기술 과제: **Docx** 문서 파서

주제: 로컬 LLM 기반 구조화 문서 파서 개발

1. 과제 개요

- 임의의 .docx 문서를 입력으로 받아, 그 내용을 읽을 수 있는 Markdown 포맷의 텍스트로 변환하되, 모든 내용은 원본 위치 정보와 함께 JSON 객체 내부에 포함된 구조로 출력하는 시스템을 구현
- 단순 텍스트는 문단 단위로 그대로 Markdown 본문에 포함
- 표는 로컬 LLM을 사용해 요약 설명 또는 마크다운 표로 변환
- 이미지는 OCR 결과와 함께 설명 문장 생성 후 Markdown에 삽입
- 모든 요소는 원본 문서 내 위치 정보(단락 번호, 페이지, 테이블 인덱스 등)를 JSON 객체 내부에 포함된 구조로 출력

2. 처리 대상 및 방식

구성 요소	처리 방식
일반 텍스트	문단 단위로 Markdown 문자열 생성 → "markdown" 필드에 저장. "paragraph_index" 등 원본 위치 정보 포함
표 (Table)	표 전체를 LLM 입력으로 넣어 설명 Markdown을 생성. "table_index"와 함께 저장
이미지 (Image)	이미지 추출 후 포함한 Markdown 설명 생성. OCR 결과 및 LLM 설명 포함. "image_index" 및 위치 기록
OCR 대상 텍스트	이미지 내 텍스트를 pytesseract 등이 추출 → "ocr_text" 및 "markdown" 필드로 출력

3. 기술 조건

항목	내용
사용 라이브러리	python-docx, pytesseract, transformers, jinja2 등 자유
모델 조건	Open-source LLM만 허용 (예: MiniGPT-4). 외부 API 호출 금지
실행 시간	전체 파이프라인 3분 이내 권장

4. README 작성 항목 (필수)

README 파일에는 다음 내용을 포함해야 합니다:

1. 설치 방법
    - requirements.txt 기반 환경 설치 안내
    - 모델 로딩이 필요한 경우 체크포인트 정보 포함

2. 실행 방법
    - 단일 명령 또는 run.sh 스크립트 안내
    - 입력 문서 예시 포함 시 명시

3. 시스템 설명
    - 파싱 방식
    - 의미 분류 로직 설명 (LLM 사용 시 프롬프트 구조 포함)
    - Markdown 생성 구조

4. 데모 결과 캡처 또는 예시 출력
    - output.json 예시 출력 포함

5. 제출

- 제출 기한: 과제 수령 후 3일 이내
- 제출 형식: zip 파일 또는 GitHub 링크
- gwanghoon@mind.ai로 제출