## 삼성디스플레이 생성형 AI (Chat Pro) 구축

**1. **프로젝트 개요:****

본 문서는 삼성디스플레이 내부 사용자를 위한 LLM 기반 생성형 AI 챗봇인 "Chat Pro" 시스템 구축 프로젝트의 핵심 내용과 요구사항을 정리한 자료입니다.

기존 Open WebUI (v0.6.7)를 기반으로 사내 특화 기능을 추가 개발하고,

다양한 내부 시스템 및 외부 서비스와의 연동을 통해 사용자 편의성과 정보 접근성을 극대화

**2. **핵심 목표:****

* Open WebUI 기반 사내 특화 기능 개발 및 시스템 통합.
* 사용자 질의에 대한 정확하고 신뢰성 있는 정보 제공 (RAG 기반).
* 외부 서비스 연동을 통한 LLM 기능 확장 (지도/날씨, 실시간 검색, Wiki 등).
* 내부 시스템(DB, EDM 등) 연동을 통한 사내 정보 활용 극대화.
* 사용자 편의성을 고려한 UI/UX 개선 및 기능 제공.
* 금칙어 필터링 등 컴플라이언스 준수 기능 구현.

**3. 주요 구성 요소 및 기술 스택:**

* **App 서버:** Python, Openweb ui (v0.6.7), Filter (금칙어 Compliance), N8N.io, Embedding 모델, Rerank 모델, Jupyter, Apache Tika, Tesseract OCR, Langchain, Pipeline, Docling, Grafana, Searxng
* **DB 서버:** Postgre, Elastic Search (VectorDB)
* **인프라:** Docker 12.8, Red hat Linux
* **LLM 모델:** Llama 405B (X), mistral large, Qwen2.5-72B-Instruct, Llama 70B

(사용자 선택 가능)

**4. 주요 기능 요구사항:**

****4.1 사용자 UI 주요 기능 (Open WebUI 기반):****

* **모델 선택:** 지원되는 다양한 LLM 모델 중 선택 가능.
* **프롬프트 템플릿:** 요약, 보고서 작성 등 미리 정의된 템플릿 선택 기능.
* **번역, 요약, 메일 초안 작성, 보고 작성:** 기본 기능 지원.
* **지식 (RAG 선택):**지식명 입력 및 적용.
* "캠퍼스별 업무 담당자 정보", "캠퍼스별 시설물 정보", "삼성 용어 사전" 등 사내 지식 선택 (최대 4개).
* **파일 첨부:**\*.txt, .doc, .docx, .xls, .xlsx, .csv, .ppt, .pptx, .pdf 확장자 지원.
* 총 용량 10MB 이내 제한 (txt는 3MB 이내).
* **사용자 LNB/TNB (RNB) 창 기능:** 새 채팅, 노트, Q 검색, 채팅 목록, 채팅 제어 (밸브), 시스템 프롬프트 설정, 고급 매개변수 설정 (스트림 응답, 함수 호출, 온도, Top K 등), 로그아웃, 권한.

****4. 2. 사용자 기능 추가 개발 요구사항 (핵심 기능):****

* **Lv1 기능 보완 : 금칙어 필터 기능:**

- 프롬프트 질의 및 답변 내용에 대한 금칙어(외설, 폭력, 혐오, 차별, 불법 조장 용어 등) 필터링.

- 금칙어 포함 시 해당 항목 고지 후 답변 불가 처리.

* **외부 플러그인 (Tools API 호출):**지도/날씨 정보 제공 (예: Open WebUI Community Tool 연동).

- 실시간 검색 엔진 연동 (예: Searxng 활용 가능).

- Wiki 검색 기능 연동 (예: Wikidata Tool 연동).

- "LLM 내부에 지도/날씨, 실시간 검색, WIKI 등 외부 플러그인(Tools API 호출) 기능을 연결, 정보나 실시간 데이터에 접근하여 확장.

→ "도구 사용(Tool Use)" , "함수 호출(Function Calling)"

"사용자 질의가 들어왔을 때, 이 질의가 어떤 정보...를 요구하는지 파악하고,

가장 적절한 검색 방법(DB, FTS, Vector)과 대상 인덱스/DB 테이블을 결정하는 단계"

* **EDM 검색 결과 List 제공:**질의 내 EDM 검색 후 결과 목록 노출.

- List 내에서 요약 및 번역 버튼 노출 (EDM API 연계 및 색인 후 요약/번역).

* **회사생활 Assistant (RAG):**사내 전화번호, 담당자 (DB 검색 기반 RAG).

- 인사 규정 (파일/문서 기반 RAG).

- IT Helpdesk (RAG 구성).

- 출/퇴근, 파우치 버스 (RAG 구성).

- 시스템 매뉴얼 (파일/문서 기반 RAG).

→ "사내 전화번호, 담당자, 인사 규정, IT Helpdesk, 출/퇴근, 파우치 버스, 시스템 매뉴얼 등 다양한 형태의 내부 지식을 활용하는 RAG(Retrieval Augmented Generation) 시스템 구축"

* **요약/번역:** 긴 문서 최대치 및 제한치 제공 (컨텍스트 크기 관리 중요).
* **메일 초안 작성:** 기본/회의/부재 템플릿 활용.
* **보고서 초안 작성:** 템플릿 활용.
* **Lv2 기능:**메일 초안 작성 (완성도 향상).

- 보고서 초안 작성 (완성도 향상).

- 문서 비교 (2~3개 문서 분석 비교).

* **Lv3 기능:**Office Add-In Component 개발 (Word, PPT, Excel).

- Word/PPT: 보고서 초안 작성 외 추가 Use Case 지원.

- Excel: 데이터 추이 분석 외 추가 Use Case 지원.

- 템플릿 정확도 증가 및 길이 제한 가이드.

- 답변 정확도 증가, 내용량 300자 이상 등.

- 통/번역 길이 제한 및 가이드 제시.

****4.3. RAG 시스템 구축 (사내 지식 활용):****

* **데이터 수집 및 전처리:**

- 각 데이터 소스 식별 (DB, 문서 파일, FAQ 등), 데이터 추출, 정제 및 구조화, 텍스트 분할(Chunking) 및 메타데이터 저장.

* **인덱싱:**

- DB 인덱싱, FTS (Full-Text Search) 인덱싱 (Elasticsearch 등), 벡터 인덱싱 (Embedding & Vector Store - Elastic Search VectorDB 활용).

* **질의 분석 및 라우팅:** 사용자 질의를 분석하여 필요한 정보 및 검색 방법(DB, FTS, Vector) 결정. 키워드 규칙, 의도 분류, LLM 기반 라우팅 등의 방법 사용 가능.

→ "사용자 질의가 들어왔을 때, 이 질의가 어떤 정보...를 요구하는지 파악하고, 가장 적절한 검색 방법...과 대상 인덱스/DB 테이블 결정"

* **정보 검색 (Retrieval):**

- 라우팅 결과를 바탕으로 DB 검색기, FTS 검색기, 벡터 검색기 호출. 하이브리드 검색 (FTS + Vector) 고려.

* **결과 통합 및 순위 재조정:**

- 여러 검색 결과 통합 및 최적화.

* **응답 생성 (Generation):**

- 검색 결과와 사용자 질의를 LLM에 전달하여 최종 답변 생성.

* **컨텍스트 크기 초과 방지:**

- 텍스트 분할, 검색 결과 제한, 계층적 요약 등의 전략 활용.

→ "거대 언어 모델 API는 입력 텍스트의 길이에 제한으로 문서를 직접 API에 전달하면 이 제한을 초과에 대한 방안 수립"

****4.4. 외부 서비스 연동 (번역 및 요약):****

* **문서 불러오기:**

- 문서 저장소 종류에 따라 적절한 방법 사용 (클라우드 스토리지 SDK, 파일 시스템, DB 커넥터, 플랫폼 API 등).

* **번역 기능:**

- 번역 API 선택 (Google Cloud Translation, Papago, DeepL 등), API 호출 함수 작성.

* **요약 기능:**

- LLM API 또는 Summarization API 선택 (KoGPT, PaLM, Gemini 등), API 호출 함수 작성.

* **컨텍스트 크기 초과 방지:텍스트 분할:**

- 문서를 적절한 크기의 청크로 분할.

* **분할 및 병합:**

- 각 청크를 개별 번역/요약 후 결과 합침.

* **슬라이딩 윈도우:**

- 문맥 유지를 위해 앞뒤 청크 일부 포함 처리.

* **계층적 요약:**

- 섹션별 요약 후 전체 요약.

****5. 구축 단계****

****프**로젝트 일정 계획**

프로젝트는 총 6단계로 나누어 진행되며, 각 단계별로 예상 소요 기간을 설정

전체 프로젝트 기간은 약 15주로 계획

|  |  |  |
| --- | --- | --- |
| 단계 | 작업 내용 | 소요 기간(Week) |
| 1 | 프로젝트 설정 및 기반 구축 | 2 |
| 2 | 쿼리 분석 모듈 개발 | 3 |
| 3 | 라우팅 모듈 개발 | 2 |
| 4 | 검색기 모듈 개발 | 3 |
| 5 | 결과 처리 모듈 개발 | 2 |
| 6 | 통합 및 최적화 | 3 |

**1**단계: 프로젝트 설정 및 기반 구축

작업 1: 프로젝트 구조 설정

작업 2: 의존성 패키지 설정

작업 3: 환경 설정 파일 구현

작업 4: 로깅 설정 구현

2단계: 쿼리 분석 모듈 개발

작업 5: 의도 분류기 인터페이스 설계

작업 6: OpenAI Function Calling 구현

작업 7: 키워드 추출 로직 구현

작업 8: 규칙 기반 쿼리 유형 분류 구현

작업 9: 쿼리 분석기 통합 구현

작업 10: 쿼리 분석 단위 테스트 구현

3단계: 라우팅 모듈 개발

작업 11: 라우팅 전략 설계

작업 12: 라우터 클래스 기본 구현

작업 13: 라우팅 전략 구현

작업 14: 하이브리드 검색 전략 구현

작업 15: 라우팅 모듈 단위 테스트 구현

4단계: 검색기 모듈 개발

작업 16: 검색기 기본 클래스 구현

작업 17: DB 연결 및 초기화 구현

작업 18: DB 검색기 구현

작업 19: Elasticsearch 연결 및 초기화 구현

작업 20: FTS 검색기 구현

작업 21: Milvus 연결 및 초기화 구현

작업 22: 임베딩 유틸리티 구현

작업 23: 벡터 검색기 구현

작업 24: RRF 알고리즘 구현

작업 25: 하이브리드 검색기 구현

작업 26: 검색기 모듈 단위 테스트 구현

5단계: 결과 처리 모듈 개발

작업 27: 결과 포맷팅 유틸리티 구현

작업 28: 결과 재순위화 모듈 설계

작업 29: 결과 재순위화 구현

작업 30: 결과 처리 모듈 단위 테스트 구현

6단계: 통합 및 최적화

작업 31: 메인 파이프라인 구현

작업 32: 샘플 데이터베이스 스키마 및 데이터 생성

작업 33: Elasticsearch 인덱스 및 샘플 데이터 생성

작업 34: Milvus 컬렉션 및 샘플 벡터 생성

작업 35: 통합 테스트 구현

**인력 투입 계획**

프로젝트 총괄 : 1명 - 전체 프로젝트 관리 및 아키텍처 구축 (2MM)

백엔드 개발자 : 2명 - Python 기반 모듈 개발 및 API 구현 (6MM)

UI 개발자 1명 : 1명 - WEB UI 개선 (3MM)

데이터 엔지니어 : 1명 - 데이터베이스 및 검색기 연동, 데이터 처리 (1MM)

****주요 고려사항:****

* **데이터 최신성 유지:**

- RAG 데이터(DB, FTS, Vector 인덱스)의 최신성 유지를 위한 업데이트 파이프라인 구축 필요.

* **검색 정확도 평가 및 개선:**

- 각 검색 방법의 정확도 평가 및 인덱싱 전략, 검색기 설정 튜닝 필요.

* **라우팅 정확도:**

- 질의 분석 및 라우팅 정확도가 시스템 성능에 큰 영향. 지속적인 개선 필요.

* **보안 및 권한 관리:**

- 사용자별 정보 접근 권한 제한 로직 필요.

* **성능 (응답 속도):**

- 비동기 처리, 캐싱, 인덱스 최적화 등을 통해 성능 개선 필요.

* **LLM 모델 선택:**

- 사용 모델의 함수 호출 기능 지원 및 컨텍스트 길이 확인 필요.

* **API 키 관리:**

- 외부 서비스 API 키 안전 관리 필수.

* **외부 API 비용:**

- 상용 API 사용 시 비용 고려 및 모니터링 필요.

* **오류 처리:**

- 다양한 예외 상황에 대한 견고한 오류 처리 로직 필요.

→ 핵심 성공 요소 **:**

* 삼성디스플레이 생성형 AI (Chat Pro) 구축 프로젝트는 Open WebUI를 기반으로 강력한 사내 특화 기능과 외부 서비스 연동을 통해 사용자에게 고도화된 AI 경험을 제공을 목표로 함
* - RAG 시스템 구축을 통해 사내 지식 활용 극대화
* - 금칙어 필터링을 통해 컴플라이언스 준수.
* - 3단계 구축 작업은 RAG 시스템의 핵심으로, 효율적인 개발 계획 수립 및 진행
* - 각 단계별 테스트와 성능 최적화
* 안정적이고 유용한 Chat Pro 시스템 구축