문서생성 자동화 방안

# 1. 개요

프로젝트 제안 및 제품 소개자료를 작성할 때 반복적으로 수행되는 문서 작업을 인공지능(AI) 기반 자동화 기술(Ax)을 활용하여 효율화할 수 있는 방안을 검토하였습니다.

이를 통해 업무 생산성을 향상시키고, 인적 자원 활용의 효율성을 높이는 것을 목표로 합니다.

# 2. 고려사항

## 2.1 인프라 환경

## 2.1.1 오픈된 LLM을 사용할 수 있는 환경

ChatGPT, Claude, Gemini와 같은 외부의 대형 언어모델(LLM)을 활용할 수 있는 환경에서는 API 호출을 통해 문서 자동화를 구현할 수 있습니다. LangChain이나 LlamaIndex와 같은 프레임워크를 활용하면 구조화된 문서 생성을 손쉽게 자동화할 수 있습니다. 이 방식은 최신 AI 기술을 바로 활용할 수 있다는 장점이 있으나, 데이터가 외부로 유출될 수 있는 보안상의 우려가 존재합니다.

## 2.1.2 폐쇄망 환경

망 분리 등으로 외부와 완전히 차단된 환경에서는 GGUF 형태의 로컬 모델(EXAONE, LLaMA 등)을 로컬에 설치하여 오프라인으로 운용해야 합니다. 벡터 데이터베이스도 Chroma나 FAISS를 사용해 로컬에 구축함으로써 RAG(Retrieval-Augmented Generation) 기반 문서 자동화가 가능합니다. 이러한 방식은 보안에 매우 유리하지만, 처리 속도나 성능에서 제약이 있을 수 있습니다.

## 2.2 보유데이터&문서의 보안

보안이 중요한 환경에서는 외부 LLM을 사용할 수 없기 때문에, 로컬에서 실행 가능한 LLM을 사용하는 방식을 고려해야 합니다. 예를 들어 LM Studio나 Ollama 등의 도구를 통해 로컬 환경에서 LLM을 실행할 수 있으며, 사내 서버에 API 형태로 배포하는 방식도 가능합니다. 이러한 방식을 통해 보안을 유지하면서도 AI 자동화를 구현할 수 있습니다.

## 2.3 활용가능 자원

## 2.3.1 노트북 등 최소 장비 활용

소형 장비만 있는 경우에는 양자화된 모델을 활용하여 노트북에서도 실행이 가능한 자동화 시스템을 구축할 수 있습니다. 예를 들어 Q4\_K\_M 수준으로 양자화된 EXAONE 모델을 활용하면, 로컬에서 PDF를 읽고 프롬프트를 생성하여 문서화하는 작업이 가능합니다.

## 2.3.2 AX 개발 환경 구축 을 위한 시간과 인력의 규모

각 자동화 방식에 따라 필요한 시간과 인력의 수준이 다릅니다. 외부 LLM을 활용할 경우 구현까지 1~2일이 소요되며, 초급 수준의 인력도 가능하지만, 폐쇄망에서 RAG를 구성하려면 약 5~7일과 중급 이상의 기술력이 필요합니다. Fine-tuning은 수 주에 걸친 시간과 고성능 GPU 및 고급 인력이 요구됩니다.

# 3. 현재 환경을 고려한 개발 방안

## 3.1 Few-shot Prompting (로컬, 노트북 활용)

PDF 문서에서 텍스트를 추출하고, 사전에 정의된 프롬프트 템플릿을 기반으로 질문-답변 형태의 문서를 구성하였습니다. 로컬 환경의 LM Studio와 같은 도구를 활용하여 LLM에게 텍스트 기반 응답을 생성하게 하였으며, 결과는 자동으로 PPT로 변환되었습니다. 이 방식은 인터넷 없이도 노트북 단독으로 수행이 가능하다는 장점이 있습니다.

## 3.2 RAG - Vector DB 구축 (로컬, 노트북 활용)

문서 내용을 벡터 형태로 변환하여 FAISS나 Chroma 벡터 DB에 저장하고, LLM이 질문을 받을 때 해당 벡터 정보를 검색하여 응답을 생성하는 구조를 구현하였습니다. 임베딩 모델로는 Instructor-XL이나 MiniLM을 사용하였으며, 실제 문서 기반 제안서 생성에서 효과적으로 작동함을 확인하였습니다.

## 3.3 Fine-tuning

Fine-tuning은 모델을 새롭게 학습시키는 과정으로, 고품질의 대량 데이터, 수작업 라벨링, 고성능 장비(GPU), 숙련된 인력 등이 필요합니다. 또한 모델의 일반화 성능을 떨어뜨릴 위험이 있으며, 구축과 유지보수 비용이 매우 높습니다. 따라서 문서 자동화 목적에는 Few-shot 또는 RAG 방식이 훨씬 현실적인 대안입니다.

# 4. Few-shot Prompting

## 4.1 개발 환경

## - 로컬 LLM 설치

- EXAONE-3.5-2.4B-Instruct-Q4\_K\_M.gguf 활용

## - python-3.13.3

## 4.2 구현 절차

## 1) 자료 확보 및 전처리

- 제안서나 산출물 생성을 위한 기반 데이터 확보

* 발주처에서 제공한 **RFP(제안요청서)**, **참고 문서**, **기술 안내서**, **정책 설명서** 등 관련 자료 수집
* 자료 형식: PDF, HWP, DOCX 등 다양한 형태
* 수집된 자료를 읽을 수 있도록 텍스트화(PDF → 텍스트 추출, 이미지 OCR 처리 등)
* 중요 섹션(사업 개요, 요구사항, 일정, 기술 조건 등)을 구조화

## 2) 프롬프트 템플릿 생성 프로그램 활용

- LLM에게 효과적으로 질의할 수 있도록 프롬프트 구조 자동 생성

* 텍스트화된 문서 내용을 기반으로 프롬프트 템플릿 자동 생성
* 구성 요소:
  + title: 질문의 주제
  + question: LLM에게 물을 질문
  + answer: 참고용 예시 또는 초기 브레인스토밍 결과
* 예시:

{  
 "title": "시스템 구성",  
 "question": "시스템 구성요소는?",  
 "answer": "사용자 포털, 분석 플랫폼 등"  
 }

## 3) 팀 내부 브레인스토밍 및 템플릿 보완

- 초안 프롬프트를 실무적 관점에서 보완

* 팀원들과의 워크숍 혹은 리뷰 세션을 통해:
  + 질문의 맥락 부족 보완
  + 불명확하거나 중복된 질문 제거
  + 문서 특성과 맞지 않는 항목 조정
* LLM이 오답을 생성하지 않도록 질문 방식과 범위를 명확히 정의

## 4) LLM을 통한 콘텐츠 생성 요청

- 완성된 프롬프트 템플릿을 바탕으로 필요한 자료 생성

* 프롬프트를 LLM에게 입력 → 초안 생성
  + 제안서의 특정 항목 (예: 추진 배경, 시스템 구성, 일정 등)을 자동 작성
* 출력 포맷: 텍스트 문서, PPT, HTML 등 목적에 따라 다양화
* 필요 시 이미지, 표, 그래프 삽입도 병행하여 요청 가능

## 5) 결과 검토 및 반복 개선

- 정확하고 품질 높은 산출물 완성

* 생성된 결과물 검토 (내부 리뷰 + 실무 검수)
* 미흡한 부분 발견 시:
  + 템플릿 수정 또는 보완
  + 질문 수 재설계
  + LLM 설정값 조정 (max tokens, temperature 등)
* 최종 산출물 완성 전까지 반복 수행