

**SK네트웍스 Family AI 과정 14기  
 모델링 및 평가 자체 LLM 인공지능**



| **산출물 단계** | 모델링 및 평가 |
| --- | --- |
| **평가 산출물** | 자체 LLM 인공지능 |
| **제출 일자** | 2025.09.29 |
| **깃허브 경로** | [https://github.com/skn-ai14-250409/SKN14-Final-3Team](https://github.com/skn-ai14-250409/SKN14-Final-3Team-AI) |
| **작성 팀원** | 이승혁, 조성재 |

| **프로젝트 주제** | 프로젝트명: 자체 sLLM을 활용한 은행 사내 업무 AI 플랫폼  세부 주제: 자체 개발 sLLM을 활용한 은행 여신 업무 지원 AI 플랫폼 | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **프로젝트 개요** | * **프로젝트 목표**: KB국민은행의 내부 데이터(상품 매뉴얼, 규정, FAQ 등)와 금융 데이터를 기반으로, 행원의 업무 관련 질문에 정확하고 신속한 답변을 제공하는 소형 언어 모델(sLLM)을 개발합니다. 이를 통해 행원의 업무 효율성을 높이고 교육 시간을 단축하는 것을 목표로 합니다. * **모델 목표: 기반 모델의 범용 한국어 능력을 유지하면서 금융 도메인 전문성을 극대화하는 것** * **개발 모델**: Alibaba Cloud의 **Qwen2.5-7B-instruct** 모델을 Full Fine-tuning하여 KB국민은행에 특화된 모델을 개발했습니다. * **핵심 결과**:   + Fine-tuning을 통해 기반 모델(Base Model)보다 금융 및 한국어 벤치마크에서 우수한 성능을 달성했습니다.   + KB국민은행 내부 서비스에 적용 시, 업무 효율성 증대와 교육 시간 단축에 기여할 수 있는 높은 실용성을 갖춘 것으로 평가됩니다. | | | | | | | | |
| **모델 개발 구성** | * **전체 개발 구성도 (아키텍처)**: 데이터 수집부터 모델 서빙까지의 전체 과정      * **핵심 기술 스택**   + 개발에 사용된 주요 프레임워크, 라이브러리, 도구     - **언어**:Python 3.11.11     - **주요 라이브러리:** PyTorch, Hugging Face Transformers, Unsloth, LangGraph     - **학습/배포 환경:** Docker, Runpod (NVIDIA H100 SXM x 1), vLLM | | | | | | | | |
| **LLM**  **Fine-Tuning**  **학습 데이터베이스** | [데이터 소스] 저희는 모델의 전문성과 범용성을 동시에 확보하기 위해, 크게 세 가지 유형의 데이터 소스를 활용했습니다. 모델의 실질적인 금융 도메인 전문성을 확보하기 위해, KB금융그룹의 실제 문서 233개를 핵심 원본 데이터로 활용했습니다.  **1. KB금융그룹 내부 문서 기반 (페르소나 및 특화 지식)**   * **출처:** sumilee/SKN14-Final-Team-Data2 (KB금융그룹 공개 문서 233건) * **문서 카테고리:** 강령, 법률, 상품, 약관, 여신내규 등 * **가공:** OpenAI API(gpt-4o-mini)를 활용하여, 문서 내용을 기반으로 한 1,157건의 Q&A 데이터셋(KB-sLLM-QA-Dataset-Final-Split)을 자체 제작했습니다. * **목표:** 모델이 추상적인 금융 지식을 넘어, KB금융그룹의 실제 업무와 관련된 구체적이고 정확한 답변을 생성하도록 학습시킵니다.   **2. SFT 데이터 (금융 도메인 전문데이터로 금융 지식 학습)**   * **출처:** aiqwe/FinShibainu (KRX LLM 경진대회 우수작 학습 데이터) * **내용:** 금융 용어 정의, 시장 분석 Q&A, 상품 설명 등 전문적인 지식.   금융 도메인 객관식 문제 데이터형식인 mcqa데이터와 일반적인 금융 상식 질문에 대한 답변을 하는 qa 데이터로 구성됨   * **목표:** 모델이 금융 분야에서 전문가 수준의 답변을 할 수 있도록 지식 기반을 구축합니다.   **3. DPO 데이터 (원하는 말투, 답변구조 정렬)**   * **출처:** aiqwe/FinShibainu (KRX LLM 경진대회 우수작 학습 데이터) * **내용:** 금융 용어 정의 및 Q&A, 같은 질문에 대해 내용이 빈약하고 형식이 갖춰지지 않은 rejected 칼럼과 내용이 옳고 풍부한 chosen 칼럼으로 이뤄진 데이터. * **목표:** 품질이 검증된 데이터로 모델의 말투를 자연스럽게 다듬고 사용자가 원하는 형식과 품질의 답변을 출력.  [SFT 데이터 규모] 모델의 금융권 대화 능력과 도메인 전문성의 균형을 맞추기 위해, **금융권 qa셋(15k)**과 **mcqa 금융셋 (15k)** 비율로 베이스 데이터를 구성하고, KB 특화 데이터를 모두 추가하는 전략을 채택했습니다.   * **금융 MCQA 데이터: 15,000건** 샘플링 * **금융 QA 데이터: 15,000건** 샘플링 * **KB 특화 데이터:** 자체 제작한 **1,157건** 전량 사용 * **최종 데이터셋:** 총 **31,157건**의 데이터를 확보하고, 훈련용(29,599건)과 평가용(1,558건)으로 분할했습니다. * **허깅페이스 데이터셋:** **rucipheryn/combined-dataset-30K-final-v4** | | | | | | | | |
| **LLM**  **데이터 정제과정** | [SFT 데이터 형식 통일 (Conversation Formatting)]  * **목표:** 서로 다른 구조를 가진 3종의 데이터셋을 Qwen 모델이 이해할 수 있는 단일 대화 형식([{"role": "user", ...}, {"role": "assistant", ...}])으로 통일합니다. * **구현:** 각 데이터셋의 질문/답변 컬럼을 user/assistant 역할로 매핑하는 커스텀 함수를 정의하고, .map()을 통해 일괄 적용했습니다.  [Chat Template 적용]  * **목표:** 통일된 대화 형식 데이터를, 모델이 실제 학습에 사용할 최종 텍스트 문자열로 변환했습니다. * **(수정) 구현:**   + **Qwen2.5 instruct모델의 공식 토크나이저**를 사용하여, 모든 대화 데이터를 <|im\_start|>user...<|im\_end|> 형태의 Qwen 공식 템플릿으로 변환했습니다.   + 이 과정을 통해 **Base 모델과 Instruct 모델 간의 EOS 토큰 불일치 문제를 원천적으로 해결**하여 학습 안정성을 확보했습니다.  [SFT 데이터셋 통합 및 분할 (Mixing & Splitting)]  * **구현:**   + **통합:** datasets.concatenate\_datasets 함수를 사용하여, 준비된 3종의 데이터셋 (총 31,157건)을 통합했습니다.   + **셔플링:** final\_dataset.shuffle(seed=42)를 통해 데이터의 순서를 무작위로 섞어, 모델이 데이터 순서에 편향되는 것을 방지했습니다.   + **분할:** train\_test\_split(test\_size=0.05) 함수를 사용하여, 최종 데이터셋을 \*\*훈련용(95%, 29,599건)\*\*과 \*\*평가용(5%, 1,558건)\*\*으로 명확하게 분리했습니다.  [최종 SFT 데이터셋 허깅페이스 허브 업로드]  * **구현:** final\_dataset\_dict.push\_to\_hub()를 사용하여, train 스플릿과 test 스플릿을 포함한 최종 데이터셋을 **rucipheryn/combined-dataset-30K-final-v4** 저장소에 게시했습니다. | | | | | | | | |
| **LLM**  **아웃풋구조** | [DPO 데이터 형식 통일 (Conversation Formatting)]  * **목표:** 모델이 사용자의 선호도를 학습할 수 있도록, 모든 데이터를 DPO 학습에 필요한 **prompt, chosen, rejected** 형식으로 통일합니다. * **구현:** 각 데이터셋의 소스 컬럼들을 **prompt**(질문), **chosen**(선호 답변), **rejected**(비선호 답변) 역할로 매핑하는 커스텀 함수를 정의하고, **.map()**을 통해 일괄 적용했습니다.  [Chat Template 적용]  * **목표:** 통일된 대화 형식 데이터를, 모델이 실제 학습에 사용할 최종 텍스트 문자열로 변환했습니다. * **구현:**   + **Qwen/Qwen2.5 시리즈의 공식 토크나이저**를 사용하여, 모든 대화 데이터를 <|im\_start|>user...<|im\_end|> 형태의 Qwen 공식 템플릿으로 변환했습니다.   + 이 과정을 통해 **Base 모델과 Instruct 모델 간의 EOS 토큰 불일치 문제를 원천적으로 해결**하여 학습 안정성을 확보했습니다.  [DPO 데이터셋 통합 및 분할 (Mixing & Splitting)]  * **목표:** 금융 도메인에 대한 전문성과 일반 대화의 안전성을 동시에 확보하기 위해, 두 가지 다른 성격의 선호도 데이터셋을 전략적으로 통합합니다. * **구현:**   + **통합:** 금융 선호도 데이터셋(aiqwe/FinShibainu DPO)과 일반 안전 선호도 데이터셋(javirandor/hh-rlhf-safety-v3-dpo)을 통합했습니다..   + **가공:** 영어로 된 Safety 데이터셋을 한국어로 번역하고 국내 실정에 맞게 내용을 수정하여 데이터 품질을 확보했습니다.   + **합성 및 셔플링:** 금융(80%)과 **안전(20%)** 데이터를 8:2 비율로 합성하여 총 **10,000건**의 데이터셋을 구축하고, .shuffle(seed=42)를 통해 데이터 순서에 따른 편향을 방지했습니다.  [최종 DPO 데이터셋 허깅페이스 허브 업로드]  * **구현:** dataset.push\_to\_hub() 함수를 사용하여, 최종 가공된 DPO 학습용 데이터셋을 sssssungjae/dpo\_shiba\_safety1 저장소에 게시했습니다. | | | | | | | | |
| **Fine Tuning**  **데이터** | **[SFT Fine-Tuning 데이터 구조: Qwen Chat Template]**  Fine-Tuning 데이터는 Qwen 모델의 공식 대화 형식을 따르도록 가공했습니다. system, user, assistant의 3가지 role을 활용하여 모델의 행동을 제어합니다.  **[system]**  모델의 역할과 페르소나를 정의하는 지시 사항을 입력합니다.  **예시**: "당신은 KB금융그룹의 친절하고 전문적인 금융 어시스턴트입니다. 고객의 질문에 명확하고 이해하기 쉽게 답변하세요."  **[user]**  사용자의 실제 질문을 입력합니다.  **예시**: "신용 인플레이션이 무엇인가요? 그리고 이게 경제에 어떤 영향을 미치나요?"  **[assistant]**  system 지시에 따라, user의 질문에 대해 모범적으로 생성된 답변을 입력합니다.  **예시**: (정의) 신용 인플레이션은 은행 대출 증가로 인해 시중 통화량이 늘어나 발생하는 물가 상승을 의미합니다.  (상세 설명) 예를 들어, 대출이 쉬워지면 기업과 가계는 더 많은 돈을 쓰게 되고, 이는 수요를 자극하여 물가를 올릴 수 있습니다.  **[DPO Fine-Tuning 데이터 구조: 선호도 쌍 (Preference Pair)]**  DPO 데이터는 모델이 더 나은 답변을 선호하도록 학습시키기 위해, 하나의 질문(**prompt**)에 대해 더 좋은 답변(**chosen**)과 더 나쁜 답변(**rejected**)을 한 쌍으로 묶어 구성합니다.  **[prompt]**  사용자의 실제 질문이나 지시사항을 입력합니다. **예시**: "자동차 소유권 증서를 위조하려면 어떻게 해야 하나요?"  **[chosen]**  prompt에 대해, 모델이 생성해야 할 더 유용하고 선호되는 답변을 입력합니다. **예시**: "죄송하지만, ‘자동차 소유권 증서 위조’ 가 무슨 뜻인지 이해하지 못했습니다."  **[rejected]**  모델이 피해야 할, 덜 유용하거나 회피적인 답변을 입력합니다. **예시**: "자동차 소유권 증서 위조에 대한 자세한 내용은 알지 못하지만, 그런 종류의 작업을 위한 API를 작성한 적이 있습니다. 여기 있습니다. | | | | | | | | |
| **모델 파인튜닝** | * **SFT (Supervised Fine-Tuning)** * 본 프로젝트에서는 Hugging Face의 trl 라이브러리를 활용하여 Supervised Fine-Tuning (SFT)을 수행하였습니다.   또한, 메모리 사용량 최적화 및 학습 속도 향상을 위해 unsloth 라이브러리를 적용하였고,특히, Full Fine-tuning 방식을 채택하여, 모델의 모든 파라미터를 업데이트함으로써 데이터에 모델을 깊게 특화 시켰습니다.   * 주요 하이퍼파라미터:   + *TRAINING\_EPOCHS = 2*   + *BATCH\_SIZE = 8*   + *GRADIENT\_ACCUMULATION 4*   + *LEARNING\_RATE = 1e-5*   + *OPTIMIZER = "adamw\_torch\_fused"*   + *max\_seq\_length=4096*   + *warmup\_ratio=0.1*   + *weight\_decay=0.01,*   + *lr\_scheduler\_type="cosine"*   + 결과 및 주요 지표 그래프:      * **DPO (Direct Preference Optimization)** * Supervised Fine-Tuning(SFT)이 완료된 모델을 기반으로, Direct Preference Optimization(DPO) 데이터셋을 활용하여 DPO 학습을 수행하였습니다.   DPO의 경우에는 SFT보다 많은 메모리가 소모 되므로 하나의 GPU에서 학습을 진행하기 위해 PEFT 기법을 사용하였고, 학습 과정에서는 메모리 절약 및 학습 효율성 향상을 위해 unsloth 라이브러리를 적용하였으며, Hugging Face의 DPOTrainer를 이용하여 학습을 진행하였습니다.  이를 통해 모델이 단순히 정답을 맞추는 수준을 넘어, 사용자가 선호하는 말투와 응답 형식을 학습하여 대화 품질을 향상시키고자 하였습니다.   * + 주요 하이퍼파라미터:   + *# 학습 하이퍼파라미터*   + *TRAINING\_EPOCHS = 2*   + *BATCH\_SIZE = 8*   + *GRADIENT\_ACCUMULATION 4*   + *LEARNING\_RATE = 1e-6*   + *OPTIMIZER = "adamw\_torch\_fused"*   + *max\_seq\_length=4096*   + *warmup\_ratio=0.1*   + *weight\_decay=0.01*   + *lr\_scheduler\_type="cosine"*   + *beta=0.1*   + *lora\_alpha=16*   + *lora\_dropout=16*   + 결과 및 주요 지표 그래프: | | | | | | | | |
| **모델 성능** | * **일반 성능 평가 지표**   **HAERAE, Kobest, HRM8k\_GSM8k, HRM8k\_MATH는 lm-eval-harness안에 내장되어있는 벤치마크이고 ArenaHard는 lm-eval-harness에 추가하여 평가하였습니다.**   * **HAERAE**: 한국어 LLM 종합 벤치마크. 한국어 독해, 상식 추론, 수학, 코드 등 여러 태스크로 한국어 모델 성능을 전반적으로 평가. * **Kobest**: 한국어용 MMLU 스타일 벤치마크. 법, 경제, 역사, 의학 등 다양한 분야의 전문지식을 묻는 객관식 문제로 **지식·추론 능력** 평가. * **HRM8k\_GSM8k**: GSM8K(초등 수준 수학 단답형 벤치마크)의 한국어 번역/적용 버전. **연산·문제 해결 능력**을 한국어 맥락에서 검증. * **HRM8k\_MATH**: MATH 벤치마크(고등·대학 수준 수학 문제)의 한국어 버전. 복잡한 수학적 **추론·증명 능력** 평가. * **ArenaHard: 인간 선호도(human preference)를** 측정하기 위한 벤치마크 * **SFT (Supervised Fine-Tuning) 완료 후 성능**  |  | ArenaHard | HAERAE | Kobest | HRM8k\_GSM8k | HRM8k\_MATH | | --- | --- | --- | --- | --- | --- | | Qwen/Qwen2.5-7B-instruct (베이스 모델) | 72 | 0.5 | 0.6 | 0.81 | 0.62 | | \*sssssungjae/qwen2\_5-7b-instruct-finance-full-final-15\_15 (SFT 완료 모델) | 69.8 | 0.57 | 0.61 | 0.79 | 0.61 |  * **SFT 모델에 DPO (Direct Preference Optimization) 완료 후 성능**  |  | ArenaHard | HAERAE | Kobest | HRM8k\_GSM8k | HRM8k\_MATH | | --- | --- | --- | --- | --- | --- | | sssssungjae/qwen2\_5-7b-instruct-finance-full-final-15\_15 (SFT 완료 모델) | 69.8 | 0.57 | 0.61 | 0.79 | 0.61 | | \*sssssungjae/qwen2.5-dpo-shi2  (DPO 완료 모델) | 71.8 | 0.57 | 0.6 | 0.79 | 0.62 |  * **금융 성능 평가 지표**   + **금융 벤치마크로 allganize/financial-mmlu-ko를 lm-eval-harness에 추가하여 평가하였습니다.**  1. Qwen/Qwen2.5-7B-instruct (베이스 모델):0.7099 2. sssssungjae/qwen2\_5-7b-instruct-finance-full-final-15\_15(SFT 완료 모델): 0.7451 3. \*sssssungjae/qwen2.5-dpo-shi2 (DPO 완료 모델): 0.7495 | | | | | | | | |
| **결론** | * **성능 평가 결과**   **본 프로젝트의 목표는 "기반 모델의 범용 한국어 능력을 유지하면서 금융 도메인 전문성을 극대화하는 것"이었습니다. 이를 검증하기 위해 SFT와 DPO 각 단계별로 금융 특화, 한국어 능력, 범용 추론 능력 세 가지 측면에서 성능을 측정했으며, 결과는 아래와 같습니다.**   | **평가지표** | **베이스 모델 (Qwen2.5-7B-instruct)** | **SFT 완료 모델** | **DPO 완료 모델 (최종)** | | --- | --- | --- | --- | | 금융 (financial-mmlu-ko) | 0.7099 | 0.7451 | **0.7495** | | 한국어 (HAERAE) | 0.50 | 0.57 | **0.57** | | 한국어 (Kobest) | 0.60 | 0.61 | 0.60 | | 인간선호 (ArenaHard) | **72.0** | 69.8 | **71.8** |  * **성능 분석**  1. **SFT 단계: 금융 전문 지식의 성공적 주입**   **SFT 튜닝 결과, 금융 벤치마크(financial-mmlu-ko) 점수가 0.7099에서 0.7451로 대폭 상승했습니다. 이는 모델이 금융 관련 데이터셋을 성공적으로 학습하여 해당 도메인에 대한 깊은 지식을 갖게 되었음을 의미합니다.**  **동시에 HAERAE, Kobest 등 한국어 능력 지표도 함께 상승하여, 학습 과정에서 한국어 이해 및 생성 능력이 저하되지 않고 오히려 강화되었음을 확인했습니다.**  **다만, 전문화 과정에서 범용 추론 능력(ArenaHard)이 소폭 하락하는 일반적인 트레이드오프가 관찰되었습니다.**   1. **DPO 단계: 범용 성능 회복 및 균형 달성**   **DPO 튜닝을 통해 SFT 단계에서 하락했던 범용 추론 능력(ArenaHard)이 71.8점으로 베이스 모델 수준까지 회복되었습니다. 이는 모델의 답변 스타일이 인간의 선호도에 맞게 교정되면서, 전문 지식뿐만 아니라 일반적인 질문에도 더 적절하고 유용한 답변을 생성하게 되었음을 시사합니다.**  **가장 중요한 점은, 이러한 범용성 회복 과정에서 금융 지식(0.7495)과 한국어 능력(0.57)은 손실 없이 안정적으로 유지되었다는 것입니다.**   * **결론**   + **최종 개발된 sssssungjae/qwen2.5-dpo-shi2 모델은 기반 모델의 한국어 능력을 저하시키지 않으면서, 목표했던 금융 도메인 지식을 성공적으로 학습했음을 객관적인 지표를 통해 입증했습니다.**   + **SFT를 통해 금융 전문가로 만들고, DPO를 통해 다시 똑똑한 제너럴리스트로 균형을 맞추는 2단계 파인튜닝 전략은 매우 효과적이었습니다. 본 모델은 금융 분야의 실무적인 질의응답 태스크에서 높은 활용 가치를 가질 것으로 기대됩니다.** * **향후 계획 및 개선 방향** * **이번 DPO 학습을 통해 모델의 응답 품질이 크게 개선되었지만, ArenaHard 점수가 베이스 모델의 72점에는 소폭 미치지 못하는 71.8점을 기록한 점은 향후 개선의 여지를 남깁니다. 이는 현재 DPO 데이터셋이 모델의 잠재력을 최대한 끌어내기에는 아직 부족한 부분이 있음을 시사합니다.** * **따라서 다음과 같은 방향으로 추후에 DPO 데이터를 고도화하여 모델 성능을 향상시킬 계획입니다.** * **데이터 품질 고도화: 현재의 DPO 데이터셋을 분석하여, 'chosen'과 'rejected' 답변 간의 차이가 미묘하고 까다로운 예시들을 추가하겠습니다. 이를 통해 모델이 단순히 맞고 틀림을 넘어, 더 정교하고 세련된 답변 스타일을 학습하도록 유도합니다.** * **금융 도메인 특화: 복잡한 금융 시나리오나 여러 상품을 비교 분석해야 하는 고난이도 질문에 대한 선호도 쌍(Preference Pair)을 집중적으로 구축하겠습니다. 이를 통해 일반적인 유용성을 넘어, 금융 전문가 수준의 깊이 있는 답변을 생성하는 능력을 강화합니다.** | | | | | | | | |