programmers

Foundation 모델 활용해보기

Foundation 모델과 HuggingFace에 대해 알아보자

한기용

keeyonghan@hotmail.com

programmers

Contents

programmers

1장 퀴즈 리뷰

1장 내용을 다시 한번 점검해보자: 퀴즈 링크

programmers

Foundation 모델의 종류 알아보기

어떤 Pre-trained 모델들이 있는지 알아보자

Foundation 모델이란?

- 대규모 사전 학습 (Pre-Trained) 모델이라고도 하는 Foundation 모델은 특히 자연어 처리(NLP), 컴퓨터 비전 등 인공 지능의 다양한 분야에 혁신을 가져옴
- 이러한 모델은 광범위한 데이터 세트에 대해 학습되며 Fine-Tuning 가능함
- 이러한 모델의 사용법은 마치 프로그래밍에서 라이브러리 사용하는 것과 흡사
  - 두 가지 사용법이 존재
    - 그대로 사용
    - Fine-Tuning해서 사용
  - 뒤에서 HuggingFace 기반으로 예를 들어볼 예정

분야별 대표 Foundation 모델

- NLP 모델
  - 이젠 대부분 Transformer 기반의 모델들이 대세. GPT, BERT 등등
- 비전 모델
  - ResNet, VGG, Inception과 같은 CNN 기반 모델들
  - Vision Transformers (ViT)처럼 Transformer 기반의 모델들도 나오고 있음
- 멀티모달 모델
  - GPT4가 가장 대표적
- 오디오/스피치 모델
  - WaveNet, BERT for Audio

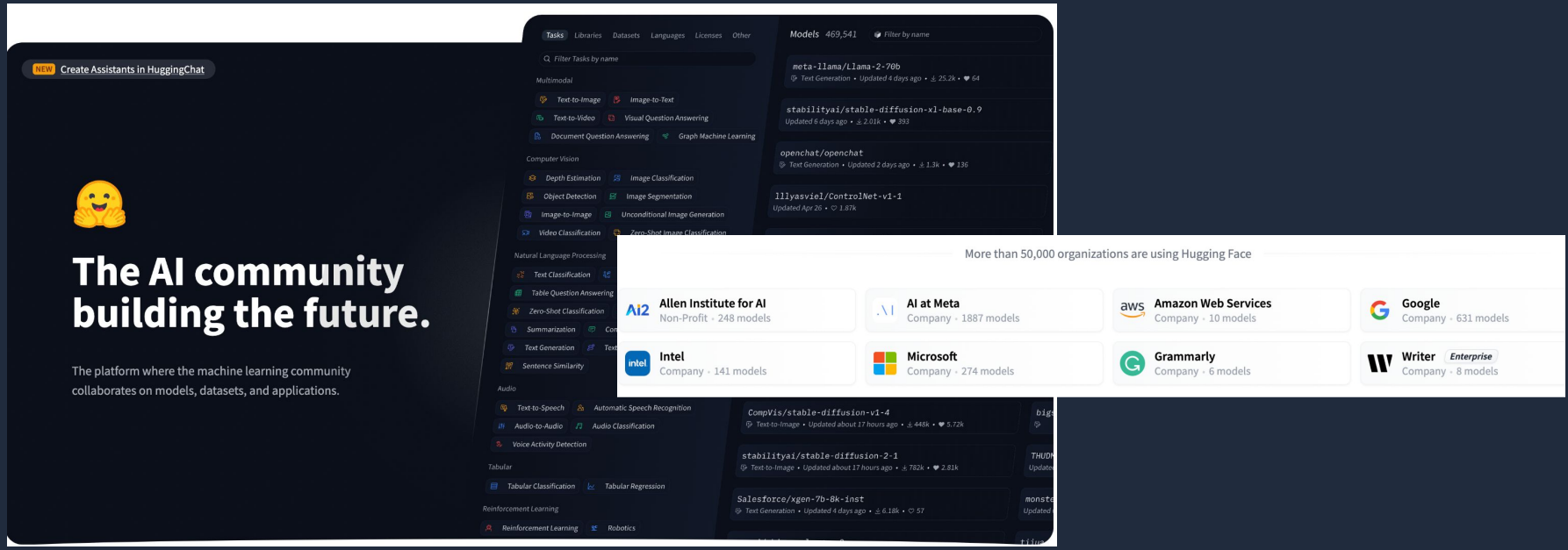
programmers

HuggingFace 소개

HuggingFace가 무엇인지 소개해보자
2016년 뉴욕에서 프랑스 AI 과학자들이 창업한 회사

Open Source AI 개발 관련 최고의 사이트

- https://huggingface.co/
- 최고의 기업들이 후원하고 있음 (클라우드 파트너, 하드웨어 파트너 등등)

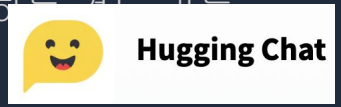


HuggingFace가 제공해주는 기능들 (1)

- HuggingFace Hub: 모델과 코드와 데이터셋 저장소. 기업용 버전도 존재
  - 데이터셋 (100,000+)
  - 모델 (480,000+)
  - Git 기반의 코드 리포
- 다양한 오픈소스 기반 AI 모델링 라이브러리
  - AutoTrain, **Transformer**, Diffuser, Accelerate, Optimum
  - 굉장히 사용하기 쉽고 직관적으로 되어있음

HuggingFace가 제공해주는 기능들 (2)

- ML 애플리케이션 서비스
  - Spaces: 사용자들이 머신 러닝 애플리케이션을 구축하고 공유할 수 있는 플랫폼으로 Hugging Face 에코시스템을 기반으로 구축된 앱을 호스팅할 수 있는 사용하기 쉬운 인터페이스를 제공
    - 16GB memory와 2 vCPU 제공
  - Inference Endpoints: 모델 추론 서비스
    - 하지만 모두 오픈소스라 원하면 직접 하는 것도 가능
- HuggingChat



  - 오픈소스 챗봇: https://huggingface.co/chat/
- HuggingFace 자체 모델도 존재
  - Bloom: HuggingFace가 주관해서 만들어진 오픈소스 기반의 언어 모델
  - StarCoder: 코드용 대규모 언어 모델(Code LLM)
  - Idefics: 딥마인드에서 처음 개발한 최첨단 시각 언어 모델인 플라밍고를 기반으로 하며 GPT-4와 마찬가지로 이 모델은 임의의 이미지 및 텍스트 입력 시퀀스를 받아들이고 텍스트 출력을 생성

HuggingFace 인기 모듈 1

- transformers:
  - 텍스트 분류, 정보 추출, 질문 답변, 요약, 번역 등 다양한 NLP 작업을 위해 설계된 광범위한 사전 학습된 모델을 제공하는 자연어 처리(NLP)를 위한 종합 라이브러리
  - TensorFlow 및 PyTorch와 같이 널리 사용되는 딥 러닝 프레임워크를 기반으로 구축되어 연구자와 실무자에게 유연성과 사용 편의성을 모두 제공
  - Pre-trained 모델로는 BERT, GPT (GPT-2 and GPT-3), RoBERTa, T5, DistilBERT 등이 존재
- datasets:
  - NLP 영역에서 데이터셋을 로드, 처리 및 평가하고 다른 머신 러닝 작업을 위해 설계된 강력하고 사용하기 쉬운 라이브러리
  - transformer 라이브러리를 보완하는 Hugging Face 에코시스템의 일부

HuggingFace 인기 모듈 2

- AutoTrain
  - ML 모델 빌딩 프로세스를 자동화해주는 라이브러리
- Diffuser
  - Diffusion 모델을 훈련하거나 finetuning하거나 배포 가능하게 해주는 라이브러리
- Accelerate
  - 하드웨어(CPU, GPU, TPU)에 관계없이 모델을 훈련하고 실행 가능하게 해주는 라이브러리
- Optimum
  - Foundation 모델의 fine-tuning을 특정 하드웨어(Intel's OpenVINO, NVIDIA's TensorRT)위에서 최적화해주는 라이브러리
- Youtube 학습 플레이 리스트

HuggingFace LLM의 사용 모드

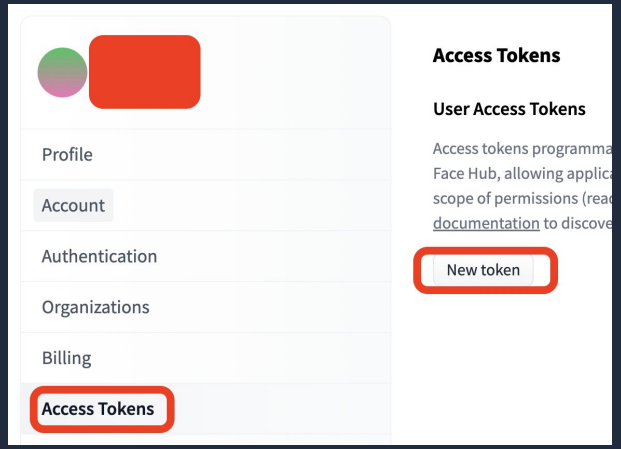
- Text Generation
- Text to Text Generation
  - Question Answering
  - Summarization
  - Translation
  - …

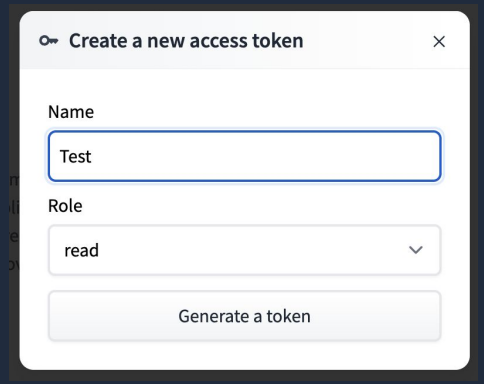
HuggingFace LLM의 사용 방법

- 하나는 모델을 다운로드 받아서 로컬에서 사용하는 방법
  - 로컬 컴퓨터의 사양에 따라 불가능한 모델도 있음
- 두번째는 HuggingFace Hub에 있는 모델을 API 형태로 사용하는 방법

HuggingFace Hub LLM의 사용방법: API

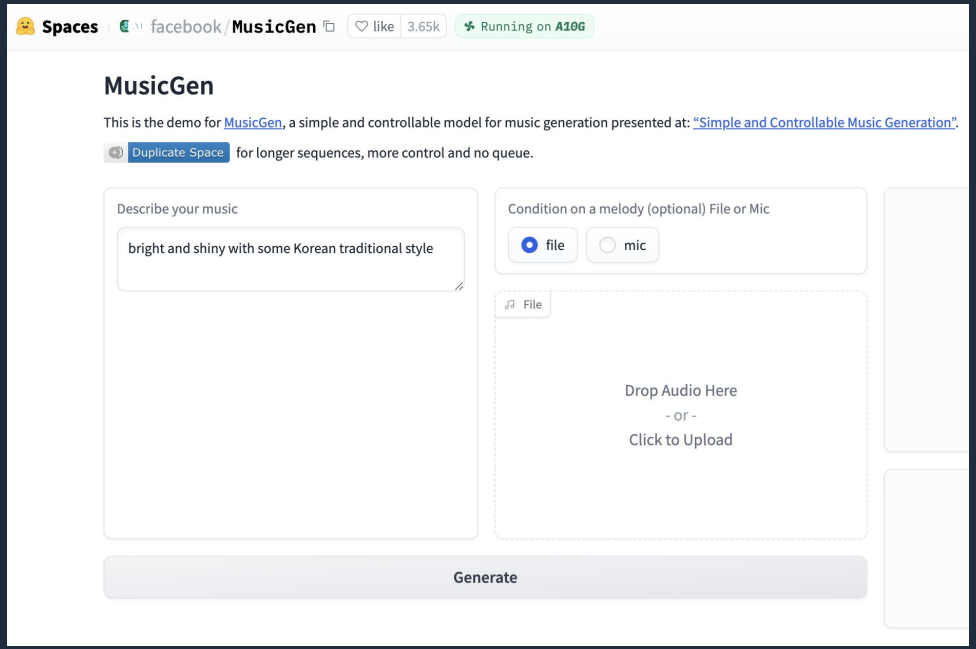
- 이 경우 API 토큰을 생성해야함
- 모든 모델을 지원하지 못함 (text2text-generation과 text-generation만 지원)
  - 이외에도 text2image, text2music 등의 모델이 더 있음





HuggingFace Spaces의 예

- https://huggingface.co/spaces/facebook/MusicGen



HuggingFace Open LLM 리더보드

- 대표 리더보드: Open LLM Leaderboard
  - 모델 성능 리더보드의 종류는 아주 다양 (한국어 버전도 존재)
- Public과 Private 두 종류의 리더보드가 존재

T	Model	Average ⬆️	ARC
💬	moreh/MoMo-72B-lora-1.8.6-DPO	77.29	70.14
🔶	abacusai/Smaug-34B-v0.1	77.29	74.23
🔶	cloudyu/Truthful_DPO_TomGrc_FusionNet_34Bx2_MoE	77.28	72.87
🔶	yunconglong/DARE_TIES_13B	77.1	74.32
🔶	yunconglong/13B_MATH_DPO	77.08	74.66
🔶	TomGrc/FusionNet_34Bx2_MoE	77.07	72.95
🔶	yunconglong/MoE_13B_DPO	77.05	74.32
🔶	cloudyu/4bit_quant_TomGrc_FusionNet_34Bx2_MoE_v0.1_DPO	76.95	73.21
🤝	alchemonaut/QuartetAnemoi-70B-t0.0001	76.86	73.38

programmers

HuggingFace를 활용해 텍스트 분석해보기

HuggingFace에 있는 텍스트 모델을 사용해서 텍스트 분석을 해보자

Few-shot Learning

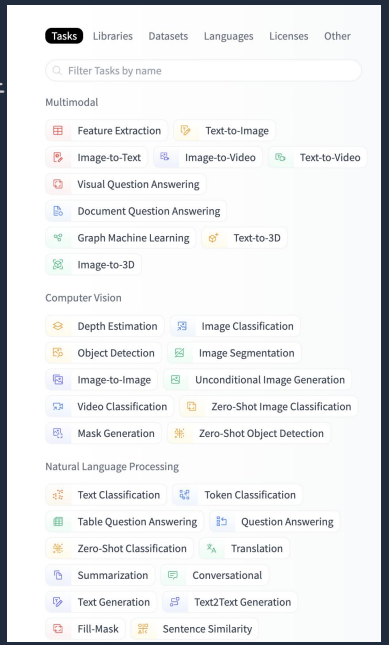
- Zero-Shot, One-Shot, Few-Shot 학습은 NLP 및 비전 분야의 접근 방식
  - 모델이 훈련을 위해 라벨이 지정된 대규모 데이터 세트가 필요 없이 작업을 처리하도록 만들어진 것을 말함
  - 최소한의 예제 만으로 작업을 이해하고 수행하는 모델 생성을 목표로 하는 'Few-shot Learning'이라는 학습 방법에 속함
    - 이는 일종의 Transfer Learning으로 볼 수 있음
  - Zero-Shot: 예제 없이 바로 새로운 태스크 학습에 사용하는 것
  - One-Shot: 예제 하나를 가지고 학습을 하고 사용하는 것
  - Few-Shot: 적은 수의 예제를 갖고 훈련하는 것! 보통 Final layer만 새로 추가하는 형태로 감
- 뒤에서 HuggingFace를 기준으로 Zero-Shot 학습을 이용해볼 예정

Fine-Tuning vs. Transfer Learning

- 차이점
  - Transfer Learning은 한 모델을 활용하여 다른 작업을 시작하는 모든 시나리오를 포괄하는 광범위한 개념
  - Fine-Tuning은 Transfer Learning의 특정 기술로, 사전 학습된 모델을 약간 조정하는 것을 포함
- 적용:
  - Transfer Learning은 사전 학습된 모델을 특징 추출기로 사용한 다음 해당 특징을 기반으로 새로운 분류기를 학습하는 등 다양한 방식으로 적용 가능.
  - Fine-Tuning은 특히 새로운 데이터 세트에 대해 사전 훈련된 모델을 계속 훈련하는 프로세스
- 훈련 깊이:
  - Transfer Learning에서는 모델의 기본 레이어를 완전히 동결하고 네트워크의 일부만 훈련 가능
  - Fine-Tuning에서는 일반적으로 새로운 데이터에 더 잘 맞도록 학습된 표현을 약간 조정하기 위해 더 많은 모델 레이어가 훈련 프로세스에 포함됨

HuggingFace에서 어떤 모델을 사용할지 검색 방법

- HuggingFace에 있는 모델은 대부분 상업적인 용도로 사용하는 것이 가능
  - 라이센스 모델 체크 필요
- https://huggingface.co/models 여기서 검색 혹은 브라우징 수
  - Multimodal
  - Computer Vision
  - Natural Language Processing
  - Audio
  - Tabular
  - Reinforcement Learning



해보려는 작업: 텍스트 감정(sentiment) 분류

- Zero-shot 분류를 해볼 예정
- facebook에서 만든 bart-large-mnli라는 모델을 사용해볼 예정
  - https://huggingface.co/facebook/bart-large-mnli
- transformers 라이브러리를 사용해볼 예정
  - 사용 방식은 분류 레이블을 제공하면서 분류 대상이 되는 텍스트 제공

사용 예:

```
from transformers import pipeline
classifier = pipeline(model="facebook/bart-large-mnli")
classifier("one day I will see the world",
    candidate_labels=['travel', 'cooking', 'dancing', 'exploration'],
    multi_label=True
)
```

HuggingFace 프로그래밍: Zero-Shot Learning 링크

```
from transformers import pipeline

classifier = pipeline(model="facebook/bart-large-mnli")
classifier("I have a problem with my iphone that needs to be resolved asap!",
    candidate_labels=["positive", "neutral", "negative"]
)
```

{'sequence': 'I have a problem with my iphone that needs to be resolved asap!',
 'labels': [**'negative'**, 'neutral', 'positive'],
 'scores': [0.7860444784164429, 0.11748620867729187, 0.09646926820278168]}

programmers

HuggingFace를 활용해 이미지 분석해보기

HuggingFace에 있는 이미지 모델들을 사용해서 이미지 분석을 해보자

해보려는 이미지 작업 두 개

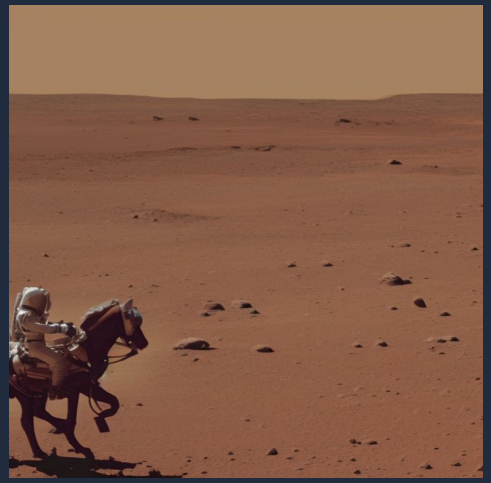
- 개/고양이 이미지 분류 작업
  - openai/clip-vit-base-patch32
  - https://huggingface.co/openai/clip-vit-base-patch32
- Stable Diffusion으로 이미지 생성 작업
  - stabilityai/stable-diffusion-2
  - https://huggingface.co/stabilityai/stable-diffusion-2

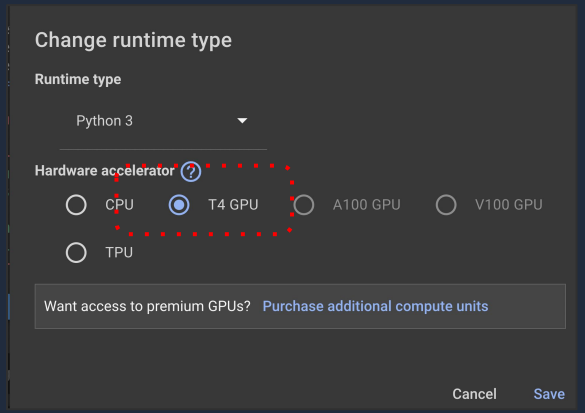
개/고양이 이미지 분류기 만들기

- Hugging Face의 CLIP은 OpenAI에 의해 개발된 모델
  - 이미지와 텍스트 모두 이해하고 이미지에 대한 설명을 텍스트로 제공하고 이를 기반으로 이미지 분류하는 zero-shot learning이 가능
  - 예를 들어, "개"와 "고양이"라는 레이블을 사용하여 이미지가 개인지 고양이인지를 판별 가능
- 구글 Colab 링크

Stable Diffusion 실행해보기

- [stabilityai/stable-diffusion-2 모델 실습]
  - Runtime을 꼭 GPU로 선택할 것!
- 입력 프롬프트: a photo of an astronaut riding a horse on mars
- 출력 이미지





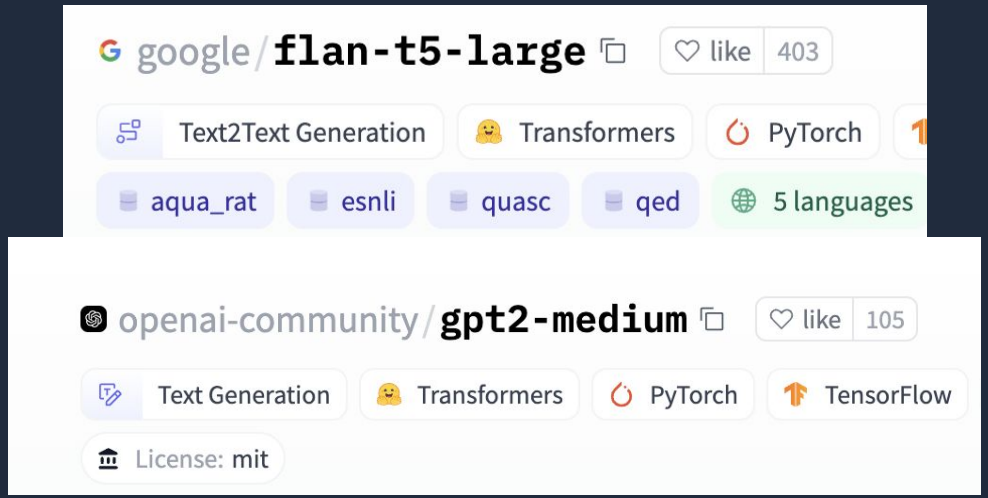
programmers

HuggingFace를 활용해 LLM 사용해보기

HuggingFace에 있는 LLM 모델들을 사용해보자

HuggingFace LLM의 사용 방법

- 하나는 모델을 다운로드 받아서 로컬에서 사용하는 방법
  - 로컬 컴퓨터의 사양에 따라 불가능한 모델도 있음
  - google/flan-t5-large: https://huggingface.co/google/flan-t5-large
- 두번째는 HuggingFace Hub에 있는 모델을 API 형태로 사용하는 방법
  - google/flan-t5-large: https://huggingface.co/google/flan-t5-large



Text Generation vs. Text2Text Generation

- Text Generation
  - Language Model 본연의 동작. 주어진 문장의 다음 단어들 입력
  - Decoder Only: gpt2-medium
- Text2Text Generation
  - ChatGPT 처럼 문장에 대해 답을 주는 방식으로 동작
  - Encoder-Decoder: google/flan-t5-large

HuggingFace Hub 모드로 사용해보기 (API 모드)

```
import os
from langchain import HuggingFaceHub

os.environ['HUGGINGFACEHUB_API_TOKEN'] = ''
llm=HuggingFaceHub(
        repo_id="google/flan-t5-large",
        model_kwargs={
            "temperature":0,
            "max_length":64}
)
print(llm("What is the capital of France?"))
```

HuggingFace 로컬 모드로 사용해보기 #1

- Text-to-Text Generation 모드

```
from transformers import pipeline

model_id = 'google/flan-t5-large'
local_llm = pipeline("text2text-generation", model=model_id, max_length=100)

print(local_llm('What is the capital of Korea? '))
```

HuggingFace 로컬 모드로 사용해보기 #2

- Text Generation 모드

```
model_id = "gpt2-medium"

local_llm = pipeline(
    "text-generation",
    model = model_id,
    max_length=100
)
question = "kpop is"
print(local_llm(question, max_length=30, num_return_sequences=5))
```

HuggingFace LLM 사용해보기

- HuggingFace LLM 사용해보기

보너스: Bark 모델을 사용한 TTS 기능 구현

- 아래 내용을 참고해서 짧은 한국어 문장을 말해주는 코드 구현하기
  - https://huggingface.co/suno/bark-small#suno-usage
- Bark 라이브러리를 이용한 TTS 실습

programmers

숙제

2장 숙제에 대해 알아보자

2장 숙제 내용

- Hugging Face에 계정 만들기
- 오늘 실습 모두 따라해보기
  - 텍스트 분류 Zero-Shot Learning 링크
  - 개/고양이 분류 zero-shot 분류기
  - stabilityai/stable-diffusion-2 모델 실습
  - ◘HuggingFace LLM 사용해보기
  - Bark 라이브러리를 이용한 TTS 실습
- 잘 이해안되는 부분 질문하기

programmers

Q & A

오늘 강의에 대해서 궁금한 부분이 있으면 알려주세요!