Lawbot 서비스 개발

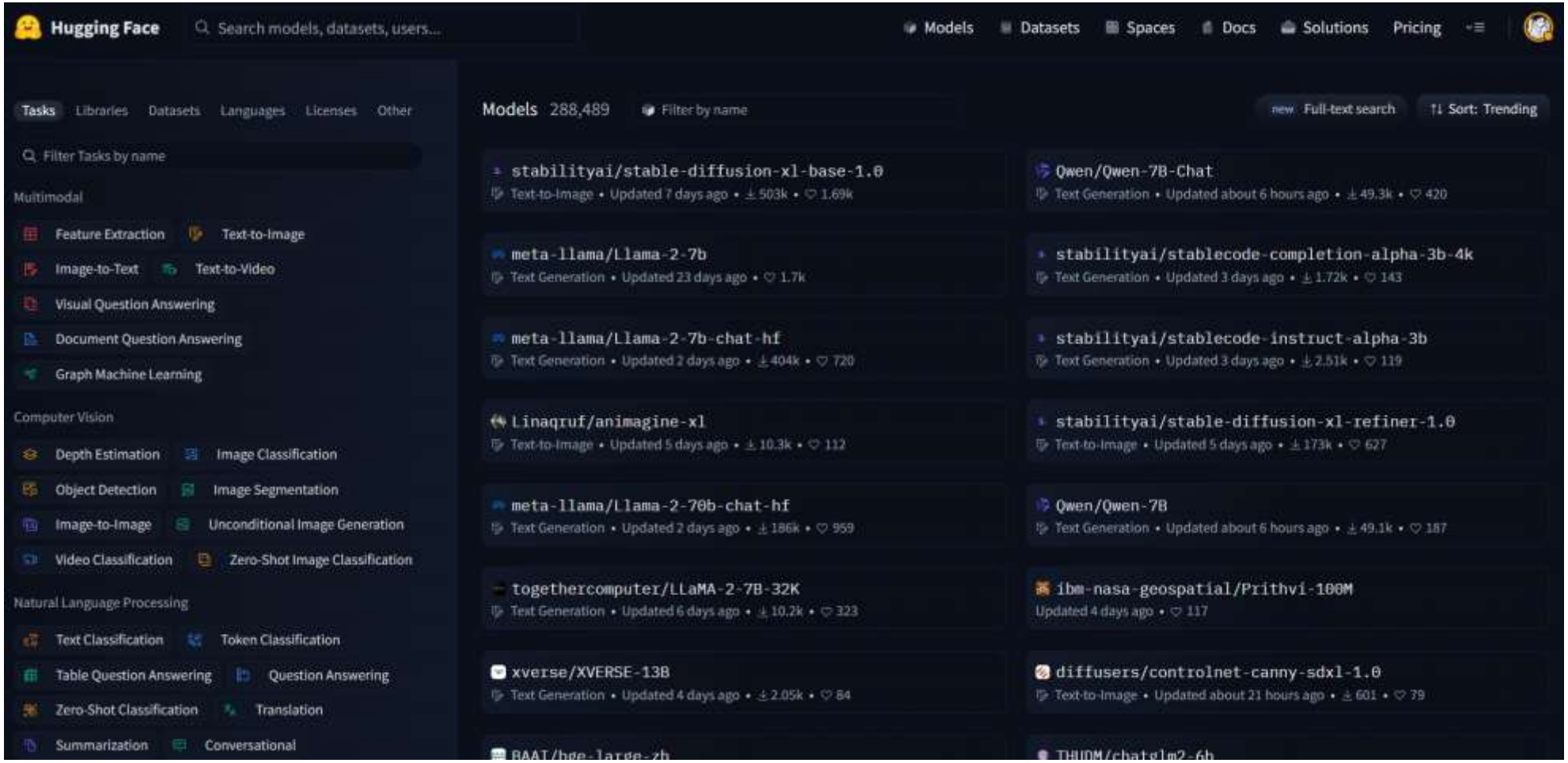
Development Method of Legal-Llama2-ko Model
(https://github.com/taemin6697/level3_nlp_finalproject-nlp-08/tree/main)

한성대학교 김태민

목차

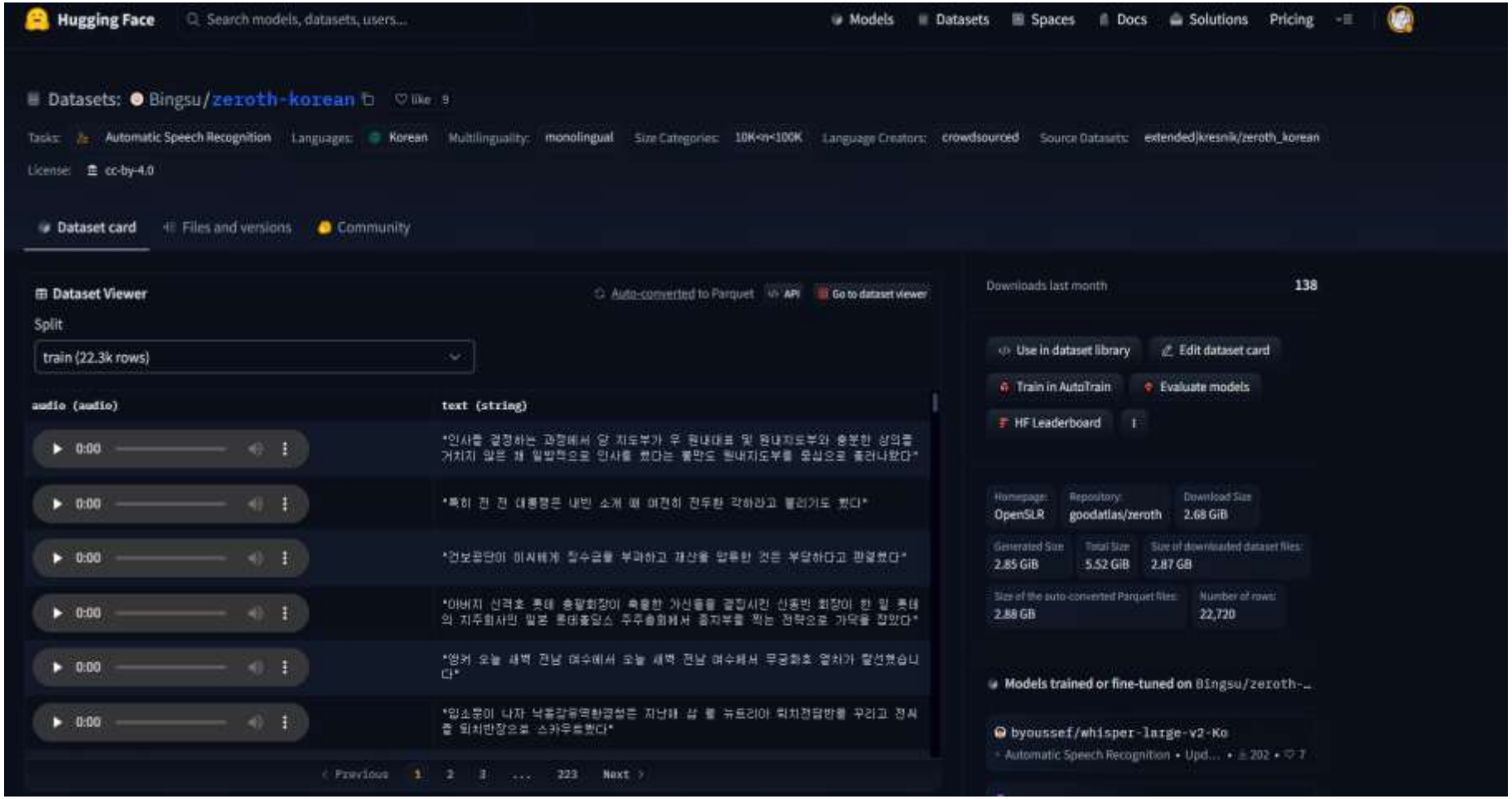
Introduction the HuggingFace

What is HuggingFace Model



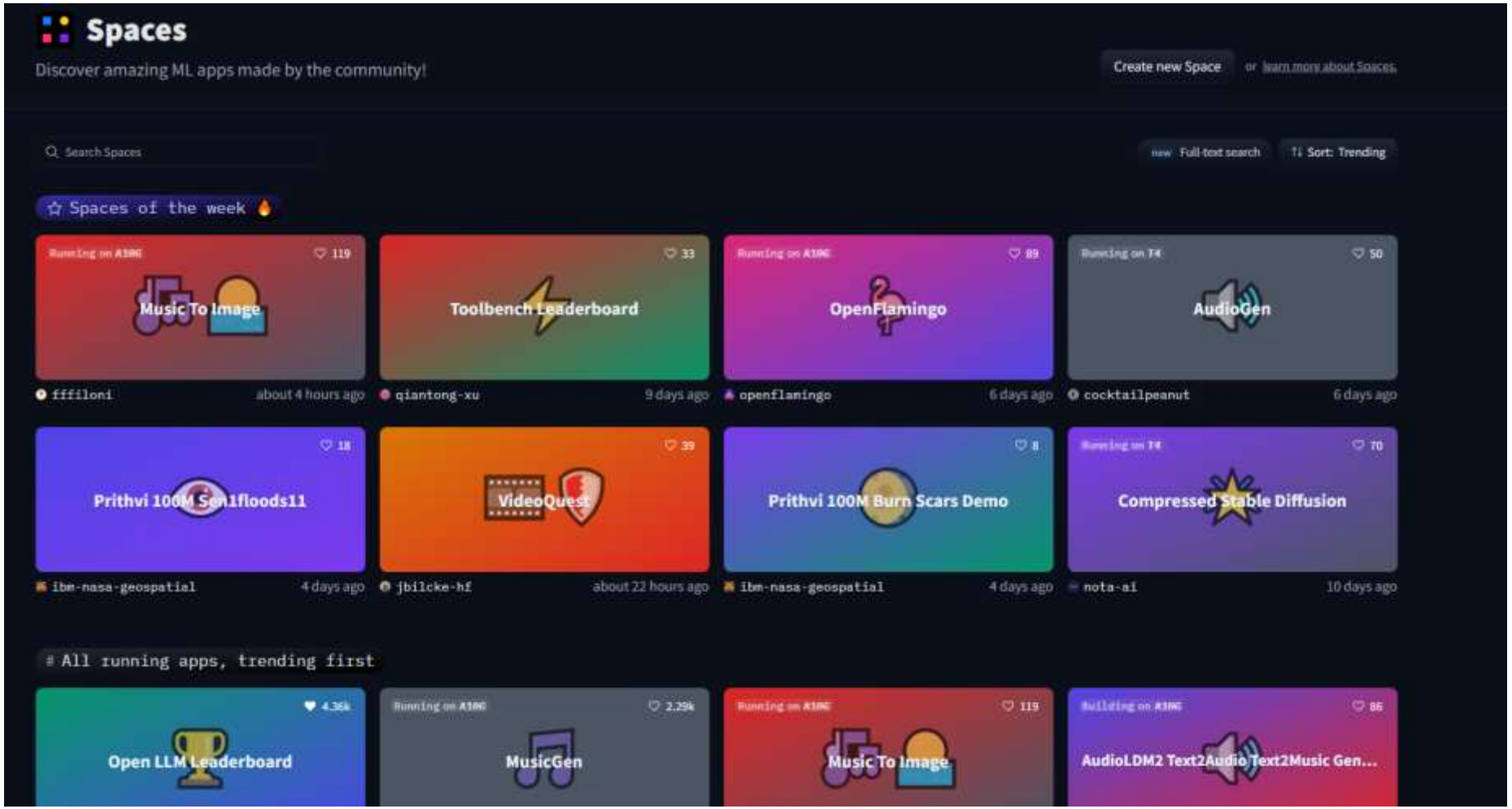
Introduction the HuggingFace

What is HuggingFace Datasets



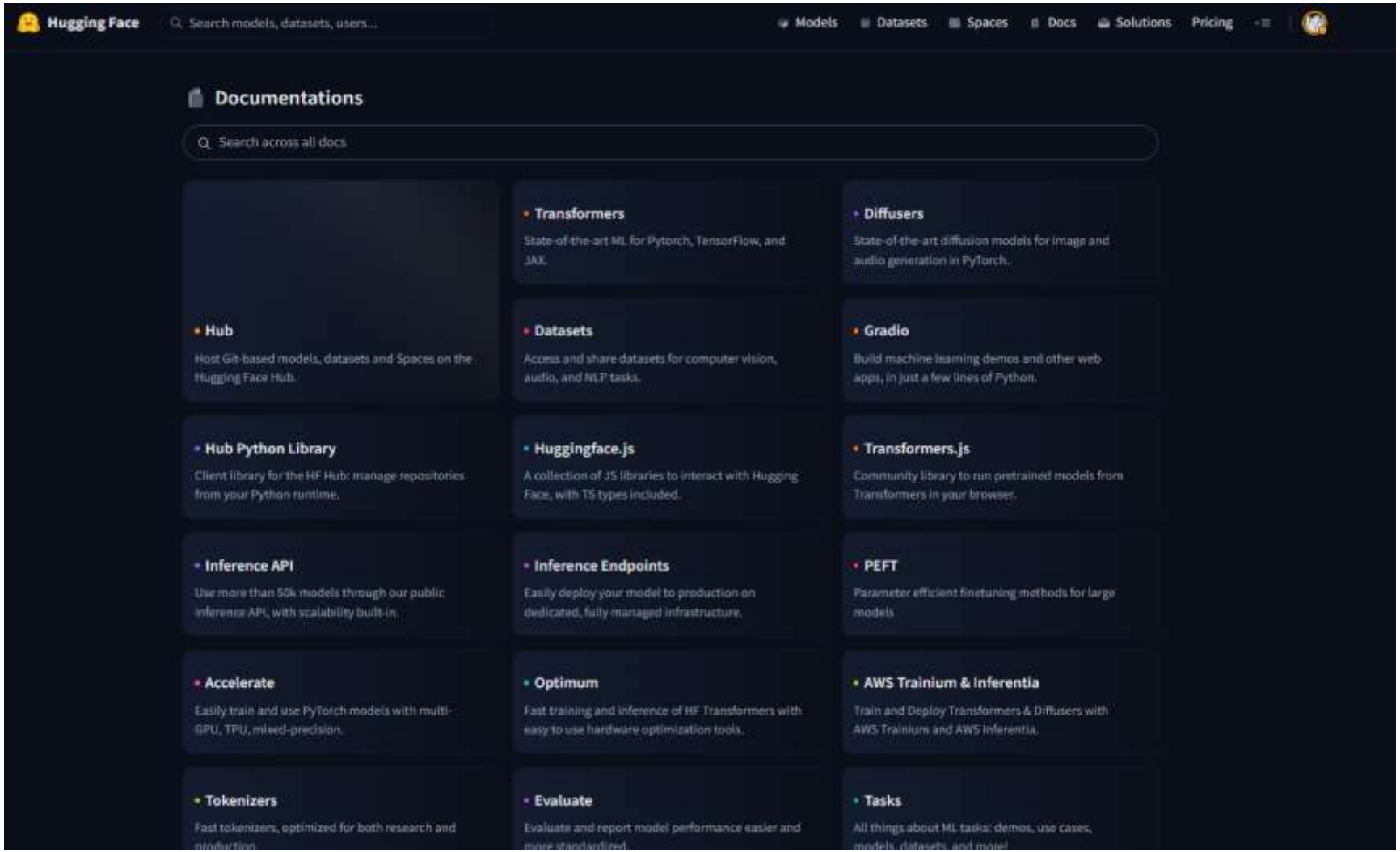
Introduction the HuggingFace

What is HuggingFace Spaces



Introduction the HuggingFace

What is HuggingFace Docs



Introduction the HuggingFace

Deep Learning Model Training Using Hugging Face

Pytorch training step

1. 모델 소스 코드 작성 (수백줄)
2. 옵티마이저 및 각종 파라미터 코드 작성 및 선언 (수백줄)
3. 로깅 함수 작성
4. 데이터 셋 로드 코드 작성 (수백 줄)
5. 데이터 셋 전처리 코드 작성
6. 데이터 셋 로더 코드 작성
7. 모델 훈련 코드 작성 (수백 줄)

일반적으로 최소 1000 줄 이상의 코드 작성이 요함

HuggingFace training step

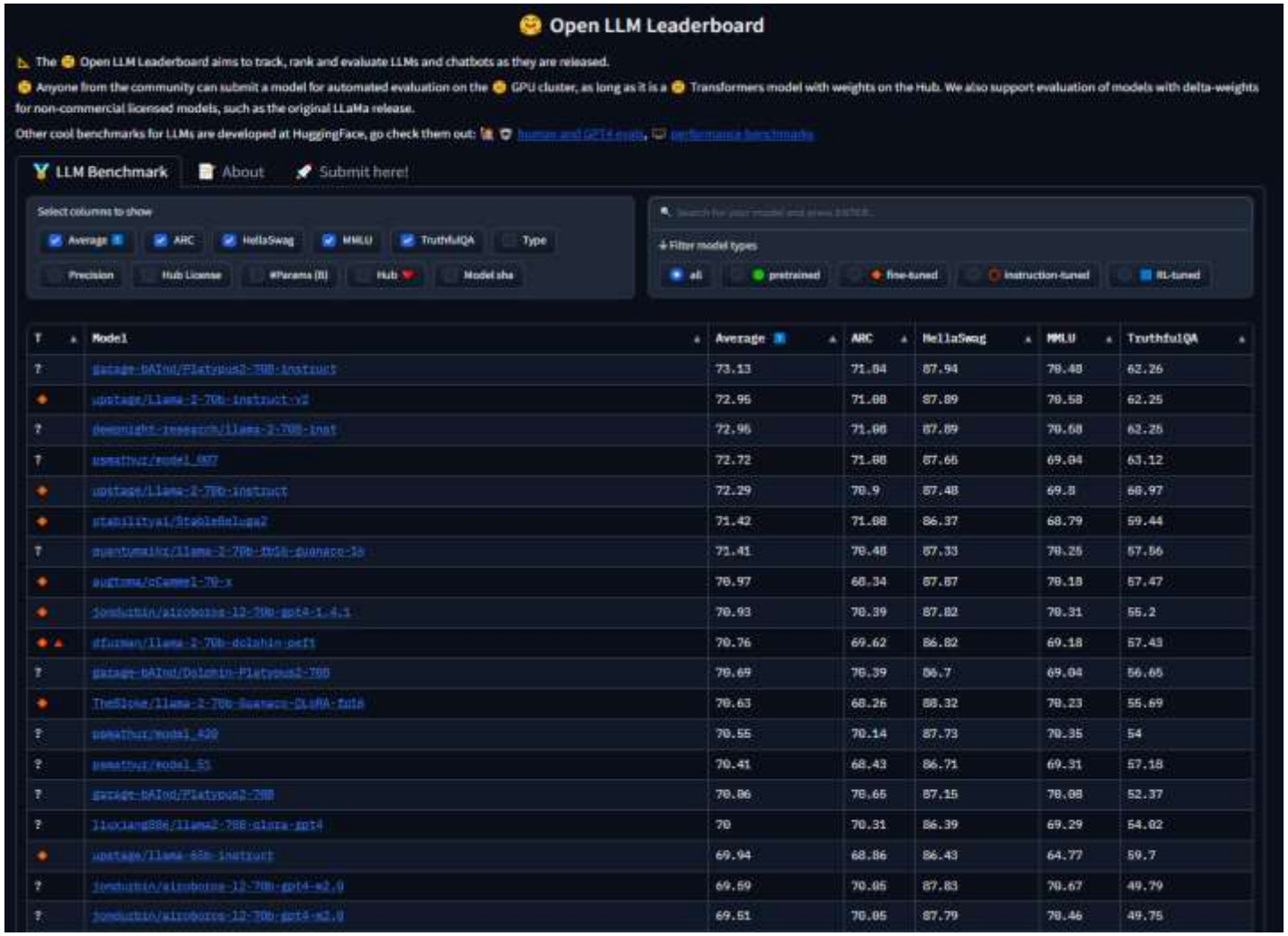
1. 불러올 모델 이름 작성 (한 줄)
2. 로드할 데이터 셋 이름 작성 (한 줄)
3. 데이터 셋 전처리 코드 작성
4. 모델 훈련 코드 작성 (50줄 내외)

약 100 줄내외의 코드 작성이 요함

실제 법률 GPT를 만들때 사용된 코드 200줄 내외!

How to use sLLM

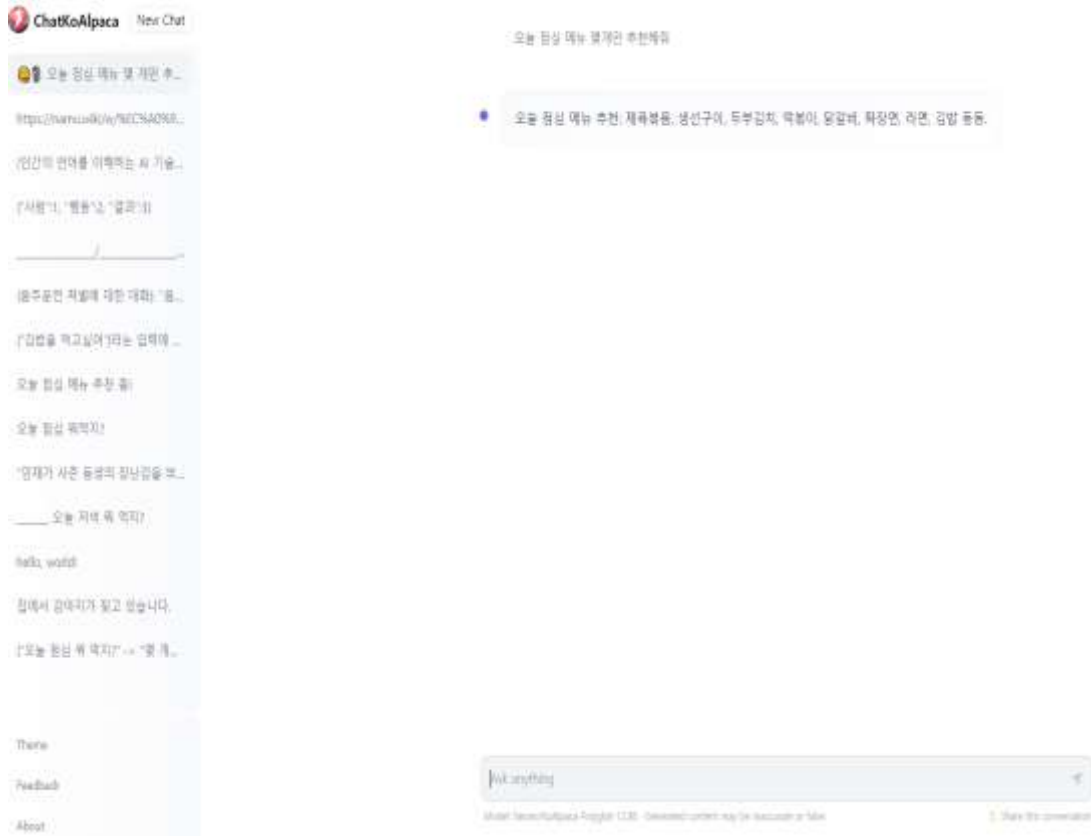
HuggingFaceH4/open_llm_leaderboard



How to use sLLM

Korean Chat sLLM list

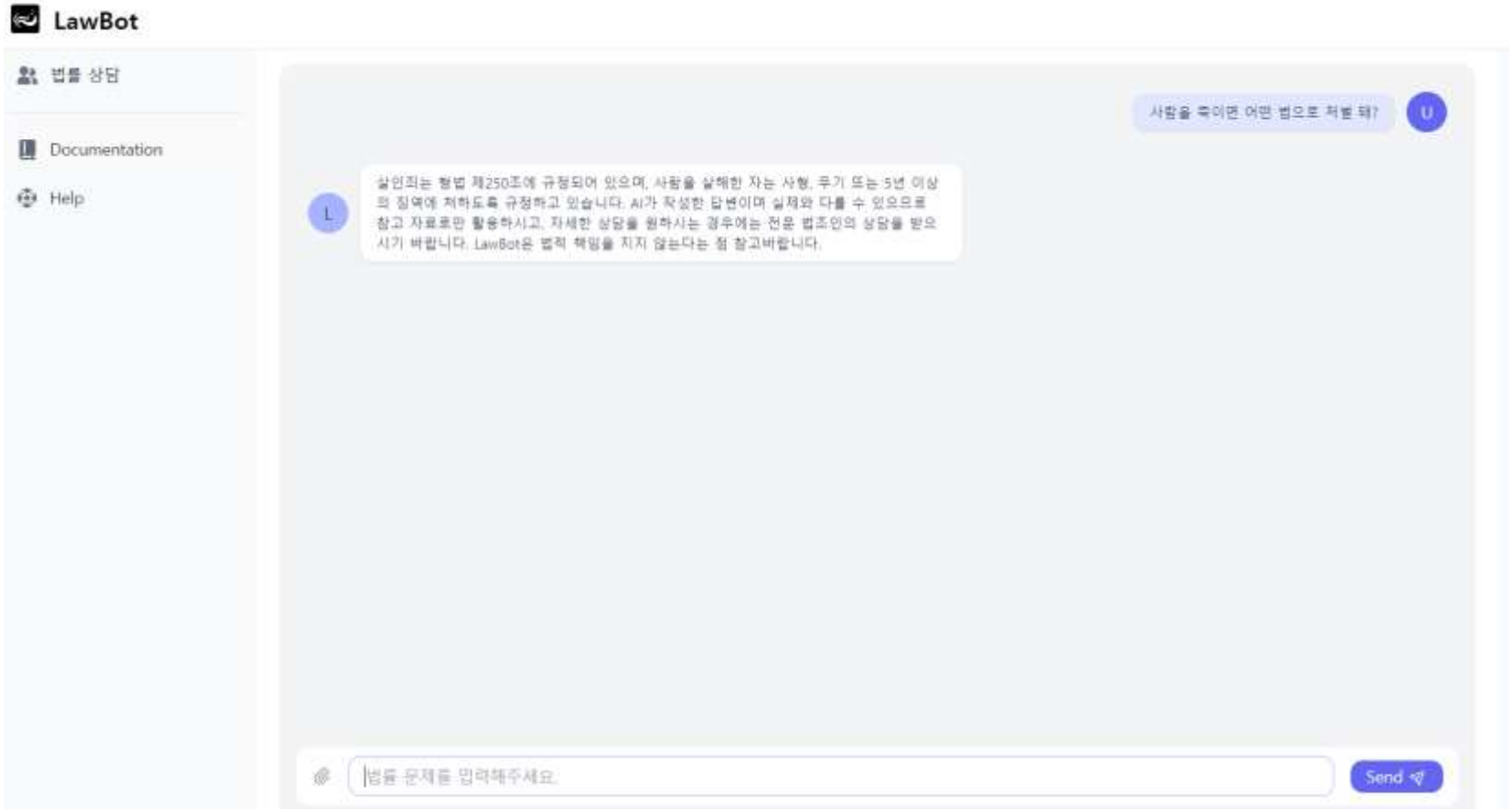
KuLLM



KoAlpaca

How to use sLLM

Korean Chat sLLM list



LLama-2-ko-Chat

What is Supervised fine-tuning

Regular LLM vs Chat LLM

질문 : 삼원색에 대해 알려줘

Llama-2-ko	[C1, R0.625 G8 B275\u200b3-4, 삼각형 그리기.\u200b아래에는 삼각형을 그립니다 (그림을 그려야 할 영역의 좌표를 지정) - 명령어와 함께 사용되는 문자열로 입력할 좌표 값과 동일한 결과를 얻음 - 예를 들어 "X=-19. Y=0" 으로 입력을 하면 좌표값을 지정된 값과 같게 출력을 한다. ##########################\u200b##3_0-6-1, 좌표 입력으로 텍스트 파일을 생성하라(ex C4 D6 E10 A10)-명령문에서 각 문자를 입력과 동일하다. -이 명령은 현재 수행중인 프로그램을 종료한다.-파일은 실행 가능한 파일과 일치할]
Llama-2-ko-Chat	[물체를 인식하고 색상에 대한 시각적인 이미지를 생성하기 위해 두 개의 눈은 서로 반대쪽 방향에서 물체의 색상 정보와 명암 정보를 받아들이기 때문에, 인간은 일반적으로 색상을 삼원색이라고 하는 세 가지 원추 형태(빨강, 녹색, 파랑색)로 분류합니다. 이 중 한가지 색상이 다른 색상보다 눈에 훨씬 더 빨리 들어오는 경향이 있다고 합니다. 하지만 모든 사람이 그렇지는 않으므로 항상 삼각형 모양으로 색상을 분류하지는 않습니다. 하지만 삼원색이 우리 눈에 잘 전달되며 색상 구별에 중요하다는 것은 부정할 수 없습니다.]

What is Supervised fine-tuning

How to Train?

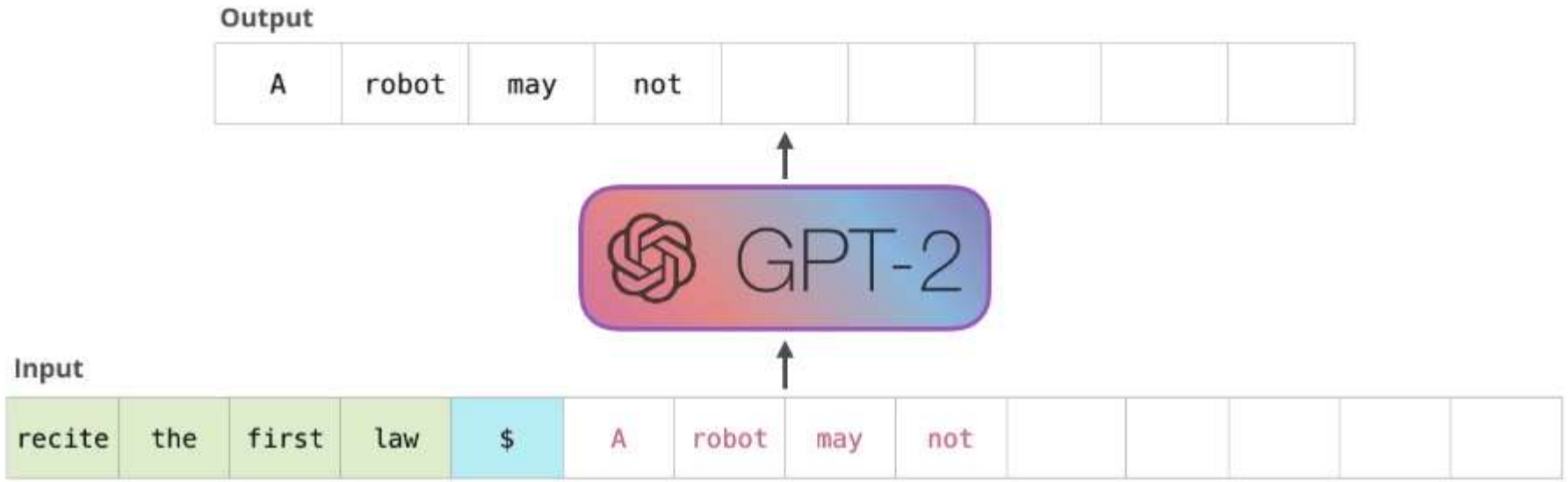
"alpaca_{idx}"	"컴퓨터 마더보드의 기능 설명하기"	""	"메인보드 또는 시스템 보드라고도 하는 마더보드는 컴퓨터의 중앙 인쇄 회로 기판입니다…
"alpaca_{idx}"	"다음 문장을 수정하여 더 간결하게 만듭니다."	"그는 5분 후에 도착할 버스를 타기 위해 버스 정류장으로 달려갔습니다."	"그는 5분 후에 도착하는 버스를 타기 위해 버스 정류장으로 달려갔습니다."
"alpaca_{idx}"	"프랑스의 수도는 어디인가요?"	""	"프랑스의 수도는 파리입니다."
"alpaca_{idx}"	"다음을 동물, 식물, 광물로 분류합니다."	"참나무, 구리 광석, 코끼리"	"동물: 코끼리 식물 참나무 광물 광물: 구리 광석"
"alpaca_{idx}"	"10줄의 시를 생성합니다."	""	"화려하고 밝은 꽃으로 가득한 초원, 따스한 햇살에 끝없이 펼쳐진 푸른 하늘, 순수하고…
"alpaca_{idx}"	""사과하다"와 같은 의미의 동사 세 개 생성하기"	""	"1. 유감 표명하기 2. 속죄하다 3. 보상하기 위해"
"alpaca_{idx}"	"한 변의 길이가 5cm인 정육면체의 총 표면적을 계산합니다."	""	"정육면체의 표면적은 정육면체 중 하나의 면적을 계산한 다음 6을 곱하면 구할 수 있습…
"alpaca_{idx}"	"능동태를 사용하여 다음 문장을 다시 작성합니다."	"기장이 뉴스 보도를 읽었습니다."	"선장은 뉴스 보도를 읽었습니다."
"alpaca_{idx}"	"주어진 문장의 단어들을 배열하여 문법적으로 올바른 문장을 만들 수 있습니다."	"갈색 여우가 빨리 뛰었습니다."	"갈색 여우가 재빨리 뛰어올랐어요."
"alpaca_{idx}"	"이 코드를 리버스 엔지니어링하여 새 버전 만들기"	"def factorialize(num): 계승 = 1 for i in range(1, num): 계승 *= i 반환 계승"	"다음은 재귀를 사용하여 숫자의 계승을 계산하는 새 버전의 코드입니다: def…

What is Supervised fine-tuning

How to Train?

Question : 프랑스의 수도는 어디인가요?

Answer : 프랑스의 수도는 파리입니다.



프랑스의 | 수도는 | 어디인가요 | ? | $ | 프랑스의 | 수도는 | 파리

http://jalammar.github.io/illustrated-gpt2/

Efficient fine-tuning using LoRA-tuning

What is LoRA? (Low-Rank Adaptation of Large Language Models)

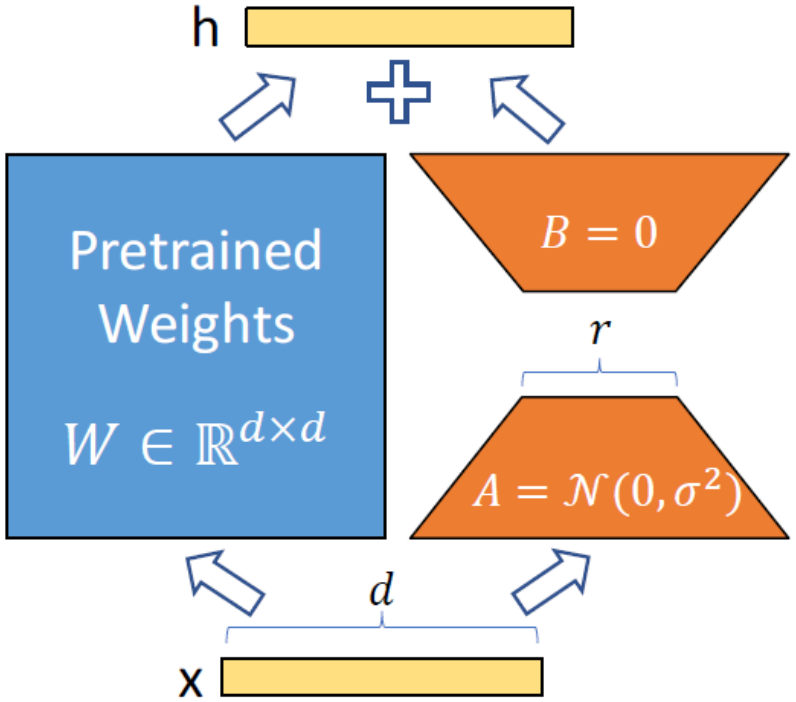


Figure 1: Our reparametrization. We only train A and B.

기존 모델의 가중치는 동결(freeze)

원본 모델이 업데이트가 되지 않음으로 VRAM절약

추가된 일부분의 모델 레이어만 훈련

LoRA 어댑터를 탈부착 가능함으로써 다양하게 모듈로 부착시킬 수 있음

Efficient fine-tuning using LoRA-tuning

HuggingFace LoRA Code

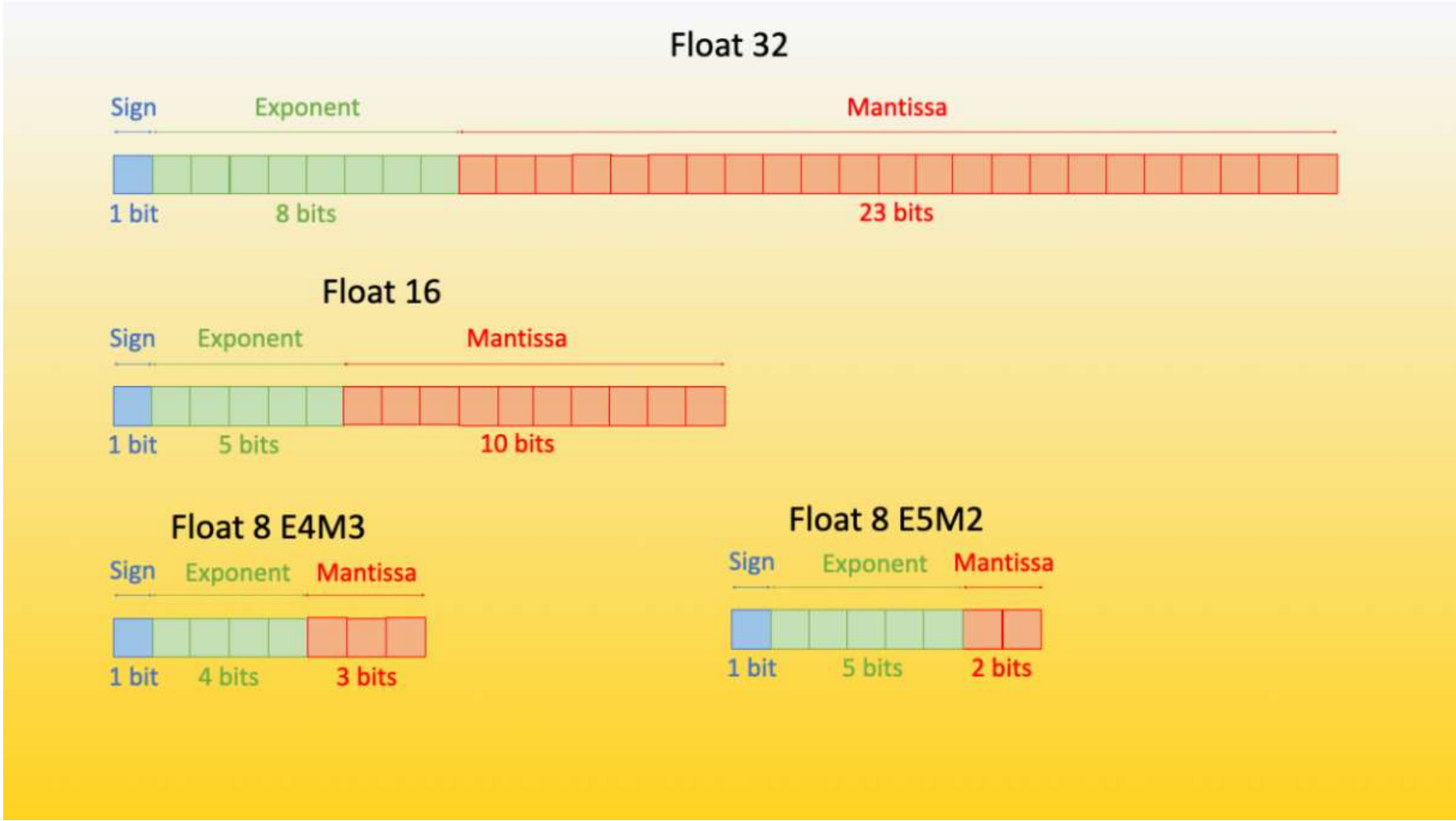
```
config = LoraConfig(
    r=8,
    lora_alpha=32,
    target_modules=["query_key_value"],
    lora_dropout=0.05,
    bias="none",
    task_type="CAUSAL_LM",
)

model = get_peft_model(model, config)
print_trainable_parameters(model)

return model, tokenizer
```

Efficient model loading and training using Bitsandbytes

What is Bitsandbytes



Efficient model loading and training using Bitsandbytes

Bitsandbytes Code

```
def load_model(model_name):
    bnb_config = BitsAndBytesConfig(
        load_in_4bit=True,
        bnb_4bit_use_double_quant=True,
        bnb_4bit_quant_type="nf4",
        bnb_4bit_compute_dtype=torch.bfloat16,
    )
    tokenizer = AutoTokenizer.from_pretrained(model_name)
    model = AutoModelForCausalLM.from_pretrained(
        model_name, quantization_config=bnb_config, device_map={"": 0}
    )
    model.gradient_checkpointing_enable()
    model = prepare_model_for_kbit_training(model)
```

Efficient sLLM training method

04/ Model Architecture – 진행 사항

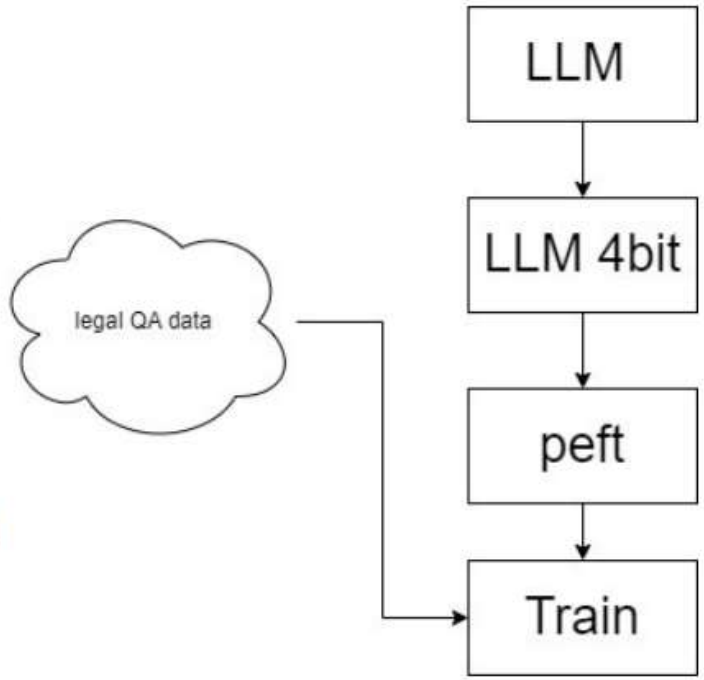
활용한 프롬프트

Law LLM Model

- KULLM : 아래는 작업을 설명하는 명령어입니다. 요청을 적절히 완료하는 응답을 작성하세요.₩n₩n### 명령어:₩n{x['question']}₩n₩n### 응답:₩n{x['answer']}

모델 학습 방법

- 현재 훈련은 **LoRA: Low-Rank Adaptation of Large Language Models** 논문을 참고하여 전체 훈련 파라미터의 0.1%만 Fine Tuning 하는 방식으로 학습
- 채팅 형식으로 학습된 모델에 법률 QA 데이터를 학습하여 모델을 구성
- bitsandbytes(4bit) : 32bit에서 4bit로 계산하여 모델의 크기 감소
- PEFT(Parameter-Efficient Fine-Tuning) : 기존의 원본 LLM의 가중치를 동결 ➔각 레이어 마다의 추가 선형 레이어(LoRA)를 삽입 ➔ 이를 훈련 시켜 VRAM의 감소와 훈련 계산량의 감소로 모델을 Full Fine-Tuning에 가까운 학습 가능
- 모든 파라미터를 학습하는 방식이 아니기에 기학습에 사용된 프롬프트를 그대로 학습에 사용.



Follow-up Research and Advancement Plan

[고찰 - 모델 & 개선 방안]

RLHF

- 실제 변호사가 직접 데이터를 만들어 점수를 평가하여야 보상모델을 만들고 RLHF를 이용하여 학습 시킨다면 성능을 개선할 수 있다고 생각합니다.

Full Finetuning Pretrained model

- 대부분의 한국 Pretrained model에는 한국법률이 추가되지 않았습니다. 따라서 많은 Hallucination과 정확하지 않은 답변을 출력하게 됩니다. 따라서 보다 고성능의 GPU를 이용하여 model을 처음부터 한국 법률을 포함시켜 훈련하게 될 경우 보다 높은 정확성을 달성 할 것이라고 생각합니다.

파라미터의 한계

- 실제 서비스되는 대형 LLM의 경우 30B, 65B, 70B의 대규모 언어 모델로 서비스가 진행됩니다. LLM의 특성상 모델의 파라미터가 많을수록 보다 높은 정확성을 달성하게 되는데 현재 가진 GPU의 한계로 최대 12.8B이며 인퍼런스의 경우 5.8B를 사용할 수 밖에 없었습니다. 보다 고성능 GPU를 사용하여 많은 파라미터를 가진 모델을 사용하게되면 성능이 향상될 것입니다.

Development environment

1) 개발에 필요한 하드웨어 구성
   - 클라우드 환경
     Google Colab 환경
     AWS 클라우드
     KT 클라우드
   - GPU 구매
     V100 (1500~2000만원)
     A100(2000만원 이상)
   - 국내 LLM 클라우드 제공 업체
     (솔트룩스 루시아GPT, 업스테이지)

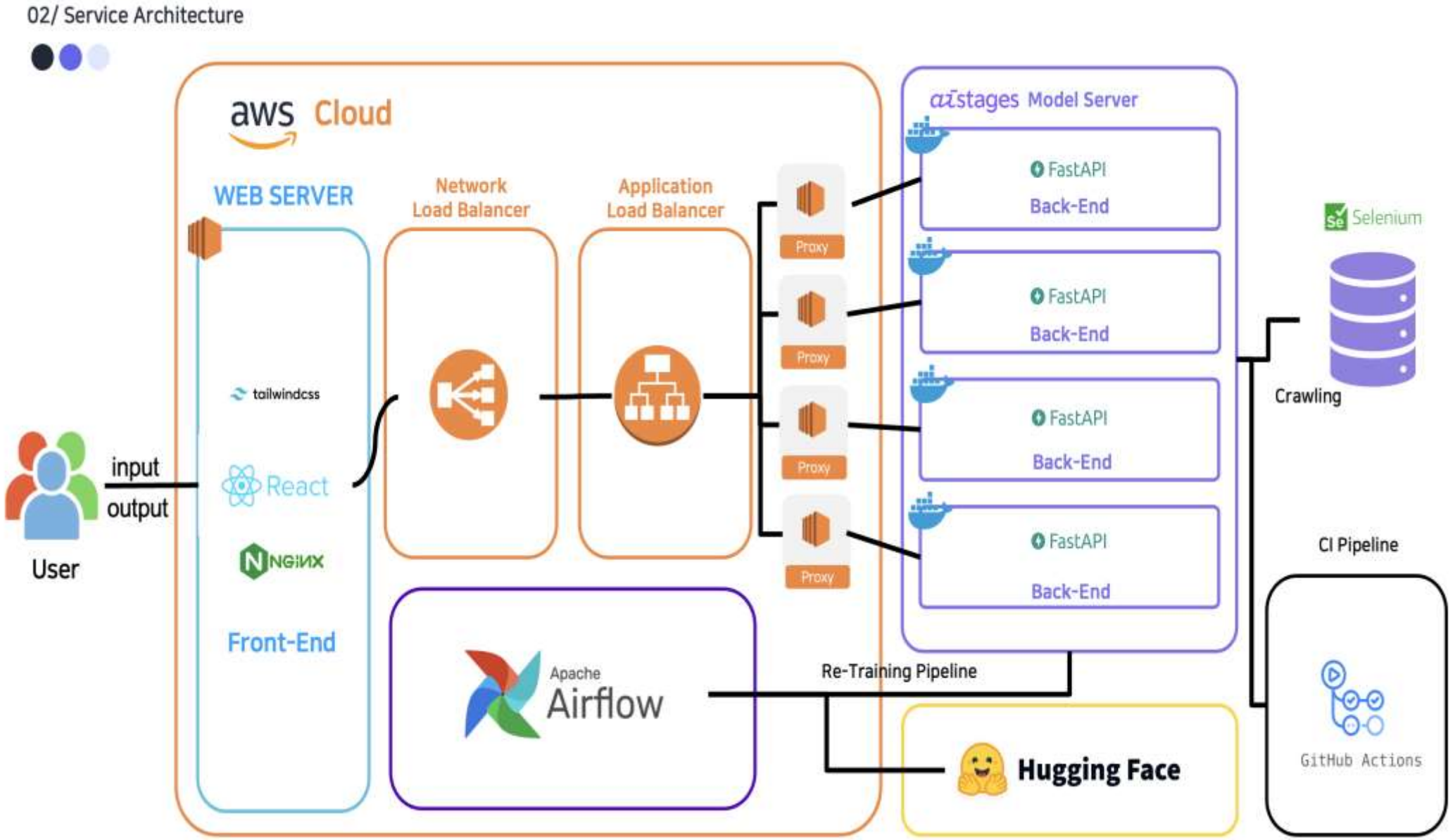
2) 개발 인력 구성 (최소 인력)
   - 백엔드, 프론트엔드, 데이터 엔지니어, ML 엔지니어 등 다수

3) 개발 기간
   - 실제 개발 기간

Development environment

내용	6월 1주차	6월 2주차	6월3주차	7월 1주차	7월 2주차	7월3주차	7월4주차
아이디어 계획							
아이디어 계획							
개발 계획							
웹 프론트엔드 개발							
와이어프레임 작성							
UI/UX + 프로토타이핑							
웹 프론트엔드 페이지 구축							
데이터 파이프라인							
데이터 탐색 & 크롤러 모듈 개발 및 데이터 수집							
데이터 전처리							
생성모델 기반 데이터 증강							
모델							
LLM 법률 조언 모델 구축							
BERT/BM25 Retrieval 기반 유사판례 추출모델							
BERT기반 질문 Classifier							
프로덕트 서빙							
백엔드 구축(API 개발)							
CI/CD 파이프라인 구축							
로드 밸런싱							
Airflow를 이용한 재학습 파이프라인							

Service Archiecture



Q&A