

Hugging Face Introduction

Hugging Face의 기본 개념과 활용 이해하기

Hugging Face

Hugging Face란 ?

Hugging Face는 자연어 처리(NLP)와 인공지능(AI)을 위한 플랫폼 및 오픈소스 커뮤니티입니다.

주요 특징:

- 다양한 AI 작업을 위한 사전 학습(Pre-trained)된 모델 제공
- Transformers와 같은 사용하기 쉬운 라이브러리
- NLP, 컴퓨터 비전 등을 지원하는 풍부한 생태계
- 모델 공유
- 미세조정(Fine-tuning) 가능

Hugging Face의 주요 구성 요소

1. Transformers 라이브러리

- 최첨단 사전 학습된 NLP 모델 제공
- 텍스트 생성, 감정 분석, 번역 등의 작업 지원

2. Datasets 라이브러리

- 머신러닝을 위한 수천 개의 데이터셋 제공

3. 모델 허브(Model Hub)

- 커뮤니티에서 공유된 사전 학습된 모델 저장소

4. Inference API

- 복잡한 기술 없이 AI 모델 배포 가능

Hugging Face의 활용 사례

1. 텍스트 분류

- 감정 분석, 스팸 탐지, 주제 분류 등

2. 언어 번역

- 텍스트를 다양한 언어로 번역

3. 텍스트 요약

- 긴 텍스트에서 간결한 요약 생성

4. 질문 답변

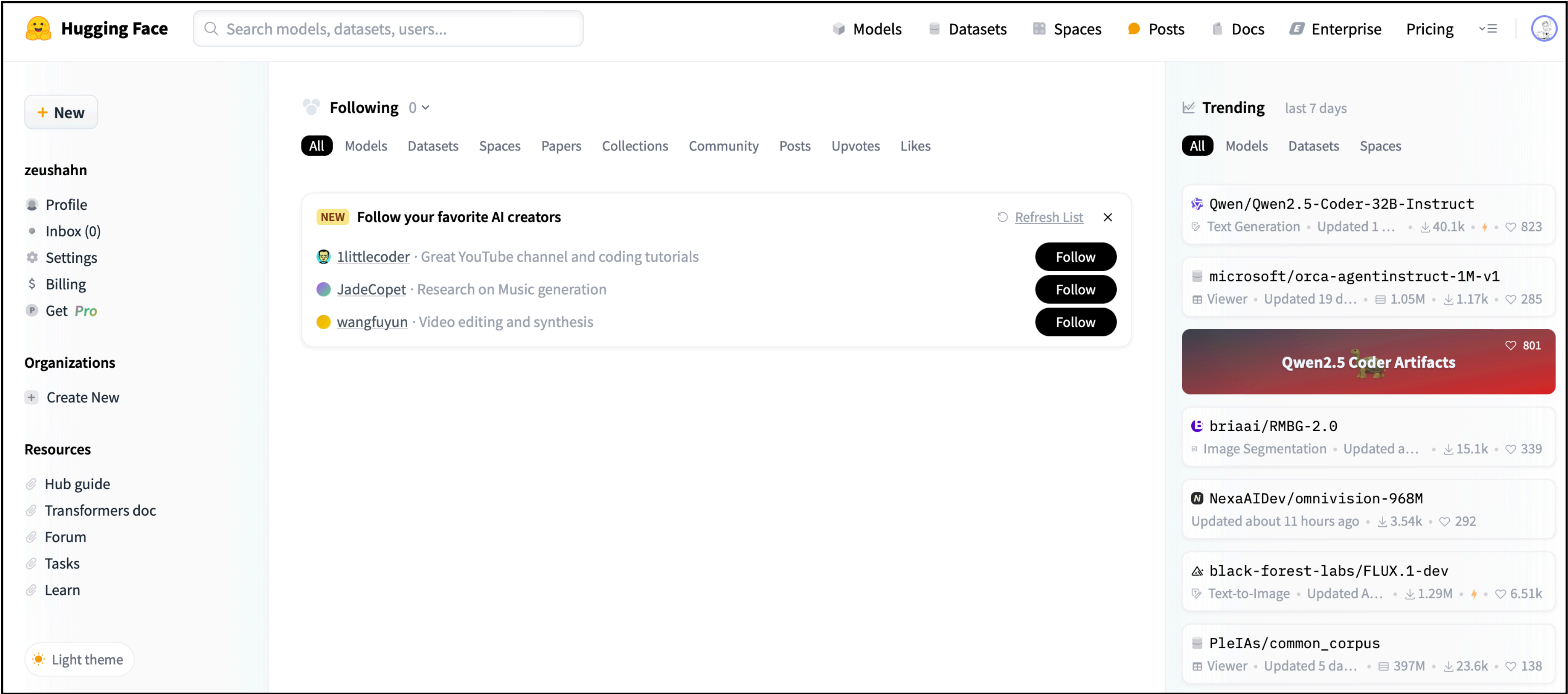
- 주어진 문맥에서 정답 추출

5. 챗봇

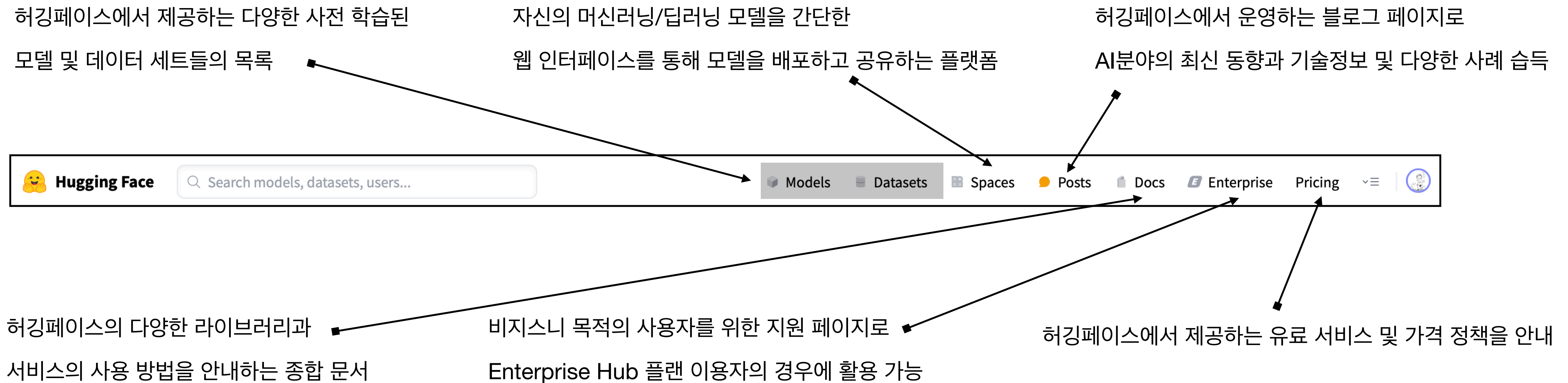
- 대화형 AI 시스템 구축

6. 이미지 분류

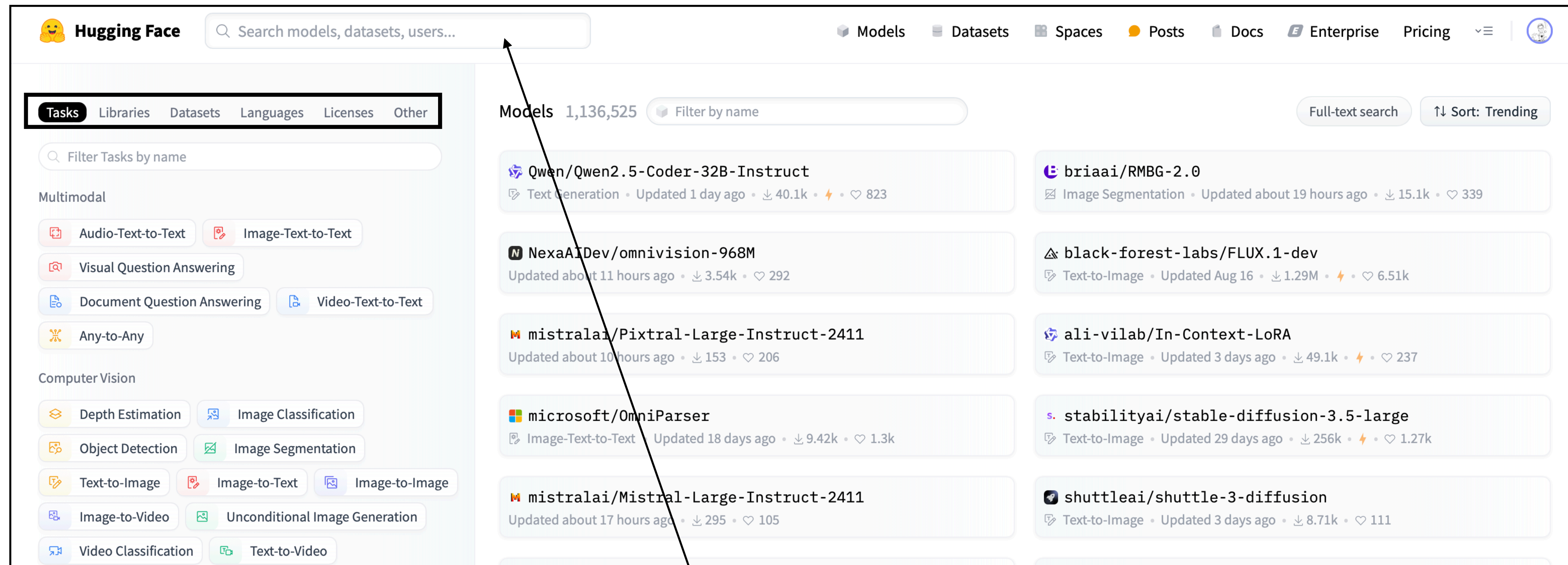
Hugging Face 알아보기



Hugging Face 알아보기



Models Page : 허깅페이스에서 가장 많이 활용되는 주요 기능



resnet-18(합성곱(CNN) 신경망)

resnet-18

이미지 분류

Library

Training Data Set

resnet 모델의 컴퓨터비전은 지원 안함

학습된 모델을 클라우드 API 서비스로 제공 가능

논문의 고유 ID로 <https://arxiv.org/abs/{arXiv ID}>

라이선스

모델의 학습과 배포 방법

모델 다운로드 추세

모델 매개변수 정보

클라우드 API Inference Endpoints를 통해 테스트 가능

microsoft/resnet-18 like 46 Follow Microsoft 5,068

Image Classification Transformers PyTorch TensorFlow Safetensors imagenet-1k resnet vision Inference Endpoints arxiv:1512.03385 License: apache-2.0

Model card Files and versions Community 5

ResNet 모델의 파일과 버전내역

ResNet model trained on imagenet-1k. It was introduced in the paper [Deep Residual Learning for Image Recognition](#) and first released in [this repository](#).

Disclaimer: The team releasing ResNet did not write a model card for this model so this model card has been written by the Hugging Face team.

Model description

ResNet introduced residual connections, they allow to train networks with an unseen number of layers (up to 1000). ResNet won the 2015 ILSVRC & COCO competition, one important milestone in deep computer vision.

Model size 11.7M params Tensor type F32

Inference API Cold

Image Classification

Drag image file here or click to browse from your device

View Code Maximize

모델에 대한 상세한 설명과 예시 코드, 성능지표, 하이퍼파라미터의 정보

허깅페이스 트랜스포머

1. 트랜스포머(Transformers)

: 사전 학습된 트랜스포머 모델 제공

- 자연어 처리 : 텍스트 분류, 개체명 인식, 질의 응답, 언어 모델링, 요약, 번역, 객관식 질의응답, 텍스트 생성.
- 컴퓨터 비전 : 이미지 분류, 객체 탐지, 객체 분할, 이미지 캡셔닝, 얼굴 인식, 포즈 추정.
- 오디오 처리 : 자동 음성 인식, 오디오 분류, 화자 식별, 음성 변환.
- 멀티 모달 : 표 질의 응답, 광학 문자 인식, 스캔한 문서에서 정보 추출, 비디오 분류, 질의 응답.

2. 토크나이저(Tokenizers)

: 텍스트 데이터를 기계가 처리할 수 있는 형태로 변환.

토큰 하는 텍스트를 의미 있는 단위인 토큰(Token)으로 분할 하는 과정.

기존에는 단어위주의 토큰을 생성 하였으나 최근 딥러닝 기반 자연어 처리 모델에서는 더 작은 단위인 하위 단어(Subword) 단위로 토큰화 한다.

- 전처리 작업 : 텍스트 정규화, 특수문자 제거 등의 전처리 작업 수행.
- 토큰화 수행 : BPE(Byte Pair Encoding), Wordpiece등 다양한 알고리즘 지원.
- 패딩 및 잘라내기 : 텍스트 길이 통일을 위한 패딩과 모델 최대 길이 조정.
- 사전 학습된 토크나이저 : 100여개 언어지원 및 사전 학습된 토크나이저를 쉽게 빠르게 적용.

허깅페이스 트랜스포머

3. 데이터 세트(Data Sets)

- : 자연어 처리, 컴퓨터 비전, 오디오 처리등에 사용되는 공개된 수백 개의 데이터세트의 접근 및 처리.
- 전처리 기능 : 데이터 필터링, 토큰화, 패딩, 정규화, 데이터 증강 등 다양한 전처리 기능을 제공.
- 병렬 처리 : 데이터를 전처리시 병렬 처리 지원.
- 스트리밍 처리 : 데이터세트가 너무 큰 경우 스트리밍 방식을 통한 메모리 처리 가능.

4. 디퓨저(Diffusers)

- : 텍스트-이미지 생성과 관련된 딥 모델 지원

5. 가속화(Accelerate)

- : 분산 학습과 하드웨어 가속화를 간소화 해준다.

모델 설정

모델 설정은 딥러닝 모델의 성능과 작동 방식에 직접적인 영향을 미친다.

사전 학습 된 모델을 사용하므로 사전에 정의된 Hyper Parameter를 조정하여 사용 가능 하다.

Model

사전 학습(Pre-training Learning)을 통해 모델이 풍부한 지식을 습득함으로써 상대적으로 적은 양의 데이터로도 우수한 성능을 달성할 수 있다.

사전 훈련된 딥러닝 모델을 새로운 작업이나 데이터셋을 재사용하는 프로세스인 전이학습(Transfer Learning)이 가능하다.

<u>텍스트 분야</u>	<u>이미지 분야</u>	<u>멀티 모달 분야</u>
BERT, RoBERTa, GPT등	ResNet, EfficientNet, ViT	CLIP, ALIGN, Flamingo
텍스 분류, 질의 응답, 번역, 요약, 생성 등	물체인식, 세그멘테이션, 이미지 캡셔닝	이미지-텍스트의 상호 관계

Model

PreTrainedModel 클래스와 ModelMixin 클래스

- 모델 불러오기
- 전이학습
- 모델 저장
- 순전파
- 출력처리

BertForMaskedLM을 이용한 마스크된 단어 예측

Model

특징 추출

이미지나 오디오와 같은 원시 데이터에서 딥러닝 모델의 입력으로 사용될 수 있는 특징(Feature)을 추출하는 역할.

특징이란 데이터에서 유용한 정보를 나타내는 부분이나 속성을 의미한다.

이미지에서는 가장자리(Edge), 질감(Texture), 색상(Color)등의 특징을 추출할 수 있다.

이미지 특징 벡터 활용 예시

- 이미지 분류 : 특징 벡터를 분류 모델의 입력으로 사용해 이미지 분류를 수행.
- 이미지 검색 : 이미지 특징 벡터 간의 유사도를 계산해 유사한 이미지 검색.
- 이미지 캡셔닝 : 이미지 특징 벡터와 텍스트 인코더를 결합해 이미지에 대한 설명(캡션)을 생성.
- 멀티 모달 과제 : 이미지와 텍스트의 특징 벡터를 결합해 멀티모달 과제(시각 질의 응답, 이미지-텍스트 검색 등)를 수행.

Image Processor

이미지 데이터를 딥러닝 모델에 입력하기 전에 필요한 전처리 작업을 수행하는 클래스.

데이터를 모델이 이해할 수 있는 형태로 변환하고 모델 추론에 적합한 크기와 형식으로 조정.

추출 클래스와 마찬가지로 이미지 전처리(크기조절, 중심자르기, 패딩, 정규화)를 수행하며 추가로 회전, 반전등의 데이터 증강기법도 적용 가능.

Image Processor는 다양한 이미지 모델에 대해 최적화된 서브 클래스를 제공한다.

예를 들어 CLIP 모델은 CLIPImageProcessor를, ViLT 모델은 ViltImageProcessor를, YOLOS 모델은 YoloslImageProcessor를 사용한다.

Auto Class

실제 프로젝트에 사용하려면 모델 불러오기, 토큰나이저 설정하기 등 여러가지 전처리 단계가 필요하다.

특히 새로운 모델을 사용하려면 관련 정보를 찾아 코드를 수정해야 하는 번거로움이 발생한다.

AutoClass를 사용하면 모델의 아키텍처 이름만으로 자동으로 설정을 인식하게 작업이 가능하다.

이를 통해 해당 모델의 적합한 설정, 토큰나이저, 모델, 특징 추출기, 이미지 프로세서 등을 쉽게 불러올 수 있다.

장점

- **자동화된 모델 불러오기** : 모델의 이름만으로 아키텍처를 자동으로 인식해 적절한 클래스를 불러온다.
- **간편한 API** : 일관된 API를 제공해 코드를 간소화하고 재사용성을 높인다.
- **모델 교체 용이성** : 동일한 API를 사용하기 때문에 다양한 모델을 쉽게 실험하고 교체가 가능하다.
- **추론 및 미세 조정 간소화** : 모델 설정, 전처리, 후처리 등의 과정이 자동화 되어 추론과 미세 조정 프로세스가 간소화 된다.

Auto Class

주요 Auto Class

- **AutoConfig** : 모델 아키텍처에 맞는 PreTrainedConfig Class 설정.
- **AutoModel** : 모델 아키텍처에 맞는 PreTrainedModel Class 설정.
- **AutoTokenizer** : 모델 아키텍처에 맞는 PreTrainedTokenizer Class 설정.
- **AutoFeatureExtractor** : 이미지 등의 데이터로 부터 특징 벡터를 추출하는 클래스 설정.
- **AutoImageProcessor** : 이미지 전처리를 수행하는 클래스 설정.

Pipeline(파이프라인)

복잡한 머신러닝을 작업을 단순화 하고 사용자 경험을 개선하며, 다양한 구성요소를 하나의 인터페이스로 통합하여 코드를 간결.

개발 생산성과 코드 가독성, 유지 보수성이 좋아진다.

모델, 토큰라이저, 프로세스 등을 내부적으로 불러오고 연결해 작업을 자동화 한다.

즉, 파이프라인 객체를 생성하고 입력 데이터를 전달만 하면 된다.

허깅페이스 파이프 라인

- 자연어 처리 파이프 라인 : 텍스트 생성, 요약, 번역, 질문, 답변 등의 과제를 수행.
- 컴퓨터 비전 파이프 라인 : 이미지 분류, 객체 탐지, 세그멘테이션 등의 과제를 수행.
- 오디오 파이프 라인 : 오디오 분류, 음성 인식등의 과제를 수행.
- 멀티모달 파이프 라인 : 텍스트, 이미지, 오디오 등 다양한 모달리티를 결합한 과제를 수행.