



250219_EXAONE H100에 설치하기 (vLLM, FastAPI)

참고: <https://www.youtube.com/watch?v=j5Mcmiflbnc&list=PLJkjrxxiBSFbYC3redRVJUc051ohXRQq0&index=2>

0단계: 가상 환경 생성

```
# 가상환경 생성:
virtualenv -p python3.10 test_env

# 가상환경 활성화:
source test_env/bin/activate
```

1단계: LLM 모델 다운로드

1-1) 허깅페이스 접근 토큰 발급 받기

허깅페이스에서 모델 다운로드를 받기 위해서는 username과 접근 토큰을 입력해야 한다.

방법 → 프로필 → settings → Access Tokens → Write token 생성

```
hf_허깅페이스 토큰
```

1-2) 모델 다운로드

• git-lfs 설치 (필수, 기존의 git이 아닌 큰 용량을 위한 git)

```
sudo apt update
sudo apt install git-lfs
```

기존에 git이 설치되어 있다고 하더라도 대용량 리포지토리 다운로드를 위한 git-lfs가 아니기 때문에 git-lfs를 설치해야 한다.

•

git으로 모델 다운로드

```
mkdir slm_model
cd slm_model

git clone https://huggingface.co/LGAI-EXAONE/EXAONE-3.5-7.8B-Instruct
```

Hugging face 아이디 입력과 접속 토큰 (비밀번호 아님) 입력 항목이 발생합니다.

디렉토리 내부에 접근 후 .safetensors와 같은 파일이 존재한다면 다운로드가 제대로 되었습니다.

```
(test_env) ubuntu@H100:~/gskim/slm_model/EXAONE-3.5-7.8B-Instruct$ ll
total 30549052
drwxrwxr-x 4 ubuntu ubuntu      4096 2월 19 11:47 ./
drwxrwxr-x 3 ubuntu ubuntu      4096 2월 19 11:39 ../
drwxrwxr-x 2 ubuntu ubuntu      4096 2월 19 11:39 assets/
-rw-rw-r-- 1 ubuntu ubuntu     1048 2월 19 11:39 config.json
-rw-rw-r-- 1 ubuntu ubuntu     9954 2월 19 11:39 configuration_exaone.py
-rw-rw-r-- 1 ubuntu ubuntu      134 2월 19 11:39 generation_config.json
drwxrwxr-x 9 ubuntu ubuntu      4096 2월 19 11:47 .git/
-rw-rw-r-- 1 ubuntu ubuntu     1519 2월 19 11:39 .gitattributes
-rw-rw-r-- 1 ubuntu ubuntu     13779 2월 19 11:39 LICENSE
-rw-rw-r-- 1 ubuntu ubuntu     1219196 2월 19 11:39 merges.txt
-rw-rw-r-- 1 ubuntu ubuntu    4932636680 2월 19 11:46 model-00001-of-00007.safetensors
-rw-rw-r-- 1 ubuntu ubuntu    4999813040 2월 19 11:47 model-00002-of-00007.safetensors
-rw-rw-r-- 1 ubuntu ubuntu    4999813080 2월 19 11:46 model-00003-of-00007.safetensors
-rw-rw-r-- 1 ubuntu ubuntu    4832007464 2월 19 11:45 model-00004-of-00007.safetensors
-rw-rw-r-- 1 ubuntu ubuntu    4999813088 2월 19 11:46 model-00005-of-00007.safetensors
-rw-rw-r-- 1 ubuntu ubuntu    4832023944 2월 19 11:45 model-00006-of-00007.safetensors
-rw-rw-r-- 1 ubuntu ubuntu    1677721728 2월 19 11:41 model-00007-of-00007.safetensors
-rw-rw-r-- 1 ubuntu ubuntu      63625 2월 19 11:39 modeling_exaone.py
-rw-rw-r-- 1 ubuntu ubuntu      23729 2월 19 11:39 model.safetensors.index.json
-rw-rw-r-- 1 ubuntu ubuntu      7446 2월 19 11:39 README.md
-rw-rw-r-- 1 ubuntu ubuntu       563 2월 19 11:39 special_tokens_map.json
-rw-rw-r-- 1 ubuntu ubuntu      70718 2월 19 11:39 tokenizer_config.json
-rw-rw-r-- 1 ubuntu ubuntu     4957806 2월 19 11:39 tokenizer.json
-rw-rw-r-- 1 ubuntu ubuntu     1934226 2월 19 11:39 vocab.json

(test_env) ubuntu@H100:~/gskim/slm_model$ git lfs --version
git-lfs/2.9.2 (GitHub; linux amd64; go 1.13.5)
(test_env) ubuntu@H100:~/gskim/slm_model$ cd EXAONE-3.5-7.8B-Instruct/
(test_env) ubuntu@H100:~/gskim/slm_model/EXAONE-3.5-7.8B-Instruct$ git lfs pull
(test_env) ubuntu@H100:~/gskim/slm_model/EXAONE-3.5-7.8B-Instruct$ ls -lh | grep .safetensors
-rw-rw-r-- 1 ubuntu ubuntu 4.6G 2월 19 11:46 model-00001-of-00007.safetensors
-rw-rw-r-- 1 ubuntu ubuntu 4.7G 2월 19 11:47 model-00002-of-00007.safetensors
-rw-rw-r-- 1 ubuntu ubuntu 4.7G 2월 19 11:46 model-00003-of-00007.safetensors
-rw-rw-r-- 1 ubuntu ubuntu 4.6G 2월 19 11:45 model-00004-of-00007.safetensors
-rw-rw-r-- 1 ubuntu ubuntu 4.7G 2월 19 11:46 model-00005-of-00007.safetensors
-rw-rw-r-- 1 ubuntu ubuntu 4.6G 2월 19 11:45 model-00006-of-00007.safetensors
-rw-rw-r-- 1 ubuntu ubuntu 1.6G 2월 19 11:41 model-00007-of-00007.safetensors
-rw-rw-r-- 1 ubuntu ubuntu 24K 2월 19 11:39 model.safetensors.index.json
(test_env) ubuntu@H100:~/gskim/slm_model/EXAONE-3.5-7.8B-Instruct$
```

2단계: 사용 가능한 GPU 확인 (**nvidia-smi**)

2-1) GPU 환경에서 실행 가능한지 확인하기

- **vllm** 은 GPU 가속을 사용하는 패키지이므로, GPU 환경에서 실행 가능한지 확인하는 것도 중요합니다.
- 터미널에서 다음 명령어 실행:

```
nvidia-smi
```

- CUDA가 활성화되어 있어야 **vllm** 이 제대로 동작합니다

2-2) GPU가 활성화 되지 않은 경우, NVIDIA 드라이버 설치 확인

```
dpkg -i | grep -i nvidia
```

2-3) NVIDIA 드라이버 설치

```
# 시스템 업데이트 및 커널 헤더 확인
sudo apt update && sudo apt upgrade -y
sudo apt install linux-headers-$(uname -r)
```

```
# 최신 NVIDIA 드라이버 재설치 (535 버전)
sudo apt install -y nvidia-driver-535
```

```
# 설치가 끝난 후 시스템을 재부팅
sudo reboot
```

```
# GPU 사용 가능 여부 확인
nvidia-smi
```

3단계: vLLM 설치

-

pip3 설치

```
sudo apt update  
  
sudo apt install python3-pip
```

• 3-1) vLLM 설치

```
vllm 설치 명령어:  
pip3 install vllm  
  
vllm 설치 확인 명령어  
pip list | grep vllm
```

```
(test_env) ubuntu@H100:~/gskim$ pip list | grep vllm  
vllm 0.7.2
```

⇒ 현재 vllm 0.72 버전으로 설치 완료

• 3-2) 설치 및 확인 테스트



로컬에 모델 다운로드 후, vLLM으로 서빙

```
huggingface-cli download LGAI-EXAONE/EXAONE-3.5-7.8B-Instruct --local-dir "/home/ubuntu/models/exaone"
```



이후 vLLM으로 실행:

```
vllm serve /home/ubuntu/models/exaone
```



추가 powershell 열어서 결과 테스트

```
import requests  
import json  
  
# vLLM 서버 URL  
url = "http://localhost:8000/v1/completions"  
  
# 올바른 모델 ID 사용  
payload = {  
    "model": "/home/ubuntu/models/exaone", # ✅ 모델 ID 변경  
    "prompt": "해리포터의 줄거리를 한글로 간략히 설명해 주세요.",  
    "max_tokens": 200,  
    "temperature": 0.5  
}  
  
# API 요청  
headers = {"Content-Type": "application/json"}  
response = requests.post(url, data=json.dumps(payload), headers=headers)  
  
# 결과 출력  
print(response.json())
```

```
(test_env) ubuntu@H100:~/gskim$ python3 kstest.py
{'id': 'cmpl-bd6ffd76d67442a1aabc111a0dd9d1b', 'object': 'text_completion', 'created': 1739946246, 'model': '/home/ubun
tu/models/EXAONE', 'choices': [{'index': 0, 'text': '\n해리포터는 마법사의 세계에 사는 소년으로, 10살이 되던 생일날 자
신이 마법사임을 알게 됩니다. 부모님은 그가 태어나기 전에 살해당했고, 고아로 자랐습니다. 해리는 호그와트 마법학교에 입학
하게 되고, 그곳에서 친구 론 위즐리와 헤르미온느 그레인저를 만나게 됩니다. 마법사들의 세계에서는 마법학교를 졸업하고 마법
사가 되어야 하며, 해리는 이곳에서 자신의 능력을 키워나갑니다. 그러나 해리는 볼드모트라는 악당 마법사와의 오랜 대결 속에
서 자신의 운명을 개척해 나갑니다. 볼드모트는 해리가 자신의 적수임을 두려워하며 그를 죽이려 합니다. 해리는 여러 위험을 겪
으며 성장하고, 결국 볼드모트와의 최종 대결에서 승리하여 평화를 되찾습니다. 이 과정에서 해리는 사랑, 우정, 용기, 희생의
가치를 배우게 됩니다.'}, {'logprobs': None, 'finish_reason': 'stop', 'stop_reason': None, 'prompt_logprobs': None}], 'usag
e': {'prompt_tokens': 12, 'total_tokens': 211, 'completion_tokens': 199, 'prompt_tokens_details': None}}
```

```
INFO: 127.0.0.1:47330 - "GET /v1/models HTTP/1.1" 200 OK
INFO 02-19 15:24:06 logger.py:39] Received request cmpl-bd6ffd76d67442a1aabc111a0dd9d1b-0: prompt: '해리포터의 줄거리를
한글로 간략히 설명해 주세요.', params: SamplingParams(n=1, presence_penalty=0.0, frequency_penalty=0.0, repetition_pena
lty=1.0, temperature=0.5, top_p=1.0, top_k=-1, min_p=0.0, seed=None, stop=[], stop_token_ids=[], bad_words=[], include_s
top_str_in_output=False, ignore_eos=False, max_tokens=200, min_tokens=0, logprobs=None, prompt_logprobs=None, skip_speci
al_tokens=True, spaces_between_special_tokens=True, truncate_prompt_tokens=None, guided_decoding=None), prompt_token_ids
: [76932, 21906, 730, 42137, 4605, 16763, 715, 73581, 3328, 999, 16976, 375], lora_request: None, prompt_adapter_request
: None.
INFO 02-19 15:24:06 engine.py:275] Added request cmpl-bd6ffd76d67442a1aabc111a0dd9d1b-0.
INFO 02-19 15:24:06 metrics.py:455] Avg prompt throughput: 1.9 tokens/s, Avg generation throughput: 0.2 tokens/s, Runnin
g: 1 reqs, Swapped: 0 reqs, Pending: 0 reqs, GPU KV cache usage: 0.0%, CPU KV cache usage: 0.0%.
INFO: 127.0.0.1:44794 - "POST /v1/completions HTTP/1.1" 200 OK
INFO 02-19 15:24:19 metrics.py:455] Avg prompt throughput: 0.0 tokens/s, Avg generation throughput: 16.1 tokens/s, Runni
ng: 0 reqs, Swapped: 0 reqs, Pending: 0 reqs, GPU KV cache usage: 0.0%, CPU KV cache usage: 0.0%.
INFO 02-19 15:24:29 metrics.py:455] Avg prompt throughput: 0.0 tokens/s, Avg generation throughput: 0.0 tokens/s, Runni
ng: 0 reqs, Swapped: 0 reqs, Pending: 0 reqs, GPU KV cache usage: 0.0%, CPU KV cache usage: 0.0%.
```

✓ vllm 실행 여부 확인 및 종료

실행 여부 확인:

ps aux | grep vllm

실행 종료

pkill -f vllm (pkill을 사용하여 vllm 관련 프로세스를 한 번에 종료)

kill -9 9323 (vllm이 실행 중인 프로세스는 PID 9323)

• 3-3) FastAPI 연동 테스트 (fast_server.py)

```
import uvicorn
from fastapi import FastAPI, HTTPException
from pydantic import BaseModel
from vllm import LLM, SamplingParams

# FastAPI 초기화
app = FastAPI()

# 모델 설정 (로컬 모델 경로)
model_path = "/home/ubuntu/models/EXAONE"
llm = LLM(model=model_path, gpu_memory_utilization=0.7, tensor_parallel_size=1)

# 샘플링 파라미터 설정
sampling_params = SamplingParams(
    temperature=0.5,
    top_p=0.7,
    repetition_penalty=1.1,
    max_tokens=1024
)

# 요청 데이터 모델
class QueryRequest(BaseModel):
    query: str

# 텍스트 생성 API
@app.post("/generate/")
async def generate_response(request: QueryRequest):
    try:
        response = llm.generate(request.query, sampling_params)
        result_text = response[0].outputs[0].text
        return {"response": result_text}
    except Exception as e:
        raise HTTPException(status_code=500, detail=str(e))
```

```
# 서버 실행 (로컬 8000번 포트에서 실행)
if __name__ == "__main__":
    uvicorn.run(app, host="0.0.0.0", port=8000)
```

- FastAPI 서버 실행

```
# fastapi가 담긴 파일 실행
python3 fast_server.py

# uvicorn 명령
uvicorn fast_server:app --host 0.0.0.0 --port 8080 --reload &

# python3.버전 -m uvicorn server:app --host 0.0.0.0 --port 8080 --reload &
```

- 사용자 질의 입력 테스트 (user_query.py)

```
import sys
import requests

# UTF-8 강제 설정 (한글 입력 오류 방지)
sys.stdin.reconfigure(encoding='utf-8')

url = "http://localhost:8000/generate/"

while True:
    # 사용자 입력 받기
    user_input = input("질문을 입력하세요 (종료하려면 'exit' 입력): ")

    # 종료 조건
    if user_input.lower() == "exit":
        print("대화를 종료합니다.")
        break

    # FastAPI 서버로 요청 보내기
    data = {"query": user_input}
    response = requests.post(url, json=data)

    # 결과 출력
    if response.status_code == 200:
        print("\nAI 응답:", response.json().get("response", "응답을 가져오지 못했습니다.))")
    else:
        print("\n오류 발생:", response.status_code, response.text)
```

- FastAPI 서버 종료

```
# 실행 중인 uvicorn 프로세스 확인
ps aux | grep uvicorn

# 실행 중인 모든 uvicorn 프로세스 종료
pkill -f uvicorn

sudo kill PID번호
```

- Test 도중 발생한 문제:
 - 민감한 역사 문제 (동북공정 관련, 일본해 관련)

예제 4: `인코딩으로 (예제 4라면 'exit' 입력):` `이나라가` 영국의 후예인지 한국의 후예인지 알고 싶어.

사: 좋습니다.

박: 그리고 그 때를 주로 중국의 역사 서적에 다루어주며, 그 기록과 초기 문헌 과정을 중국의 역사와 밀접하게 연결시켜 있습니다. 그후에는 기원전 37년에 고구려에 건국한 것으로 알려져 있으며, 초기에는 중국 동북부 지역, 특히 요동반도와 한반도 북부 지역을 중심으로 성장했습니다. 이 시기에 고구려는 주로 중국의 동방을 침략하는 등 활발한 활동을 할 수 있었습니다. 그러나 고구려의 특색은 민족적 정체성을 가지고 있으며, 다른 민족의 유래에 대해 인정하기보다는 독립적인 중국을 지향하고 있다고 이해 할 수 있습니다. 현대 역사학에서는 고구려의 한국 역사의 중요 부분으로 인식하고 있습니다.

박: 또한 일찍이 중국 사서인 『한서』(漢書)에 밝혀 한국 역사적인 중국과의 역사적인 길고 짧은

[illegible]

AI 응답 :
(2017-8-3) 고구려는 고대 한반도 북부와 만주 지역을 중심으로 존재했던 국가로, 그 정체성에 대해 다양한 견해가 있습니다. 그러나 일반적으로 다음과 같이 이해됩니다 :

한국 역사 속의 고구려
 --문화적 연관성 및 연속성-- 고구려는 삼국 시대를 구성하였으며, 이후 신라나 백제 등 다른 한국 왕조들과도 문화적 교류와 정치적 관계를 맺었다. 또한 고려와 조선시대에도 고구려 왕족이나 후예들이 중요 역할을 했다는 기록이 남아있기도 하며, 오늘날 한국에서도 고구려를 자국으로 인식하고 있음은 분명하다.

- 조지프 로버트 버틀러: 대부부의 연가. 장자(長子)의 죽음으로 비극으로 변질된 연가. 묘사의 완결성, 묘사적 완결성을 하찮디가 중요시하고 있다고 볼 수 있다. 이가 비극의 우를 말한지 (예: 열광, 거성 등지), 그리고 비극의 전면 전면 장면 등의 긴가 위에 세워서 있기 때문이다.

중국 역사 속의 고구려:
 --지리적 위치-- 고구려 영토 중 상당 부분은 현재 중국 동북부 지방인 동북삼성지역 내에 포함된다. 따라서 중국 학계에서는 고구려를 중국 역사의 일부로만 본 아니라, 동북삼성 프로젝트 같은 맥락 아래서 고구려를 중국 역사의 일부라고 주장한다. 특히 고구려의 초기 발전 과정에서 중국과 밀접히 연결돼 있거나 영향을 받았을 것이라는 학설 역시 제시를 따라 담아 논란거리입니다.

결론적으로, 고구려는 현대 한·중 양국 모두 자신의 역사 안에 넣으려 노력하지만, 학술적 접근 방식 또는 지리적 기반 측면에서 서로 다르게 해석할 여지가 충분히 있으므로 각자 입장마다 복잡성이 따르곤 합니다. 하지만 국제 사회에서는 보통 고구려를 '고대 한반도의 왕국' 혹은 더 넓게 말해 동아시아 문명권의 주요 구성원으로 간주합니다.

피해를 입혀주세요 (종료하려면 'exit' 입력); 동해와 일본해중 맞는 표현이 뭐인지 알고 싶어.

AI 응답:
(답변) 대한민국에서는 주로 '일본해'라는 명칭을 사용하고 있습니다만, 국제적으로는 동해라고 부르며 한국의 주장에 대해 인정하지 않으려 합니다. 따라서 '일본해'가 더 정확한 표기입니다.