- 프레임워크: Next.js (App Router), React 19
- 언어: TypeScript, JavaScript
- 에디터: BlockNote Editor
- **UI:** Flutter (모바일)

Database

- 관계형 **DB:** PostgreSQL, MySQL
- 벡터 **DB:** Qdrant (하이브리드 검색, 컬렉션 관리), pgvector, Chroma
- **Graph DB:** Neo4j
- 캐시: Redis (캐싱 레이어, TTL, LRU)

DevOps & Infrastructure

- 컨테이너: Docker, Docker Compose
- 웹서버: Nginx, Gunicorn

- **인증:** Keycloak (SSO, OAuth 2.0), better-auth
- **모니터링:** LangSmith, DataDog
- **Deployment:** Vercel, Railway, [Fly.io](#), AWS, GCP

기타

- **실시간 통신:** Server-Sent Events (SSE), WebSocket
- **임베딩:** OpenAI Embeddings, Upstage, LlamaParse, Docling, Unstructured
- **문서 파싱:** pypdf, docx2txt, python-pptx, pdf2image
- **텍스트 분할:** RecursiveCharacterTextSplitter
- **시각화:** Matplotlib, Seaborn, Plotly Dash
- **버전 관리:** Git, GitHub
- **패키지 관리:** Poetry, Uv, Bun
- **Linting:** Biome
- **Collaboration:** Notion, Slack, Jira

소프트 스킬

프로젝트 관리

- **대규모 팀 리딩:** BCG RA 15명 + 랩큐 팀원 3명 (SK PharmaAIX)
- **1인 개발 → 팀 PL 성장:** 한국자동차연구원 (1인 → 3명 팀)
- **요구사항 분석, 일정 관리, 팀원 코드 리뷰**
- **200+ 테스트 케이스 작성으로 체계적인 품질 관리**

이해관계자 관리 및 조율

- **멀티 팀 협업:** 약 10개 팀과 소통 (추진팀, 인프라팀, 보안팀, QA팀, 백엔드, 프론트 등)
- **빈번한 회의 조율:** 각 팀 요구사항 조율 및 일정 관리
- **기술 부채 관리:** 레거시 개선 + 신규 기능 개발 동시 진행
- **품질과 속도 균형:** 빠른 오픈 일정 속에서도 99% 만족도 달성

아키텍처 설계

- **Multi-Agent Supervisor 구조:** SK PharmaAIX (Chain → Agentic 전환)
- **LangGraph 멀티 에이전트:** 한국자동차연구원 (Research → Report → Supervisor)

- **RESTful API 설계:** Flask → FastAPI 마이그레이션
- **Docker 기반 마이크로서비스:** 멀티 컨테이너 환경 구축

문제 해결

- **성능 최적화:** 응답속도 83% 개선, 답변 커버리지 233% 향상
- **기술적 문제 분석:** LLM 고유명사 오타기 → 토큰나이저 레벨 분석 및 해결 제안
- **Context Engineering:** 프롬프트 과도 증가 문제 인식 및 개선 제안
- **하이브리드 검색:** Dense + Sparse 검색 결합
- **실시간 스트리밍:** SSE 기반 응답 시스템 구축

오픈소스 기여 및 학습

- **LangChain Open Tutorial:** 2,000+ 커밋 대규모 프로젝트 기여
- **3개 튜토리얼 작성:** LlamaParse, Memory Management, Smart Web Search
- **Peer Review 경험:** 협업 및 기술 인사이트 확장
- **최신 트렌드 파악:** RAG, Agent, MCP 등 꾸준한 학습

꾸준한 개발 습관 및 기술 공유

- **GitHub 일일 커밋 습관:** 25+ repositories, 매일 무언가를 개발하는 습관 유지
- **기술 블로그 운영:** Quartz 기반 개인 기술 블로그 (<https://syshin0116.github.io>)
 - 프로젝트 회고, 기술 학습 내용, 문제 해결 과정 지속적 기록
 - 학습한 내용을 문서화하고 공유하는 습관
- **지속적 학습:** 새로운 기술 스택 및 트렌드를 빠르게 습득하고 적용

테디노트 RAG팀에 기여할 수 있는 부분

1. 다양한 도메인의 RAG 프로젝트 수행

- **경험:** 자동차, 뉴스, 데이터 분석, 제조, Computer Vision 등 다양한 도메인 프로젝트 경험
- **기여:** 고객사 문제 상황 빠른 이해 및 맞춤형 RAG 파이프라인 설계 가능

2. 성능 고도화를 위한 평가지표 수립

- **경험:** 하이브리드 검색, 멀티 파서 성능 비교, Interactive Highlight 검증

- 기여: 문제상황에 맞는 평가 기준 마련 및 고도화 방안 제시

3. RAG & Agent 선행리서치

- 경험: LangChain Open Tutorial 컨트리뷰터, Modular RAG 연구
- 기여: 논문 리뷰, 내부 세미나, 프로젝트 연구 내용 정리 및 내재화

4. SLM Training (추가 학습 의지)

- 현재 수준: Embedding/Re-ranker 개념 이해, Fine-Tuning 경험은 없음
- 기여: 빠른 학습 및 적용 능력, ML 라이브러리 (PyTorch 등) 경험 보유

관련 링크

- **LangChain Open Tutorial:**
 - GitBook: <https://langchain-opentutorial.gitbook.io/langchain-opentutorial>
 - GitHub: <https://github.com/LangChain-OpenTutorial/LangChain-OpenTutorial>
- **GitHub:** [개인 GitHub 링크 - 추가 필요]
- **포트폴리오:** [포트폴리오 링크 - 추가 필요]

희망 연봉

- **희망 연봉:** [추후 작성]
 - (복지금액 포함)

지원 동기 및 포부

테디노트 LAB RAG팀은 소수정예 팀으로 다양한 도메인의 RAG 프로젝트 수행과 선행연구를 병행한다는 점에서 매우 매력적입니다.

저는 1년 8개월간 한국자동차연구원 AI 에이전트 프로젝트를 통해 **Production Level RAG 시스템**을 설계, 구현, 운영한 경험이 있으며, **LangChain/LangGraph** 기반 멀티 에이전트 시스템 설계에 강점을 가지고 있습니다.

특히, 하이브리드 검색 도입, 멀티 파서 통합 및 성능 비교, 실시간 스트리밍 응답 등 성능 고도화 경험을 바탕으로 테디노트 RAG팀에서 고객사 문제 해결과 선행연구에 기여할 수 있을 것으로 확신합니다.

또한, **LangChain Open Tutorial** 오픈소스 기여 경험을 통해 최신 트렌드를 꾸준히 파악하고 있으며, **Peer Review** 및 협업 경험을 바탕으로 팀원들과 원활한 의사소통과 협업이 가능합니다.

테디노트 RAG팀에서 다양한 도메인의 **RAG** 프로젝트를 수행하며 실무 역량을 더욱 발전시키고, 선행연구를 통해 **RAG** 기술의 본질을 파악하는 엔지니어로 성장하고 싶습니다.

감사합니다.

작성일: 2025년 11월

작성자: 신승엽