

# Hugging Face Introduction



[Hugging Face – The AI community building the future.](https://huggingface.co/)

# Hugging Face

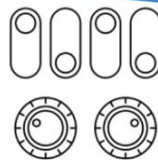
Transformers library

Use it with



Use it for

Applying state of the art  
transformer models



Fine-tuning pretrained  
transformer models

# Hugging Face: Using Transformers

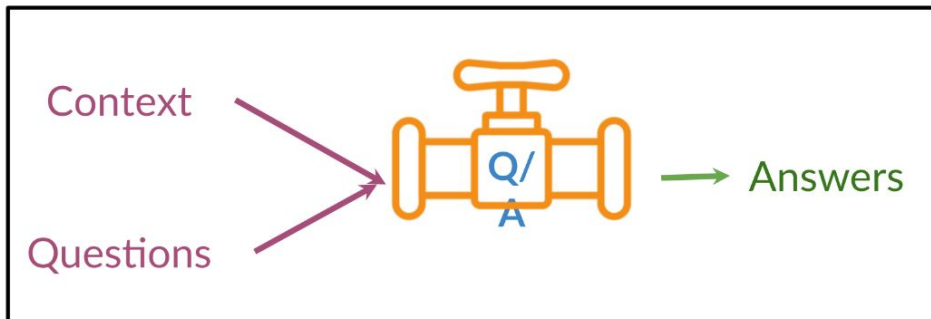
Pipelines



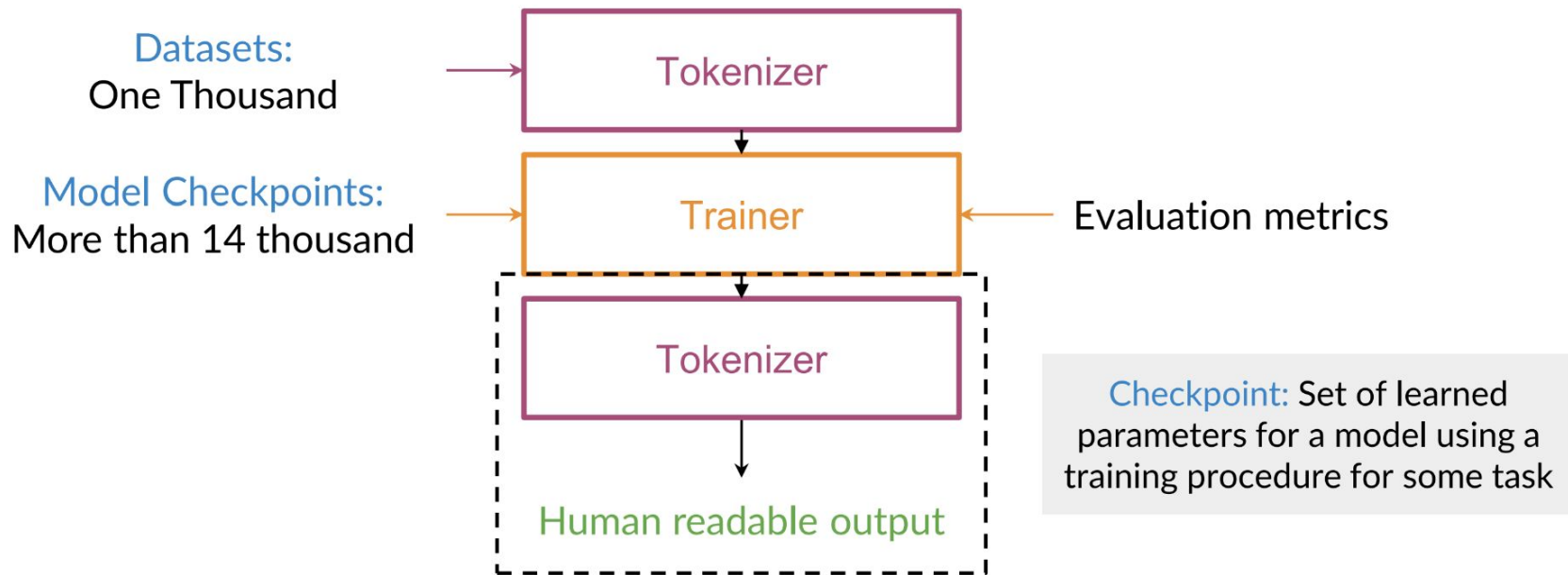
1. Pre-processing your inputs

2. Running the model

3. Post-processing the outputs



# Hugging Face: Fine-Tuning Transformers



# Tasks

Pipelines



Initialization

Task

Model  
Checkpoint

Use

Inputs for  
the task

Sentiment Analysis

Sequence

Question Answering

Context and  
questions

Fill-Mask

Sentence and  
position

# Checkpoints



Huge number of model checkpoints that you can use in your pipelines.

But **beware**, not every checkpoint would be suitable for your task.

# Model Checkpoints

## Model Checkpoints:

More than 15 thousand  
(and increasing)

Upload the architecture  
and weights with 1 line  
of code!

Model	Dataset	Name in 🤗
DistilBERT	Stanford Question Answering Dataset (SQuAD)	distilbert-base-cased-distilled-squad
BERT	Wikipedia and Book Corpus	bert-base-cased
...	...	...

# Model Card and Datasets

Tasks **Libraries** Datasets Languages Licenses Other

Filter Libraries by name

PyTorch TensorFlow JAX Transformers

TensorBoard Diffusers Safetensors

Stable-Baselines3 PEFT ONNX ML-Agents

Sentence Transformers Flair Timm Sample Factory

Keras Adapter Transformers spaCy ESPnet

Transformers.js fastai Core ML NeMo

Rust Joblib Scikit-learn fastText speechbrain

PaddlePaddle OpenCLIP BERTopic Fairseq

OpenVINO Graphcore TF Lite Stanza Asteroid

PaddleNLP allenNLP SpanMarker Habana Pythae

pyannote.audio

Tasks Libraries **Datasets** Languages Licenses Other

Filter Tasks by name

Multimodal

Feature Extraction Text-to-Image Image-to-Text

Text-to-Video Visual Question Answering

Document Question Answering Graph Machine Learning

Computer Vision

Depth Estimation Image Classification

Object Detection Image Segmentation

Image-to-Image Unconditional Image Generation

Video Classification Zero-Shot Image Classification

Natural Language Processing

Text Classification Token Classification

Table Question Answering Question Answering

Zero-Shot Classification Translation

Summarization Conversational

Text Generation Text2Text Generation Fill-Mask

Sentence Similarity

Audio

Text-to-Speech Text-to-Audio

Automatic Speech Recognition Audio-to-Audio

Audio Classification Voice Activity Detection

Tabular

Tabular Classification Tabular Regression

Reinforcement Learning

Reinforcement Learning Robotics

Tasks Libraries **Datasets** Languages Licenses Other

Filter Datasets by name

glue squad mozilla-foundation/common\_voice\_7\_0

imdb imagenet-1k wikipedia xtreme

common\_voice mozilla-foundation/common\_voice\_11\_0

conll2003 bookcorpus masyas/gtzan samsum

squad\_v2 fka/awesome-chatgpt-prompts clinc\_oos

c4 Open-Orca/OpenOrca super\_glue

cnn\_dailymail OpenAssistant/oasst1

facebook/voxpopuli librispeech\_asr

huggan/smithsonian\_butterflies\_subset billsum

PolyAI/minds14 beans universal\_dependencies

oscar google/fleurs garage-bAInd/Open-Platypus

wmt16 mozilla-foundation/common\_voice\_8\_0

tweet\_eval databricks/databricks-dolly-15k mc4

ehartford/dolphin mozilla-foundation/common\_voice\_13\_0

kde4 amazon\_reviews\_multi tatsu-lab/alpaca

togethercomputer/RedPajama-Data-1T sst2 cc100

tiuae/falcon-refinedweb multi\_nli wnut\_17

klue relbert/semeval2012\_relational\_similarity

jondurbin/airoboros-3.1 jondurbin/airoboros-2.2.1

food101 superb snli opus\_books

scene\_parse\_150 ehartford/wizard\_vicuna\_70k\_unfiltered

gsm8k bigcode/starcoderdata xnli eli5

wikiann financial\_phrasebank Anthropic/hh-rlhf

cerebras/SlimPajama-627B cifar10 esb/datasets

cppe-5

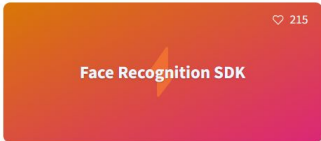


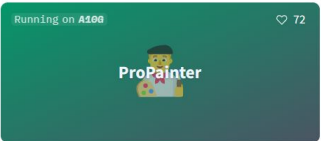



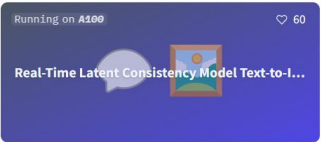


# Hugging Spaces

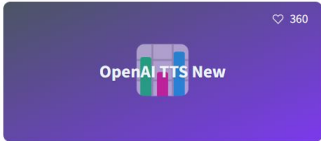




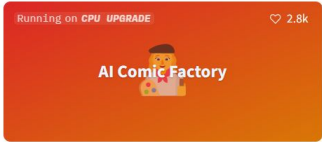

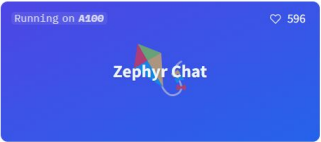
Discover amazing ML apps made by the community!

[Create new Space](#)

☆ Spaces of the week 🔥

 <p>Face Recognition SDK</p> <p>FaceOnLive 2 days ago 215</p>	 <p>ID Document Recognition SDK</p> <p>FaceOnLive 2 days ago 170</p>	 <p>Distil Whisper Web</p> <p>Xenova 7 days ago 70</p>	 <p>ProPainter</p> <p>Running on A100 2 days ago 72</p>
 <p>Chat with DeepSeek Coders 33B</p> <p>deepseek-ai 6 days ago 70</p>	 <p>Chat with DeepSeek Coders 7B</p> <p>deepseek-ai 4 days ago 35</p>	 <p>Whisper vs Distil-Whisper</p> <p>distil-whisper 4 days ago 38</p>	 <p>Real-Time Latent Consistency Model Text-to-Image</p> <p>radames 2 days ago 60</p>

# All running apps, trending first

 <p>OpenAI TTS New</p> <p>yshazma 2 days ago 360</p>	 <p>Face Recognition SDK</p> <p>FaceOnLive 2 days ago 215</p>	 <p>ID Document Recognition SDK</p> <p>FaceOnLive 2 days ago 170</p>	 <p>Open LLM Leaderboard</p> <p>HuggingFaceH4 about 19 hours ago 6.08k</p>
 <p>Face Liveness Detection SDK</p> <p>FaceOnLive 2 days ago 154</p>	 <p>AI Comic Factory</p> <p>jbilcke-hf 2 days ago 2.8k</p>	 <p>Fast Stable Diffusion</p> <p>prodia 16 days ago 541</p>	 <p>Zephyr Chat</p> <p>Running on A100 14 days ago 596</p>

# Hugging Spaces APIs

## Pipelines for inference

The `pipeline()` makes it simple to use any model from the [Hub](#) for inference on any language, computer vision, speech, and multimodal tasks. Even if you don't have experience with a specific modality or aren't familiar with the underlying code behind the models, you can still use them for inference with the `pipeline()`! This tutorial will teach you to:

- Use a `pipeline()` for inference.
- Use a specific tokenizer or model.
- Use a `pipeline()` for audio, vision, and multimodal tasks.

```
from transformers import pipeline

transcriber = pipeline(task= "automatic-speech-recognition" )

transcriber( "https://huggingface.co/datasets/Narsil/asr_dummy/resolve/main/mlk.flac" )
```

Output:

```
{'text': 'I HAVE A DREAM BUT ONE DAY THIS NATION WILL RISE UP LIVE UP THE TRUE MEANING OF ITS TREES' }
```

## Load pretrained instances with an AutoClass

With so many different Transformer architectures, it can be challenging to create one for your checkpoint. As a part of 🤗 Transformers core philosophy to make the library easy, simple and flexible to use, an AutoClass automatically infers and loads the correct architecture from a given checkpoint. The `from_pretrained()` method lets you quickly load a pretrained model for any architecture so you don't have to devote time and resources to train a model from scratch. Producing this type of checkpoint-agnostic code means if your code works for one checkpoint, it will work with another checkpoint - as long as it was trained for a similar task - even if the architecture is different.

```
from transformers import AutoTokenizer
tokenizer = AutoTokenizer.from_pretrained( "bert-base-uncased" )
```

```
sequence = "In a hole in the ground there lived a hobbit."
print (tokenizer(sequence))
```

Output:

```
{'input_ids': [101, 1999, 1037, 4920, 1999, 1996, 2598, 2045, 2973, 1037, 7570,
10322, 4183, 1012, 102],
'token_type_ids': [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
'attention_mask': [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]}
```

Load pretrained instances with an Auto  
Class

AutoTokenizer

AutoImageProcessor

AutoFeatureExtractor

AutoProcessor

AutoModel

```
from transformers import AutoImageProcessor
```

```
image_processor = AutoImageProcessor.from_pretrained( "google/vit-base-patch16-224" )
```

```
from transformers import AutoFeatureExtractor
```

```
feature_extractor = AutoFeatureExtractor.from_pretrained(  
...     "ehcalabres/wav2vec2-lg-xlsr-en-speech-emotion-recognition" )
```

```
from transformers import AutoProcessor
```

```
processor = AutoProcessor.from_pretrained( "microsoft/layoutlmv2-base-uncased" )
```

```
from transformers import AutoModelForSequenceClassification
```

```
model = AutoModelForSequenceClassification.from_pretrained( "distilbert-base-uncased" )
```

```
from transformers import AutoModelForTokenClassification
```

```
model = AutoModelForTokenClassification.from_pretrained( "distilbert-base-uncased" )
```

