

# Cover Letter

류지환 (Jihwan Ryu)

## AI Agent Engineer 지원

지원 회사: Upstage

ryoo0504@gmail.com

### 1. 지원 동기

업스테이지의 Document Intelligence가 보험 클레임 처리에서 95% 이상의 정확도를 달성했다는 사례를 보면, "LLM을 실제 비즈니스 문제 해결에 적용 가능한 제품으로 구현한다"는 AI Product Division의 방향성에 깊이 공감했습니다. 특히 JD에서 강조하는 "진짜 고객의 문제를 해결"한다는 표현이 인상적이었습니다. 연구 성과를 논문으로 끝내지 않고, 실제 Fortune 500 기업과 보험사가 사용하는 제품으로 전환하는 과정에 참여하고 싶어 지원합니다.

저는 Eco<sup>2</sup> 프로젝트에서 재활용 분류 도메인의 Multi-Agent 시스템을 설계하며 비슷한 문제를 경험했습니다. 사용자가 "이거 어떻게 버려?"라고 물으면, 시스템은 이미지를 분류하고, 18개 분류 규정에서 관련 정보를 검색하고, 사용자 위치 기반으로 수거장소를 안내해야 합니다. 단순한 LLM 호출이 아니라, Intent 분류 → RAG 검색 → 품질 평가 → 응답 생성의 전 구간에서 정확도를 관리해야 했습니다. 이 과정에서 Confidence Scoring 체계를 설계하고, RAG 품질이 낮을 때 Web Search로 Fallback하는 정책을 구현하며 "정확도가 곧 사용자 신뢰"라는 점을 체감했습니다.

업스테이지의 Information Extract가 인보이스, 클레임, 계약서에서 key-value를 추출하는 문제는, 제가 Eco<sup>2</sup>에서 폐기물 규정 문서에서 분류 기준을 추출하던 문제와 구조적으로 동일합니다. 다만 업스테이지가 다루는 보험·금융 도메인은 오류 비율이 훨씬 크고, 규제 요건이 엄격하며, 문서 형식의 변이가 큽니다. 이러한 고난도 환경에서 Agent 기반 제품을 개발하고, 벤치마크를 설계하며, 고객 피드백을 반영해 정확도를 높이는 경험을 통해 엔지니어로서 한 단계 성장하고 싶습니다.

### 2. 관련 경험

#### Agent 시스템 설계 및 구현

- LangGraph StateGraph 기반 10개 전문 서브에이전트 설계 (waste\_rag, location, bulk\_waste, weather 등)
- Send API를 활용한 동적 병렬 라우팅으로 복합 질의를 단일 요청에서 처리
- Intent-Based Enrichment: 주요 Intent에 weather, web\_search 등 보조 컨텍스트 자동 주입

#### RAG 파이프라인 설계 및 튜닝

- Anthropic Contextual Retrieval 패턴 기반 2-tier 검색 (18개 분류 규정 + 80개 상황 태그)
- Classification → Keyword Fallback 전략, Relevance scoring으로 chunk 우선순위 결정
- Feedback Node가 RAG 품질을 LLM으로 평가, 저품질 시 Web Search → General LLM Fallback Chain

## Function Calling / Tool Use

- Location Agent: Native SDK(OpenAI/Gemini) Function Calling으로 Kakao Local API 도구 호출
- 8개 공공/외부 API 통합 (MOIS, KECO, KMA, Kakao Local, DuckDuckGo/Tavily)

## 품질 평가 및 벤치마크

- Intent Confidence Scoring: LLM 신뢰도 + 키워드 매칭 보정 + Chain-of-Intent 전이 보너스
- FeedbackQuality 점수 기반 응답 품질 평가, 0.6 미만 시 Fallback 트리거
- k6 부하 테스트로 VU 50→500(x10), RPM 60→528(x8.8) 스케일링 검증

## 프롬프트 엔지니어링

- Global/Local Prompt Optimization 패턴: 캐릭터 정의(Global) + Intent별 지침(Local) 분리
- 구조화 응답 템플릿으로 할루시네이션 억제, LRU 캐싱으로 프롬프트 로드 최적화

## 3. 기술 스택

Agent Framework	LangGraph, OpenAI Agents SDK
LLM	GPT-5.2, Gemini-3.0-flash
Backend	Python 3.11, FastAPI, Celery, Gevent, Taskiq
Infrastructure	Kubernetes (24-Node), Redis, PostgreSQL, RabbitMQ
Observability	Prometheus, Grafana, Jaeger, OpenTelemetry, LangSmith

## 4. 입사 후 기여 방향

- Agent 파이프라인 설계: 문서 워크플로우 특화 Multi-Agent 아키텍처 설계
- 품질 평가 체계 구축: RAG 품질 벤치마크 및 Confidence 기반 Fallback 정책 수립
- 프로덕션 안정성: Circuit Breaker, Checkpoint, Resilience 패턴 적용

2026년 1월

류지환 드림