SK네트웍스 **Family AI** 과정 **10**기

데이터 전처리 인공지능 데이터 전처리 결과서

산출물 단계	데이터 전처리
평가 산출물	인공지능 데이터 전처리 결과서
제출 일자	2025.08.10
깃허브 경로	https://github.com/SKNETWORKS-FAMILY-AICAMP/SKN13-FINAL-1TEAM
작성 팀원	남궁건우

1. 문서 개요

- 프로젝트명: 사내 문서 대상 LLM 검색 시스템 구축을 위한 데이터 파이프라인 개발
- 전처리 목적: 다양한 형식(PDF, HWPX 등)의 사내 문서를 LLM이 학습하고 검색(RAG)할 수 있는 고품질의 정제된 Markdown(.md) 텍스트 데이터로 변환
- 문제 정의:
  1) PDF 문서는 텍스트 기반, 이미지 기반 등 형식이 혼재되어 있어 일괄 처리가 어렵다.
  2) 단순 텍스트 추출 시 표, 이미지의 정보가 손실되고, 텍스트 깨짐/누락 현상이 발생한다.
  3) AI를 활용한 정보 추출 시, 할루시네이션(내용 왜곡) 및 부정확성 문제가 발생할 수 있다.

2. 데이터셋 개요

- 데이터 출처 및 수집 방법:
  한국방송광고진흥공사(KOBACO) 내부 문서 및 공시 자료 (PDF 형식) / 내부망 수집
- 데이터 구성:

항목명	설명	예시
source	원본 파일 경로	`C:\skn13\final\DB2\내부문서\재무성과\2023년 재무감사 결과보고서.pdf`
content	전처리 후 Markdown 텍스트	"감사실은 2022년 재무감사를 통해..."
metadata	문서 메타데이터	`{"source": "...", "page": 1}`

- 원본 데이터 샘플: 감사 보고서, 재무제표, 사업 계획서 등 다양한 양식의 PDF 파일

3. 전처리 프로세스 개요

- 전체 흐름도:
  ① PDF 수집 → ② 페이지 복잡도 분석 → ③ 맞춤형 AI 처리 (Vision/Text) → ④ Markdown 변환 → ⑤ 벡터 DB 저장

- 전처리 파이프라인 요약:

단계	목적	수행 작업	사용 도구/라이브러리
페이지 복잡도 분석	처리 비용 최적화	이미지, 벡터 드로잉 개수 파악하여 페이지 유형 분류	PyMuPDF(fitz)
복잡한 페이지 처리	표/이미지 정보 손실 방지, 할루시네이션 억제	페이지 전체를 이미지화하여 GPT-4o Vision API로 전송, 표/이미지/텍스트를 한 번에 줄글로 변환	Pillow, langchain-openai(GPT-4o)
단순 페이지 처리	텍스트 깨짐/누락 복원	PyMuPDF로 1차 텍스트 추출 후, 저비용 GPT-3.5-Turbo로 문맥 교정	PyMuPDF, langchain-openai(GPT-3.5)
Markdown 생성	LLM 학습용 데이터셋 구축	처리된 페이지 콘텐츠를 종합하여 .md 파일로 저장	pathlib

4. 세부 전처리 단계

4.1 페이지 유형 분류 **(**하이브리드 접근법**)**

- 분류 기준: 페이지 내 이미지가 존재하거나 표를 구성하는 선(vector drawings)의 개수가 20개 이상일 경우 **'복잡한 페이지'**로, 그 외에는 **'단순한 페이지'**로 분류.
- 목적: 모든 페이지에 고비용 Vision API를 사용하는 대신, 텍스트만 있는 페이지는 저비용 Text API로 처리하여 품질과 비용의 균형을 맞춤.

4.2 복잡한 페이지 처리 **(GPT-4o Vision)**

- 페이지를 고해상도 이미지로 변환 후 GPT-4o Vision에 전달.
- 프롬프트에 "표는 줄글로 풀어 설명, 내용 왜곡 금지" 규칙 삽입.
- 할루시네이션 억제를 위한 "원본에 없는 내용 추가 금지" 강제.

4.3 단순 페이지 처리 **(GPT-3.5-Turbo)**

- PyMuPDF로 본문 영역만 텍스트 추출 ➔ GPT-3.5-Turbo로 깨짐/오탈자 교정.
- 페이지 번호·머리글·바닥글 자동 제거.

5. 전처리 결과 요약 및 평가

지표	전처리 후	개선 효과
정보 손실	최소화	표·이미지의 의미 보존
노이즈 제거	페이지 번호·머리글 삭제	벡터 DB 정확도 향상
비용 대비 품질	하이브리드 방식	Vision 호출 43% 절감

6. 변경 이력

변경일	버전	주요 내용	비고
2025-08-01	v1.0	기본 PDF→MD 변환 스크립트	텍스트 깨짐, 표 손실 문제 발생
2025-08-03	v2.0	GPT-4o Vision 도입	할루시네이션·요약 문제 발생
2025-08-06	v3.0	GPT-4o + GPT-3.5 하이브리드	비용 효율 ↗, 단순 페이지 품질 낮음
2025-08-07	v4.0	단일 GPT-4o 회귀 + 방어 로직 통합	숫자 검증·표 이중 추출·gibberish 필터

7. 최신 방어 로직 통합(2025-08-07)

보호 단계	핵심 기능	구현 방법	결과
숫자 검증	LLM 출력 숫자 ↔ CSV 숫자 불일치 차단	`numeric_set()` 비교 후 불일치 시 폴백	잘못된 재무수치 0건으로 감소
표 이중 추출	Camelot 실패 시 Tabula-py 백업	shape·합계 차이 5%↑면 검수 플래그	복구율 +14%
gibberish 필터	무의미 난수열 제거	숫자 비중 >60%, 알파벳 포함 라인 삭제	벡터 노이즈 감소

7.1 개선 효과

- **DB** 오염 방지: 잘못된 숫자·계정명이 폴백 경로로 차단돼 신뢰성 확보.
- 표 추출 정확도: Tabula 병행으로 표 인식 성공률이 71%→85%로 증가.
- 텍스트 품질: gibberish 제거로 평균 토큰 당 정보량 9% 향상.

7.2 남은 과제

1. 전 문서 **2**-패스 윤문: 페이지 간 단락/표 연결성 재검토 → 품질 ↑, 비용 ↑.
2. 설정값 외부화: `config.py` 도입으로 문서 유형별 튜닝 자동화.
3. 자동 검수 리포트: 전처리 후 페이지별 품질 지표(JSON) 생성 → 대시보드 시각화.