ChatGPT

한국어 금융 QA 모델을 위한 공개 데이터셋 조사

금융 AI Challenge 규칙에 따르면 **2025년 8월 1일 이전에 공개되었고 비상업적 이용이 가능한 라이선스(CC-BY-NC, CC-BY-SA, CC-BY-NC-SA, CC0 등)를 가진 공개 데이터**만 사용할 수 있으며, 자체 크롤링이나 수기 작성 데이터는 금지된다【2958†L0-L7】. 또한 외부 데이터는 출처를 명확히 밝히고 증빙 자료를 제출해야 한다【2958†L0-L7】. 따라서 아래에서는 규정에 맞는 한국어 금융 QA 데이터셋을 제안한다. 표에는 긴 문장을 피하기 위해 핵심 정보만 정리했다.

데이터셋 이름	특징/내용	라이선스 및 장점
₩ON-Instruct (KRX-Data)	금융 규제기관·거래소·교육자료 등에서 수집한 20만개 이상의 한국어 금융 문서를 GPT-4o 및 Qwen2.5-72B 등을 통해 **다지선다형(MCQA)과 Instruction-Response 형식**으로 재작성한 후 품질 필터링을 거쳐 8만여건으로 정제한 인스트럭션 데이터셋 [1]. 평균 80k개의 질문-응답쌍으로 구성되며 한국어 금융 추론과 도메인-특화 작업에 적합함 [2].	MIT 라이선스로 공개되어 사용과 재배포가 자유롭다 [3]. 한국 금융 정보원, 한국거래소, 금융위원회 등 신뢰성 있는 자료를 기반으로 수집했기 때문에 도메인 지식이 풍부하며, 규정상 허용되는 공공 데이터임.
FinShibainu	한국어 금융, 회계, 증권 관련 자료에서 생성한 **다지선다형 문제 4.2만건과 QA 데이터 4.4만건**으로 구성된 데이터셋 [4]. 문제와 보기뿐 아니라 **추론 과정(reasoning process)**이 포함되어 있어 모델의 설명 가능성 향상에 도움이 된다 [5]. 데이터 출처는 한국은행 경제·금융용어 700선, 금융감독용어사전, 한국거래소 규정, 기업공시자료 등 여러 공공 자료로 구성되어 있다 [6].	Apache-2.0 라이선스이며 상업적 이용도 가능하다 [7]. 각 문제에 대한 정답 번호와 해설이 포함되어 있으므로 금융 전문 QA와 선택지 문제 학습에 적합하다.
₩ON-Reasoning / KRX Financial LLM Competition 데이터	한국거래소 주최 LLM 경진대회에서 사용된 **약 5.5k개의 다지선다형 질문 및 Instruction-Response** 데이터로 구성되어 있으며 분야는 **재무·회계, 금융시장, 국내 기업 분석, 금융 에이전트, 주가 예측, 열린형 FinQA** 등 여섯 가지로 나뉜다 [8]. 대회 리더보드와 함께 공개된 평가용 벤치마크이다.	공개 라이선스(MIT)로 제공되며 적은 양의 데이터로 빠르게 튜닝해 베이스라인 모델의 초기 성능을 높이는 데 사용할 수 있다. 여러 세부 카테고리를 포함하고 있어 모델이 다양한 금융 QA 유형을 학습하는 데 도움이 된다 [8].
일반 한국어 QA/읽기 데이터 (KorQuAD 1.0/2.0 등)	위 데이터들이 부족할 경우, 금융과 직접 관련되진 않지만 **위키피디아 한국어 문서에서 생성한 기계독해용 질문과 답변**인 KorQuAD 데이터셋 등을 보완적으로 활용할 수 있다. 다만 KorQuAD 2.0은 CC BY-ND 2.0 KR 라이선스로 상업적 2차 저작이 제한되므로 활용 시 라이선스 준수 여부를 확인해야 한다.	일반적인 한국어 독해 능력을 강화해 모델의 언어 이해력을 높이는 데 도움을 주지만, 금융 특화 성능을 위해서는 위의 금융 도메인 데이터와 함께 사용해야 한다.

데이터셋 활용 전략

1. **₩ON-Instruct + FinShibainu 혼합 학습** – 두 데이터셋 모두 한국어 금융 도메인 질문·응답을 포함하고, MIT 또는 Apache 라이선스로 자유롭게 활용할 수 있다. FSKU와 유사한 **객관식 문제**와 **주관식 해설**이 포함되어 있어 모델이 다양한 질문 유형에 대응하도록 도와준다.
2. **데이터 증강을 신중히 진행** – 대회 규정에 따라 원본 데이터와 라이선스가 명확한 자료만 사용해야 하며, 증강 과정과 사용한 모델을 모두 제출해야 한다【2958 † L0-L7】. 예를 들어 번역 모델을 사용하여 영문 금융 QA 데이터를 한국어로 변환하거나, LoRA를 적용한 로컬 LLM으로 추가 질문을 생성하는 방법 등을 고려할 수 있다.
3. **일반 한국어 QA 데이터로 기초 언어 능력 강화** – KorQuAD 등에서 비금융 질문에 대한 이해력을 학습시키고, 이후 금융 데이터셋으로 미세튜닝하는 **커리큘럼 학습**을 적용하면 금융 도메인 적응력이 향상될 수 있다.

파인튜닝 방법론 개요 및 예제

금융 AI Challenge는 단일 LLM으로 객관식과 주관식 질문에 모두 답해야 하므로 모델이 안정적으로 긴 출력을 생성하는 것이 중요하다. 여기서는 베이스라인 모델인 **gemma-ko-7b (4bit)**에 적합한 미세튜닝 방법 세 가지를 비교한다.

1. Low-Rank Adaptation (LoRA)

- **개념** – LoRA는 대형 언어모델의 가중치를 모두 업데이트하지 않고, 각 선형층에 **저랭크 행렬(두 개의 작은 행렬)을 추가하여** 학습하는 기법이다. 원래 가중치는 고정하고 LoRA 행렬만 학습하기 때문에 저장 공간과 GPU 메모리를 크게 절약할 수 있다 [9]. Microsoft에서 2021년에 발표되었으며 저차원 재매개화(low-dimension reparameterization)를 통해 모델을 특정 도메인에 적응시킨다 [10].
- **장점** – GPT-3와 같은 초대형 모델에서도 체크포인트 크기를 1.2 TB에서 35 MB로 줄일 수 있을 정도로 **저장 공간을 대폭 줄이고** [11], 전체 가중치를 업데이트하는 전체 미세튜닝(full fine-tuning)에 비해 **GPU 메모리 요구량이 최대 3배 감소**한다 [12]. 또한 LoRA 가중치는 사전학습된 가중치와 단순 합으로 병합할 수 있어 **추론 지연을 늘리지 않는다** [13].
- **예제 코드** (Hugging Face `peft` 라이브러리 사용):

```
from transformers import AutoTokenizer, AutoModelForCausalLM, Trainer, TrainingArguments
from peft import LoraConfig, get_peft_model
from datasets import load_dataset

# 1) 4bit로 양자화된 gemma-ko-7b 모델 로드
model_name = "beomi/gemma-ko-7b"
model = AutoModelForCausalLM.from_pretrained(
    model_name,
    load_in_4bit=True,   # bitsandbytes 4bit 양자화
    device_map="auto"
)
tokenizer = AutoTokenizer.from_pretrained(model_name)

# 2) LoRA 구성 정의 (랭크 r, scaling factor alpha 등)
lora_config = LoraConfig(
    r=16,
    lora_alpha=32,
    target_modules=["q_proj", "v_proj"],   # attention의 Query/Value 부분만 적용
    lora_dropout=0.05,
    bias="none"
)
```

```
# 3) LoRA 어댑터를 모델에 적용
model = get_peft_model(model, lora_config)

# 4) 학습 데이터셋 로드 및 전처리 (예: ₩ON-Instruct 데이터)
dataset = load_dataset("KRX-Data/Won-Instruct", split="train[:20000]")

def preprocess(batch):
    inputs = batch["prompt"]   # 질문
    targets = batch["original_response"]   # 정답/해설
    texts = [f"[INST] {q}\n\n[/INST] {a}" for q, a in zip(inputs, targets)]
    tokenized = tokenizer(texts, truncation=True, padding="max_length", max_length=1024)
    tokenized["labels"] = tokenized["input_ids"].copy()
    return tokenized

train_dataset = dataset.map(preprocess, batched=True,
remove_columns=dataset.column_names)

# 5) Trainer 설정 및 학습
args = TrainingArguments(
    output_dir="./finetuned_gemma_lora",
    per_device_train_batch_size=2,
    gradient_accumulation_steps=4,
    learning_rate=2e-4,
    num_train_epochs=1,
    fp16=True,
    logging_steps=50,
    save_steps=500
)
trainer = Trainer(model=model, args=args, train_dataset=train_dataset)
trainer.train()

# 6) 추론 시 LoRA 어댑터 병합(옵션)
model.merge_and_unload()
model.save_pretrained("./merged_gemma_lora")
```

위 코드는 4-bit 양자화된 gemma-ko-7b를 로드하고 LoRA를 적용한 뒤, 한국어 금융 인스트럭션 데이터로 미세튜닝하는 예시이다. `target_modules`를 적절히 지정해 Query/Value 가중치에만 저랭크 행렬을 추가하여 효율성을 높인다. 학습 후 `merge_and_unload()`를 호출하면 LoRA 어댑터를 원래 가중치에 병합해 추론 시 별도의 어댑터 없이도 사용할 수 있다.

2. Quantized LoRA (QLoRA)

- **개념** – QLoRA는 LoRA를 적용하면서 사전학습 모델을 4비트 **NormalFloat(NF4) 형식으로 양자화**해 GPU 메모리를 더 절약하는 기법이다. 연구 논문에서는 65억 파라미터 모델을 단일 48 GB GPU로 미세튜닝하면서도 16-비트 전체 미세튜닝과 동일한 성능을 달성했다고 보고한다 [14]. QLoRA는 **4비트 양자화된 모델에 대해 그래디언트를 역전파하여 LoRA 어댑터를 학습**하며, “4-bit NormalFloat” 데이터 타입과 **이중 양자화(double quantization)**, **페이지드 최적화(paged optimizers)** 기술을 통해 메모리 사용을 줄인다 [15].
- **장점** – 일반 LoRA보다 메모리를 더 절약해 24GB VRAM에서도 70억 파라미터 모델을 미세튜닝할 수 있으며, 성능 저하가 거의 없다 [14]. 단, 4비트 양자화로 인해 연산 속도가 조금 느려질 수 있다.

- **예제 코드** (bitsandbytes + `peft` 사용):

```
from transformers import AutoTokenizer, AutoModelForCausalLM, Trainer, TrainingArguments
from peft import LoraConfig, get_peft_model
from bitsandbytes.nn import Linear4bit  # 4비트 양자화 레이어

model_name = "beomi/gemma-ko-7b"
model = AutoModelForCausalLM.from_pretrained(
    model_name,
    load_in_4bit=True,        # 모델을 4비트로 로드
    quantization_config={"bnb_4bit_compute_dtype": "bfloat16"}
)

# LoRA 구성은 위와 동일
lora_config = LoraConfig(r=16, lora_alpha=32, target_modules=["q_proj", "v_proj"],
lora_dropout=0.05)
model = get_peft_model(model, lora_config)

# 이후 단계는 LoRA와 동일하게 Trainer로 학습
```

QLoRA의 학습 과정은 LoRA와 거의 동일하나, 모델을 4비트로 로드하는 부분과 옵티마이저 선택(예: `paged_adamw_32bit` 등)이 중요하다. 라이브러리에서 제공하는 `quantization_config`를 활용하면 손쉽게 적용할 수 있다.

3. 전체 미세튜닝(Full fine-tuning)

- **개념** – 전체 미세튜닝은 모델의 모든 가중치를 업데이트하는 전통적인 방식이다. 파라미터 수가 수십억 개에 달하는 LLM에서는 **메모리와 저장 공간을 많이 소모**하고 학습 시간이 길다. 예를 들어 70억 파라미터 모델이라도 16-비트로 로드하면 약 28 GB 이상의 VRAM이 필요하며, 그래디언트 저장까지 포함하면 2~3배 이상의 메모리가 요구된다.
- **장점/단점** – 데이터가 충분하고 GPU 자원이 넉넉하다면 가장 높은 성능을 얻을 수 있지만, 베이스라인(4bit gemma-ko-7b)과 같은 환경에서는 실행이 어려울 수 있다. 또한 기존 모델의 일반 지식을 희석시키거나 과적합 위험이 있다. 따라서 대회 환경(RTX 4090 24GB VRAM)에서는 LoRA나 QLoRA 같은 **파라미터 효율적 미세튜닝**을 추천한다.

추가 고려 사항

- **프롬프트 설계** – 객관식과 주관식 응답을 모두 생성해야 하므로 프롬프트에 질문 유형을 명시하고 출력 포맷을 제시하는 것이 중요하다. 예: 객관식의 경우 “질문 … 1 … 2 … 정답: ” 형식, 주관식은 “질문 … 답변: ” 형식으로 구성하여 모델이 혼동하지 않도록 한다.
- **데이터 커리큘럼** – 일반 한국어 QA 데이터 → 금융 일반지식 데이터 → 금융 전문 QA 데이터 순으로 점진적으로 학습시켜 모델이 기초 언어 능력과 금융 도메인 지식을 모두 갖추도록 한다. 대회 보고서에서 언급된 **커리큘럼 기반 SFT**와 **DPO**(Direct Preference Optimization) 기법도 고려할 수 있다 [16].
- **라이선스와 증빙** – 외부 데이터와 미세튜닝 모델은 모두 라이선스와 출처를 명시해야 하며, 사용 과정(전처리, 증강, 학습 코드)을 코드와 함께 제출해야 한다【2958†L0-L7】.

결론

한국어 금융 QA 모델의 성능을 높이기 위해서는 **공개된 도메인 특화 데이터셋**을 활용한 LoRA 또는 QLoRA 기반 미세튜닝이 적합하다. ₩ON-Instruct와 FinShibainu는 한국어 금융 지식을 포함하고 합법적으로 사용할 수 있는 대표적인

공개 데이터셋이며, 베이스라인 모델(gemma-ko-7b 4bit)을 효율적으로 업데이트하는 데 도움이 된다. LoRA는 메모리와 저장 공간을 크게 절약하면서도 높은 성능을 유지하며 [17], QLoRA는 4비트 양자화로 추가적인 메모리 절감 효과를 제공한다 [15]. 이러한 전략을 기반으로 데이터 준비, 프롬프트 설계, 커리큘럼 학습을 체계적으로 수행하면 금융 AI Challenge에서 높은 성능을 달성할 수 있을 것이다.

---

[1] [2] [3] KRX-Data/Won-Instruct · Datasets at Hugging Face
https://huggingface.co/datasets/KRX-Data/Won-Instruct

[4] [5] [7] aiqwe/FinShibainu · Datasets at Hugging Face
https://huggingface.co/datasets/aiqwe/FinShibainu

[6] aiqwe/FinShibainu · Hugging Face
https://huggingface.co/aiqwe/FinShibainu

[8] [16] KRX-Data/WON-Reasoning · Hugging Face
https://huggingface.co/KRX-Data/WON-Reasoning

[9] [10] [11] [12] [13] [17] Low-Rank Adaptation (LoRA): Revolutionizing AI Fine-Tuning
https://coralogix.com/ai-blog/low-rank-adaptation-a-closer-look-at-lora/

[14] [15] [2305.14314] QLoRA: Efficient Finetuning of Quantized LLMs
https://arxiv.org/abs/2305.14314