요구사항 정의서

날짜	@August 10, 2025 12:13 PM
단계	개발 단계
유형	참고문서
작성자	㉿ 종철 이

FSKU LLM Challenge Model Product Requirements Document (PRD) (v4 - 최종 실행 계획본)

1. Goals and Background Context (목표 및 배경)

Goals (목표)

- [ ] 최종 점수 0.7점 이상을 달성하여 대회에서 우승합니다.
- [ ] RAG와 'Distill-M 2' 전략의 기술적 우수성을 입증합니다.
- [ ] 제한된 오프라인 환경에서 전체 추론 파이프라인을 안정적으로 실행합니다.
- [ ] 모든 데이터 출처와 증강 과정을 투명하게 기록하고 제출합니다.

Background Context (배경)

본 프로젝트는 '금융산업, 정보보호, IT 거버넌스' 등 광범위한 전문 분야에 대한 질문에 답변하는 고성능 LLM을 개발하는 과제입니다. 인터넷이 차단된 엄격한 하드웨어 및 시간 제약 조건 하에서, 사전 학습 데이터 없이 RAG와 지식 증류 기법을 활용하여 모델의 성능을 극대화하는 것을 목표로 합니다. 특히, 학습 과정의 리소스 제약이 없다는 규칙을 활용하여, 고성능 '교사 모델'로 생성한 합성 데이터셋으로 '학생 모델'을 학습시키는 전략적 이점을 가집니다.

2. Requirements (요구사항)

Functional Requirements (기능 요구사항 - FR)

1. **FR1:** 시스템은 `test.csv` 파일을 입력으로 받아 처리할 수 있어야 합니다.
2. **FR2:** 시스템은 `ID`와 `answer` 두 개의 컬럼을 가진 `submission.csv` 파일을 최종 결과로 생성해야 합니다.
3. **FR3:** 객관식 질문에 대해, 모델은 정답 보기의 '숫자'를 텍스트 형태로 생성해야 합니다.
4. **FR4:** 주관식 질문에 대해, 모델은 질문에 대한 서술형 답변을 텍스트로 생성해야 합니다.
5. **FR5:** 모든 답변은 RAG 시스템을 통해 검색된 정보를 바탕으로, 단일 LLM이 **새롭게 생성(Generate)**한 결과물이어야 합니다.

Non-Functional Requirements (비기능 요구사항 - NFR)

1. **NFR1 (성능):** 지정된 추론 환경에서 전체 테스트 데이터셋에 대한 추론을 **4시간 30분 이내**에 완료해야 합니다.
2. **NFR2 (이식성):** 모든 코드는 지정된 리소스 환경(Ubuntu 22.04, Python 3.10 등)에서 오류 없이 설치되고 실행될 수 있어야 합니다.
3. **NFR3 (최종 제출물):** 최종 제출 패키지에는 **소스 코드, 모델 가중치, 외부 데이터 증빙 자료, 결과 보고서**가 반드시 포함되어야 합니다.
4. **NFR4 (코드 구조): 추론(Inference) 코드는 `inference.py` 또는 `inference.ipynb`와 같이 반드시 별도의 파일로 작성**되어야 합니다.
5. **NFR5 (의존성 관리):** 모든 Python 라이브러리 의존성은 `requirements.txt` **파일에 정확한 버전과 함께 명시**되어야 합니다.
6. **NFR6 (데이터 증빙):** 사용된 **모든 외부 데이터, 전처리 코드, 그리고 데이터의 출처 및 라이선스 증빙 자료**를 제출해야 합니다.
7. **NFR7 (경로 및 인코딩):** 코드 내 모든 파일 입/출력 경로는 **상대 경로**여야 하며, 모든 코드와 주석은 **UTF-8**로 인코딩되어야 합니다.

---

3. Technical Assumptions (기술적 가정)

1. Repository Structure (저장소 구조): 모노레포 (Monorepo)

2. Service Architecture (서비스 아키텍처): 모듈형 ML 파이프라인 (Modular ML Pipeline)

3. Testing Requirements (테스트 요구사항): 단위 + 통합 테스트 (Unit + Integration)

---

4. Epic List (에픽 목록)

1. Epic 1: `기반 구축 및 데이터 파이프라인 (Foundation & Data Pipeline)`

- **Goal:** RAG 시스템을 위한 벡터 데이터베이스와 모델 학습을 위한 고품질 합성 Q&A 데이터셋을 생성하는, 재현 가능한 데이터 처리 파이프라인을 완성합니다.

2. Epic 2: `핵심 모델 파인튜닝 및 MVP 구현 (Core Model Fine-Tuning & MVP)`

- **Goal:** 'Distill-M 2' 전략의 1차 학습을 완료하고, RAG 시스템과 통합하여 MVP 성공 기준을 충족하는 End-to-End 추론 파이프라인을 구현합니다.

3. Epic 3: `성능 최적화 및 최종화 (Performance Optimization & Finalization)`

- **Goal:** LLM-as-Reranker 통합, 하이퍼파라미터 튜닝, 양자화를 통해 모델의 최종 점수를 목표치(0.7 이상)까지 극대화하며, 최종 제출 패키지를 완성합니다.

---

5. Epic 1: 기반 구축 및 데이터 파이프라인 (상세)

Story 1.1: `프로젝트 초기화 및 환경 설정`

- **As a** developer, **I want** to '모노레포' 구조를 설정하고 핵심 라이브러리 의존성을 정의하여, **so that** 모든 팀원이 일관되고 재현 가능한 개발 환경에서 작업할 수 있다.
- **Acceptance Criteria:**
  1. 모노레포 폴더 구조가 생성된다.
  2. `requirements.txt` 파일에 초기 라이브러리 목록과 버전이 명시된다.
  3. 기본 린터(linter) 및 포매터(formatter) 설정이 완료된다.

Story 1.2: `외부 데이터 수집 및 정제 파이프라인 구축`

- **As a** data engineer, **I want** to 원본 외부 데이터를 불러와 정제하고, 데이터의 출처를 기록하는 스크립트를 작성하여, **so that** 깨끗하고 신뢰성 있는 데이터를 확보할 수 있다.
- **Acceptance Criteria:**
  1. 스크립트는 `/data/raw`에서 원본 데이터를 읽는다.
  2. 스크립트는 기본 정제 작업을 수행하고 결과를 `/data/processed`에 저장한다.

3. 데이터 처리 과정을 추적할 수 있는 로그 또는 메타데이터가 생성된다.
4. `KoreanEnglishTextProcessor` 로직에 기반한 국영문 혼합 텍스트 정제 기능이 구현된다.
5. 저작권이 엄격한 ISO 표준 문서는 수집 대상에서 제외된다.
6. KOGL(공공누리) 데이터는 유형별 라이선스를 확인하고 기록한다.

Story 1.3: RAG 지식 베이스(FAISS) 구축

- **As a** data scientist, **I want** to 정제된 텍스트를 청크로 분할하고 벡터로 변환하여 FAISS 인덱스를 구축하여, **so that** RAG 시스템이 관련 정보를 빠르고 정확하게 검색할 수 있다.
- **Acceptance Criteria:**
  1. 스크립트는 텍스트를 적절한 크기의 청크로 분할한다.
  2. 임베딩 모델을 사용하여 각 청크를 벡터로 변환한다.
  3. FAISS 인덱스 파일이 디스크에 저장된다.
  4. 샘플 질문으로 인덱스가 정상 작동하는지 확인하는 테스트 함수가 포함된다.

Story 1.4: 합성 Q&A 학습 데이터셋 생성

- **As a** machine learning engineer, **I want** to 고성능 '교사 모델'을 사용하여 고품질 Q&A 쌍을 대량으로 생성하여, **so that** '학생 모델'을 파인튜닝하기 위한 맞춤형 학습 데이터셋을 확보할 수 있다.
- **Acceptance Criteria:**
  1. 스크립트는 (제약 없는 환경에서) '교사 모델'을 로드한다.
  2. 스크립트는 텍스트 청크를 입력하여 관련된 질문과 답변 쌍을 생성한다.
  3. 생성된 Q&A 쌍은 `/data/finetune` 폴더에 구조화된 형식으로 저장된다.
  4. 부적절한 데이터를 필터링하는 품질 검증 로직이 포함된다.

---

6. Epic 2: 핵심 모델 파인튜닝 및 MVP 구현 (상세)

Story 2.1: 교사-학생 응답 생성 파이프라인 구축

- **As a** machine learning engineer, **I want** to 합성 데이터셋을 입력받아, 교사 모델과 학생 모델 각각에 대한 응답(logits)을 생성하고 저장하는 파이프라인을 구현하여, **so that** 'Distill-M 2'의 대조적 증류 훈련에 사용할 핵심 데이터를 준비할 수 있다.
- **Acceptance Criteria:**
  1. 스크립트는 교사 모델과 학생 모델을 로드한다.
  2. 스크립트는 `/data/finetune` 폴더의 합성 데이터셋을 입력으로 받는다.
  3. 두 모델을 사용하여 각 질문에 대한 응답(logits)을 생성하고, 파일로 저장한다.
  4. `vllm` 라이브러리를 사용하여 응답 생성 과정을 효율적으로 처리한다.

---

7. Epic 3: 성능 최적화 및 최종화 (상세)

Story 3.1: `Distill-M 2 최종 훈련 구현`

- **As a** machine learning engineer, **I want** to 생성된 교사-학생 응답 쌍을 입력받아 DistiLLM 훈련 형식으로 재포맷하고, `DistiLLMTrainer`를 사용하여 최종 모델을 훈련하는 스크립트를 구현하여, **so that** 대조적 증류 방식을 통해 학생 모델의 성능을 교사 모델 수준으로 끌어올릴 수 있다.
- **Acceptance Criteria:**
  1. 스크립트는 Story 2.1에서 생성된 교사-학생 응답 쌍 데이터를 로드한다.
  2. 데이터를 DistiLLM 훈련에 맞는 형식으로 재포맷한다.
  3. `DistiLLMTrainer`와 `distillm_v2` 손실 함수를 사용하여 모델 훈련을 실행한다.
  4. 최종 훈련된 모델의 LoRA 가중치가 `/models` 디렉토리에 저장된다.
  5. Distill-M 2의 주요 하이퍼파라미터(alpha, beta, temperature) 튜닝을 위한 실험 로직이 포함된다.

Story 3.2: `LLM-as-Reranker 구현 및 통합`

- **As a** machine learning engineer, **I want** to 1차 검색 결과의 순위를 **'단일 학생 LLM'**을 활용하여 재평가하고 순위를 재조정하는 `LLM-as-Reranker` 컴포넌트를 구현하고 통합하여, **so that** 최종 LLM에 전달되는 컨텍스트의 품질을 극대화하고 '단일 LLM' 규칙을 완벽하게 준수할 수 있다.
- **Acceptance Criteria:**

1. `Multi-Stage Retriever`가 1차 검색 결과를 학생 LLM에 전달하도록 수정된다.
2. 학생 LLM이 후보 문서들의 순위를 재조정하는 프롬프트 기반 로직이 구현된다.
3. 재조정된 최종 컨텍스트가 `Inference Orchestrator`로 전달된다.

Story 3.3: End-to-End 추론 및 제출 파일 생성

- **As a** developer, **I want** to 최종 훈련 및 Reranker가 통합된 모델과 RAG 시스템을 통해 `test.csv` 전체를 처리하고 `submission.csv` 파일을 생성하여, **so that** 대회에 제출할 수 있는 완전한 결과물을 만들어낼 수 있다.
- **Acceptance Criteria:**
  1. 스크립트는 `test.csv` 파일을 입력으로 읽고 모든 행을 순회한다.
  2. 각 질문에 대해 Reranker가 포함된 RAG 검색, 프롬프트 구성, 모델 추론을 수행한다.
  3. 결과를 취합하여 규칙에 맞는 `submission.csv` 파일을 생성한다.
  4. 총 실행 시간을 측정하고 기록한다.

Story 3.4: 최종 양자화 및 성능 튜닝

- **As a** machine learning engineer, **I want** to 최종 훈련된 모델에 양자화 전략을 적용하고 미세 조정하여, **so that** 모델의 점수 하락을 최소화하면서 추론 속도와 리소스 사용량을 최적화할 수 있다.
- **Acceptance Criteria:**
  1. 다양한 양자화 수준을 모델에 적용하고 테스트한다.
  2. 각 수준별 최종 점수와 추론 시간을 측정하고 기록한다.
  3. 4.5시간 제한을 안정적으로 충족하면서 가장 높은 점수를 내는 양자화 전략을 최종 선택한다.
  4. 최종 양자화된 모델 가중치가 제출을 위해 저장된다.

Story 3.5: 최종 제출 패키지 구성 및 검증

- **As a** developer, **I want** to 대회의 모든 규칙에 따라 최종 제출 패키지를 구성하고, 재현성 검증을 완료하여, **so that** 우리의 제출물이 유효하며 심사위원들이 성공적으로 우리의 점수를 복원할 수 있도록 보장할 수 있다.
- **Acceptance Criteria:**

1. 제출에 필요한 모든 파일(코드, 모델, `requirements.txt`, 증빙 자료, 보고서)을 포함한 최종 디렉토리가 구성된다.
2. 모든 코드 내 파일 경로가 상대 경로인지 최종 확인된다.
3. 깨끗한 환경에서 재현성을 검증하는 스크립트가 포함된다.
4. 최종 결과 보고서가 작성되고 패키지에 포함된다.