Hugging Face:
A state-of-the art data science platform for ML and AI practitioners

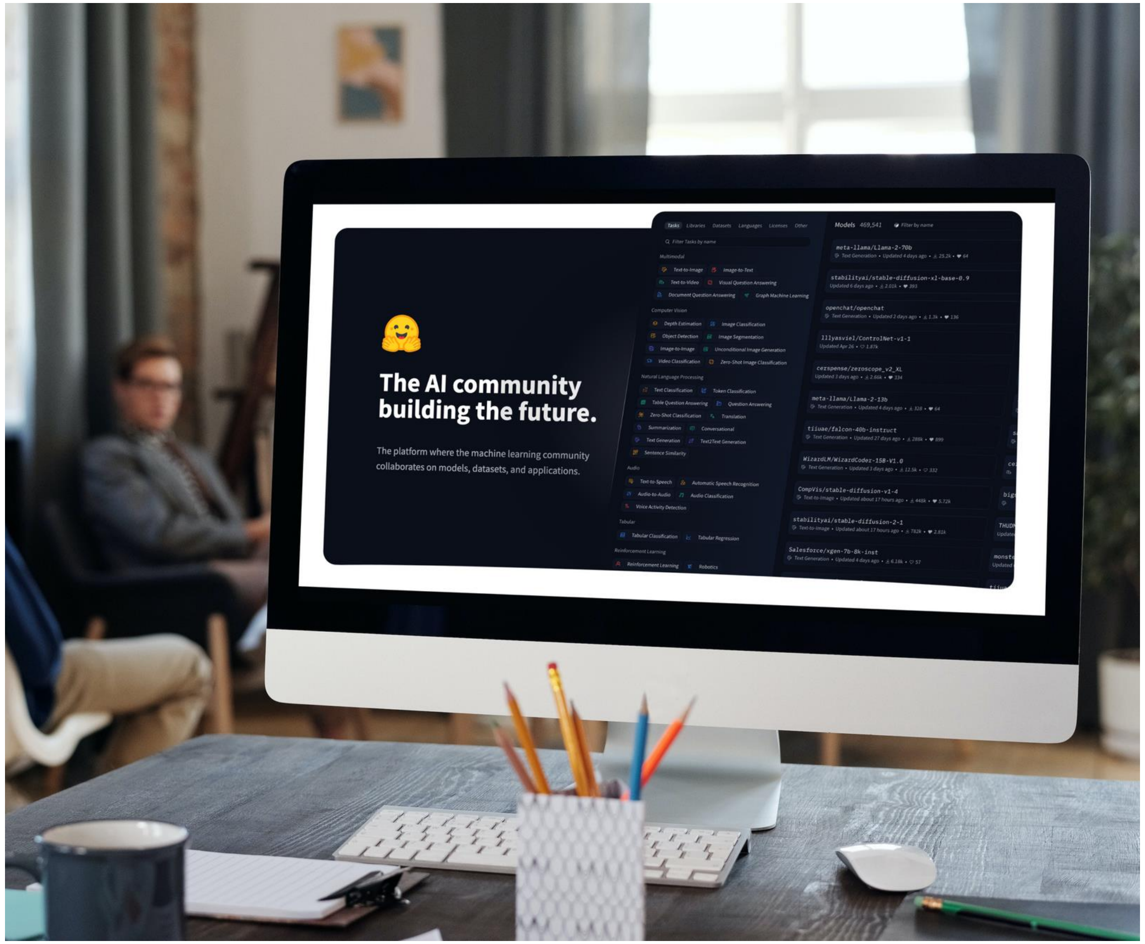
By Martial Luyts

KU LEUVEN

LEUVEN STATISTICS
RESEARCH CENTRE

What is Hugging Face?

- Platform that is transforming the field of ML and AI through open source and open science
- It offers thousands of ML models, datasets and demo apps.
- Likewise to GitHub, collaboration with other ML experts is possible
- Link: **huggingface.co**


The AI community building the future.
The platform where the machine learning community collaborates on models, datasets, and applications.

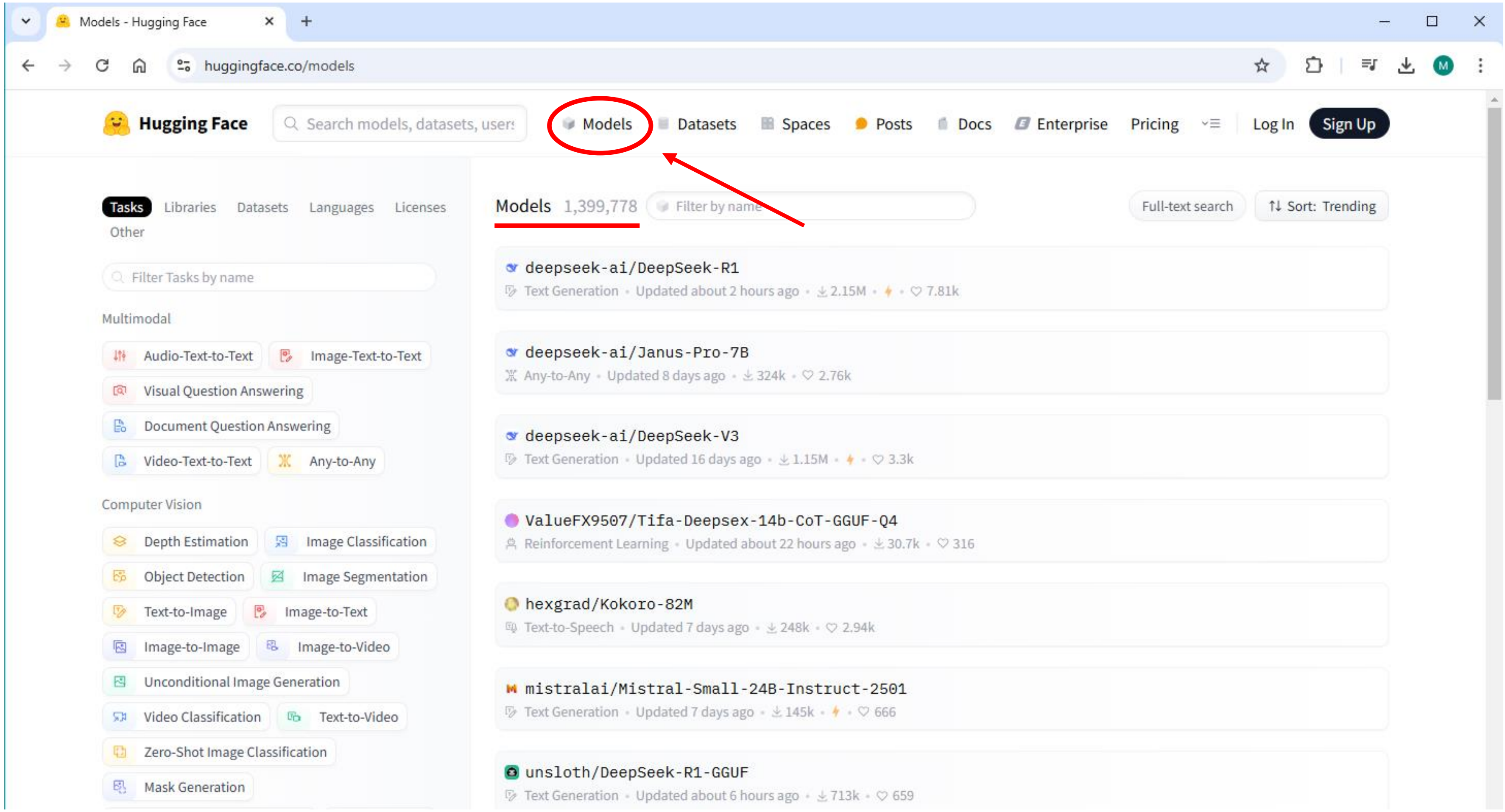
Why is this useful for our datathon?

- Leverage state-of-the-art open-source LMMs that are developed by ML experts from all over the world
- On Hugging Face,
  - > 300000 open-source pre-trained models &
  - > 100000 datasets

are available

- It offers webpages (Spaces) where you can test several models easily

Exploring the platform



Models 1,399,778

deepseek-ai/DeepSeek-R1
Text Generation • Updated about 2 hours ago • 2.15M • • 7.81k

deepseek-ai/Janus-Pro-7B
Any-to-Any • Updated 8 days ago • 324k • 2.76k

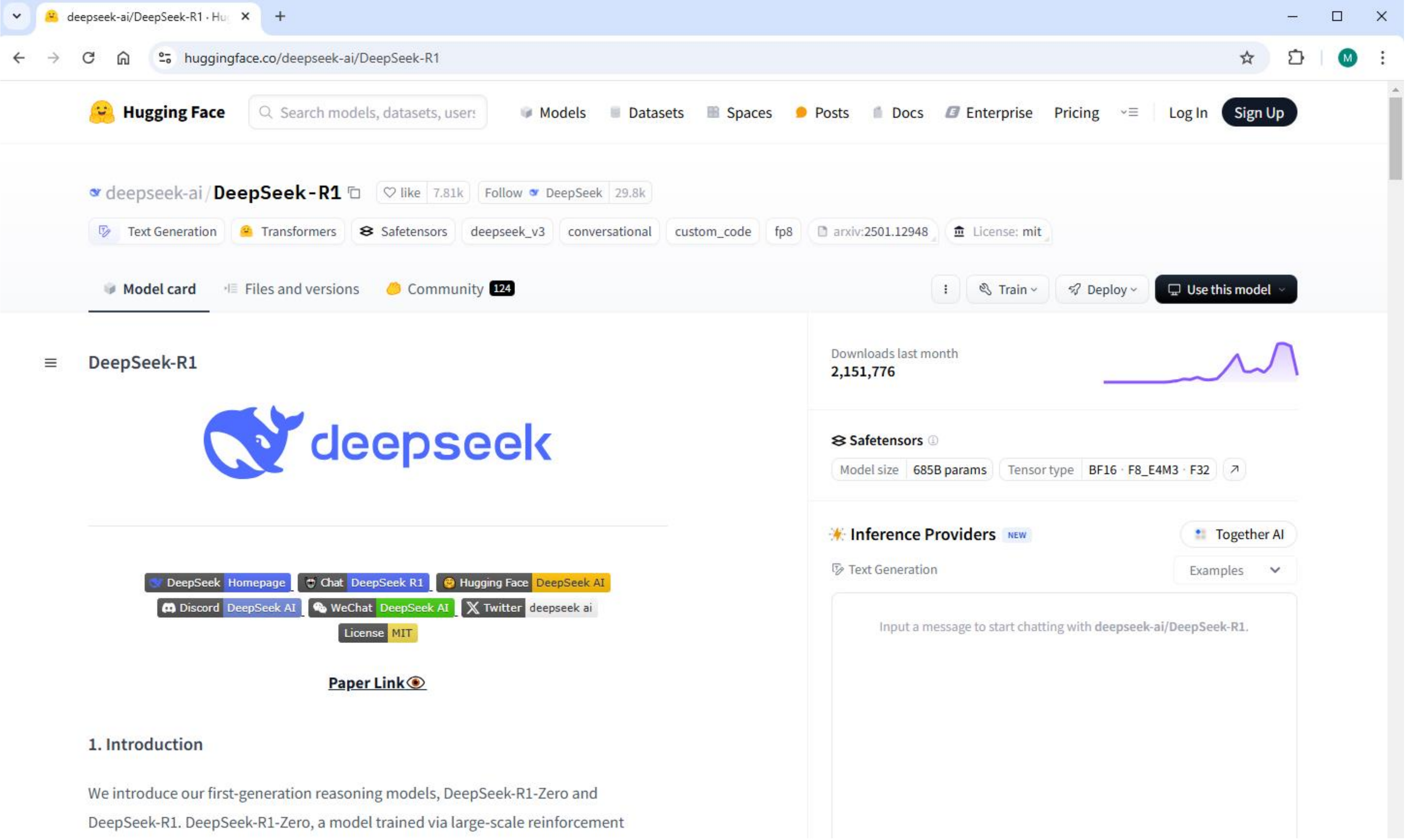
deepseek-ai/DeepSeek-V3
Text Generation • Updated 16 days ago • 1.15M • • 3.3k

ValueFX9507/Tifa-Deepsex-14b-CoT-GGUF-Q4
Reinforcement Learning • Updated about 22 hours ago • 30.7k • 316

hexgrad/Kokoro-82M
Text-to-Speech • Updated 7 days ago • 248k • 2.94k

mistralai/Mistral-Small-24B-Instruct-2501
Text Generation • Updated 7 days ago • 145k • • 666

unsloth/DeepSeek-R1-GGUF
Text Generation • Updated about 6 hours ago • 713k • 659



deepseek-ai / DeepSeek-R1

Downloads last month
2,151,776

Safetensors

Model size 685B params · Tensor type BF16 · F8_E4M3 · F32

Input a message to start chatting with **deepseek-ai/DeepSeek-R1**.

DeepSeek-R1

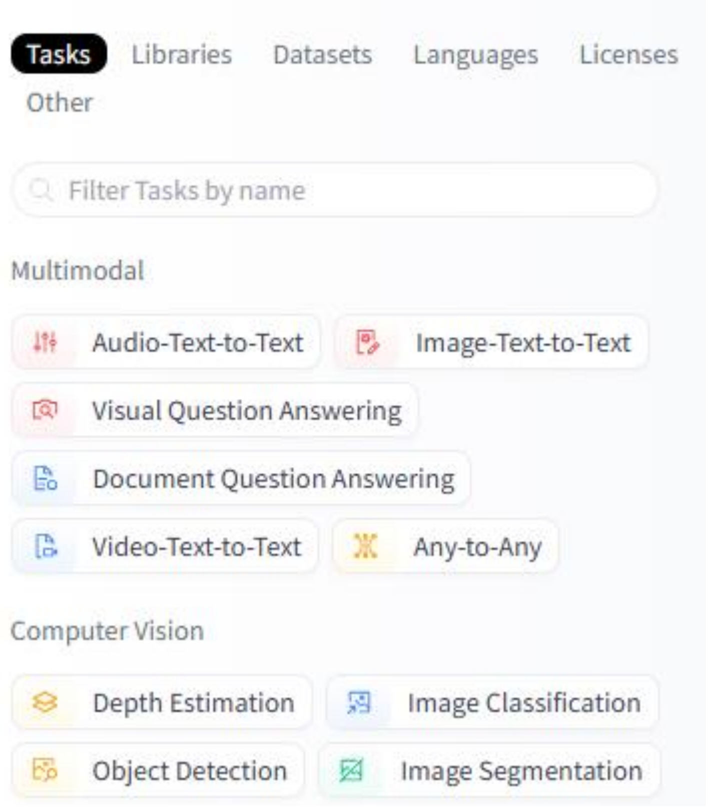
Paper Link👁️

1. Introduction

We introduce our first-generation reasoning models, DeepSeek-R1-Zero and DeepSeek-R1. DeepSeek-R1-Zero, a model trained via large-scale reinforcement

Different applications

- Every model is designed for one (or more) specific task(s):
  - Question Answering
  - Zero-Shot Classification
  - Translation
  - Summarization
  - Text Generation
  - …



Tasks Libraries Datasets Languages Licenses Other

Filter Tasks by name

Multimodal

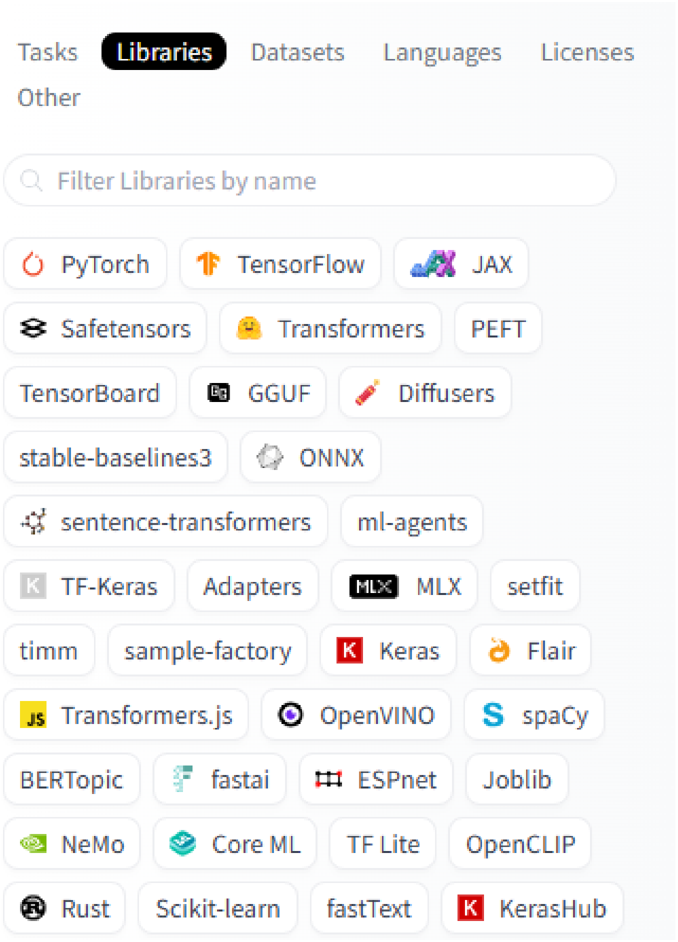
- Audio-Text-to-Text
- Image-Text-to-Text
- Visual Question Answering
- Document Question Answering
- Video-Text-to-Text
- Any-to-Any

Computer Vision

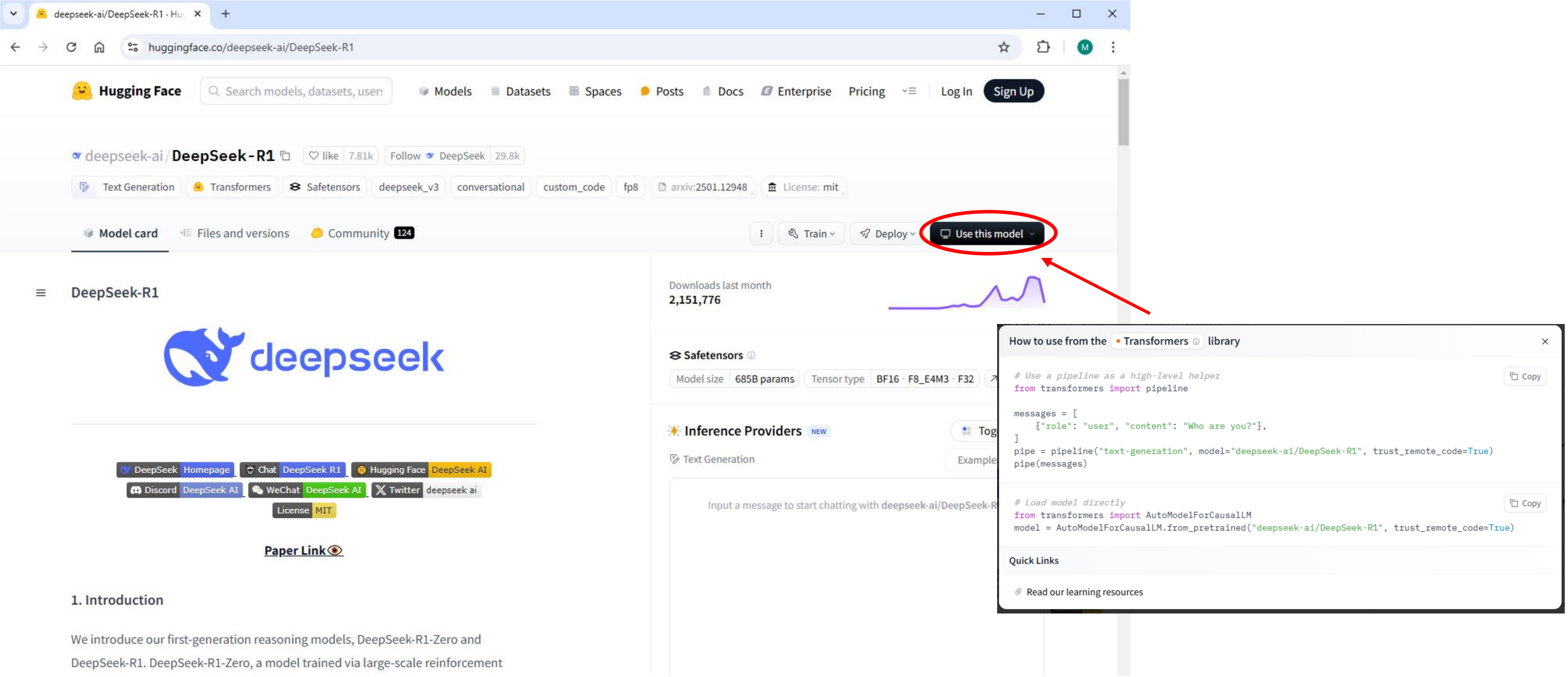
- Depth Estimation
- Image Classification
- Object Detection
- Image Segmentation

Accessing these models

- Throughout libraries within Python



- More easily, you can have a look at sample code



1. Introduction

We introduce our first-generation reasoning models, DeepSeek-R1-Zero and DeepSeek-R1. DeepSeek-R1-Zero, a model trained via large-scale reinforcement

How to use from the Transformers library

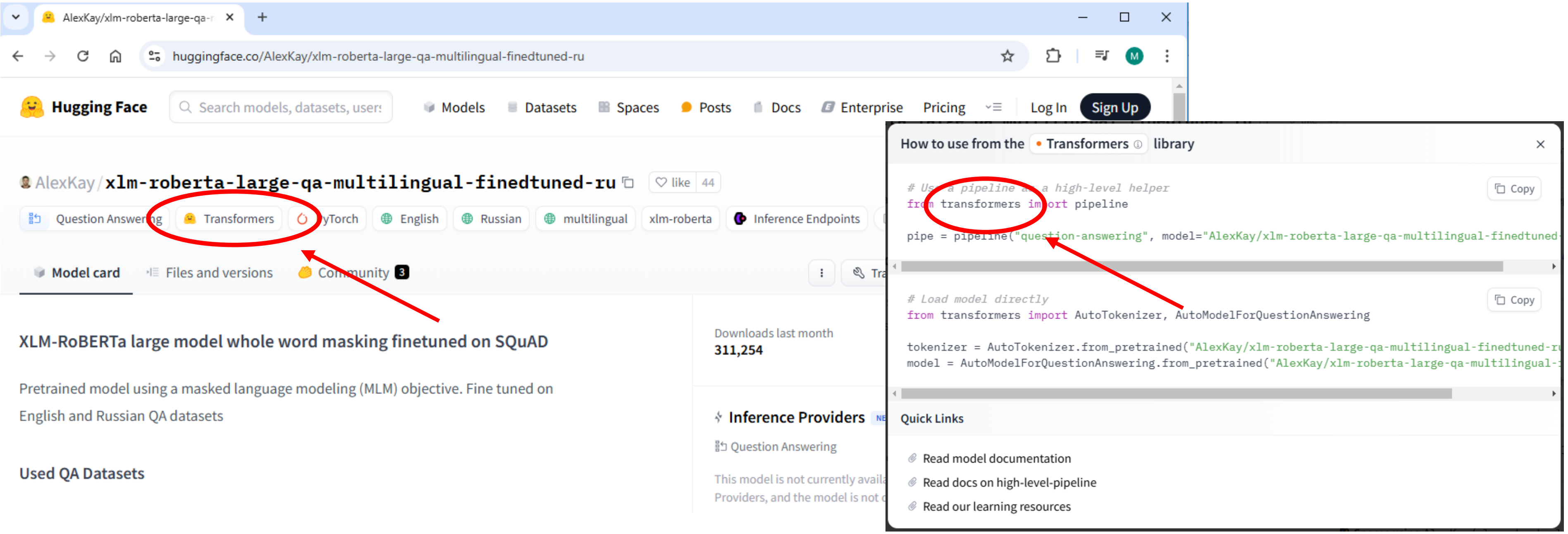
```
# Use a pipeline as a high-level helper
from transformers import pipeline

messages = [
    {"role": "user", "content": "Who are you?"},
]
pipe = pipeline("text-generation", model="deepseek-ai/DeepSeek-R1", trust_remote_code=True)
pipe(messages)
```

```
# Load model directly
from transformers import AutoModelForCausalLM
model = AutoModelForCausalLM.from_pretrained("deepseek-ai/DeepSeek-R1", trust_remote_code=True)
```

Example: The Transformers package

- A useful Python package is **Transformers**, maintained by Hugging Face and the community
- Provides many pretrained LLM models, originating from Hugging Face



XLM-RoBERTa large model whole word masking finetuned on SQuAD

Pretrained model using a masked language modeling (MLM) objective. Fine tuned on English and Russian QA datasets

Used QA Datasets

Downloads last month
311,254

Inference Providers

Question Answering

This model is not currently availa
Providers, and the model is not c

How to use from the Transformers library

```
# Use a pipeline as a high-level helper
from transformers import pipeline

pipe = pipeline("question-answering", model="AlexKay/xlm-roberta-large-qa-multilingual-finedtuned-
```

```
# Load model directly
from transformers import AutoTokenizer, AutoModelForQuestionAnswering

tokenizer = AutoTokenizer.from_pretrained("AlexKay/xlm-roberta-large-qa-multilingual-finedtuned-ru
model = AutoModelForQuestionAnswering.from_pretrained("AlexKay/xlm-roberta-large-qa-multilingual-f
```

Quick Links

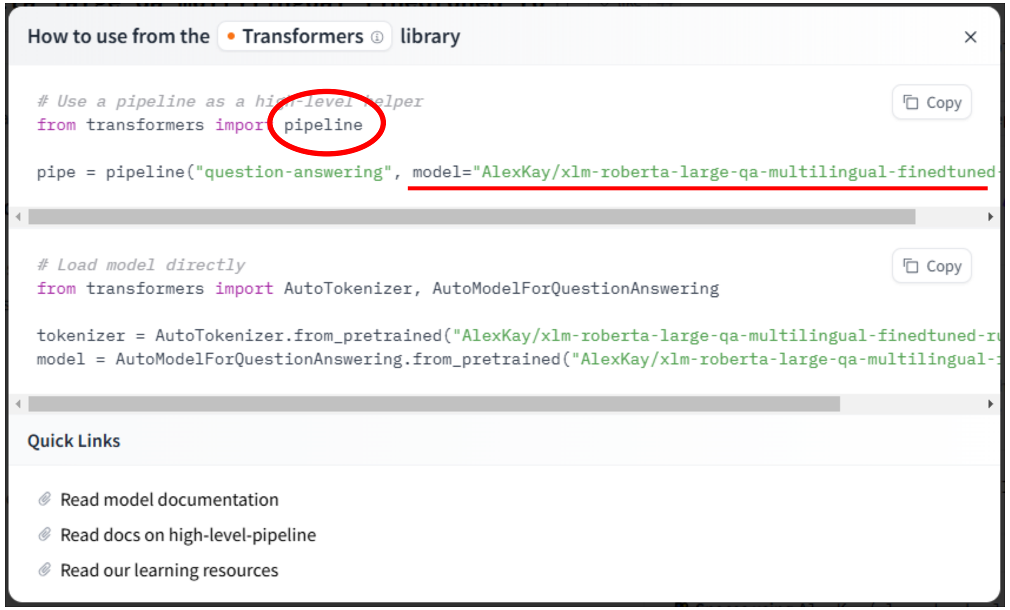
- Read model documentation
- Read docs on high-level-pipeline
- Read our learning resources

The pipeline function

- API of the Transformers library
- Groups all the steps needed to go from raw text to usable predictions.
- The core of a pipeline:

  → The model used

- Advantage: Includes all necessary preprocessing as well as some postprocessing steps related to the chosen model



How to use from the Transformers library

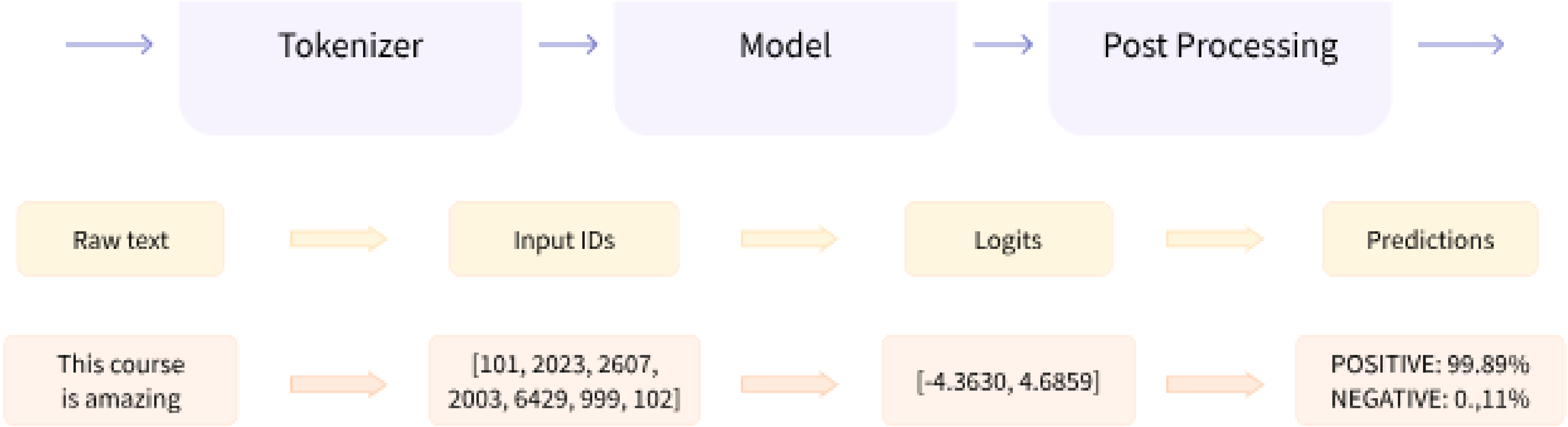
```
# Use a pipeline as a high-level helper
from transformers import pipeline

pipe = pipeline("question-answering", model="AlexKay/xlm-roberta-large-qa-multilingual-finedtuned
```

```
# Load model directly
from transformers import AutoTokenizer, AutoModelForQuestionAnswering

tokenizer = AutoTokenizer.from_pretrained("AlexKay/xlm-roberta-large-qa-multilingual-finedtuned-ru
model = AutoModelForQuestionAnswering.from_pretrained("AlexKay/xlm-roberta-large-qa-multilingual-
```

The pipeline function



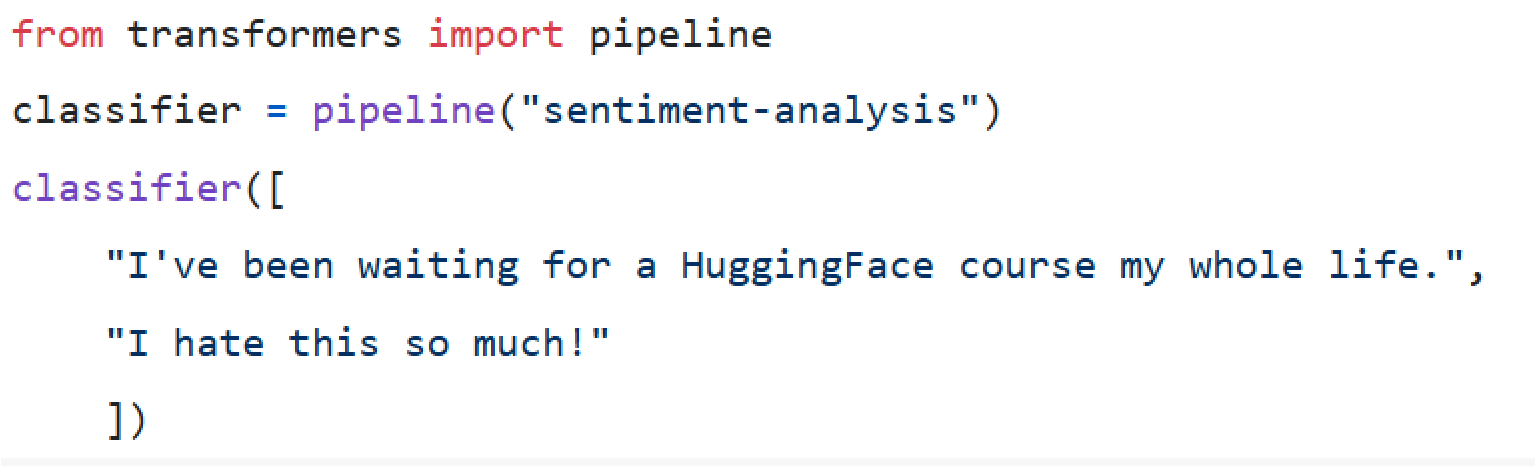
Tokenizer → Model → Post Processing

Raw text → Input IDs → Logits → Predictions

This course is amazing → [101, 2023, 2607, 2003, 6429, 999, 102] → [-4.3630, 4.6859] → POSITIVE: 99.89%
NEGATIVE: 0.,11%

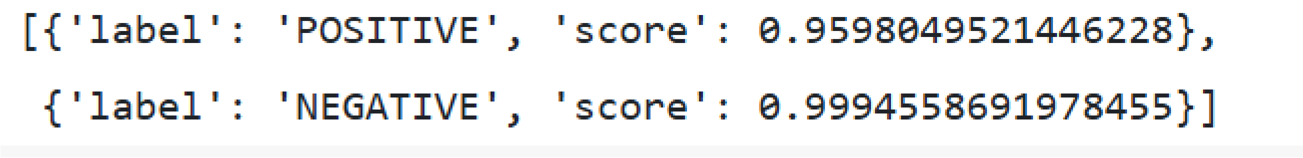
Text classification:

- Code:



```
from transformers import pipeline
classifier = pipeline("sentiment-analysis")
classifier([
    "I've been waiting for a HuggingFace course my whole life.",
    "I hate this so much!"
    ])
```

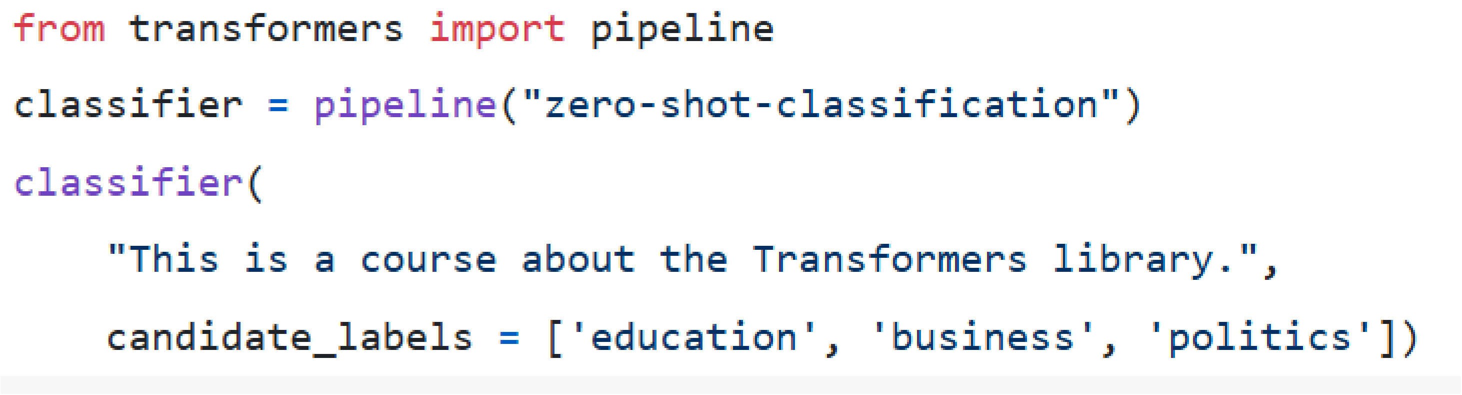
- Output:



```
[{'label': 'POSITIVE', 'score': 0.9598049521446228},
 {'label': 'NEGATIVE', 'score': 0.9994558691978455}]
```

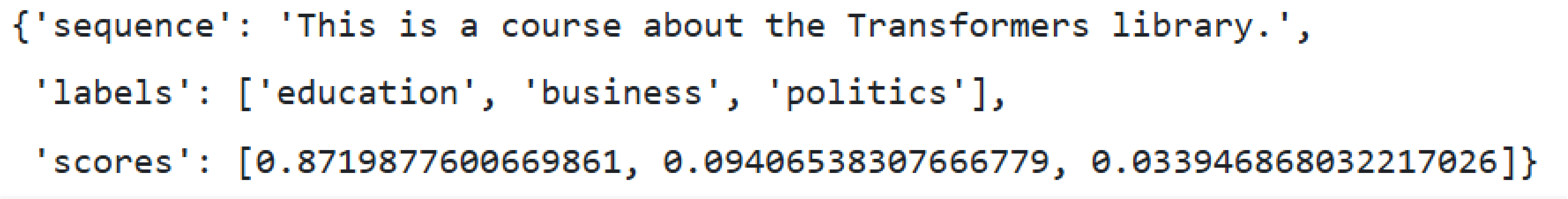
Zero-shot classification:

- Code:



```
from transformers import pipeline
classifier = pipeline("zero-shot-classification")
classifier(
    "This is a course about the Transformers library.",
    candidate_labels = ['education', 'business', 'politics'])
```

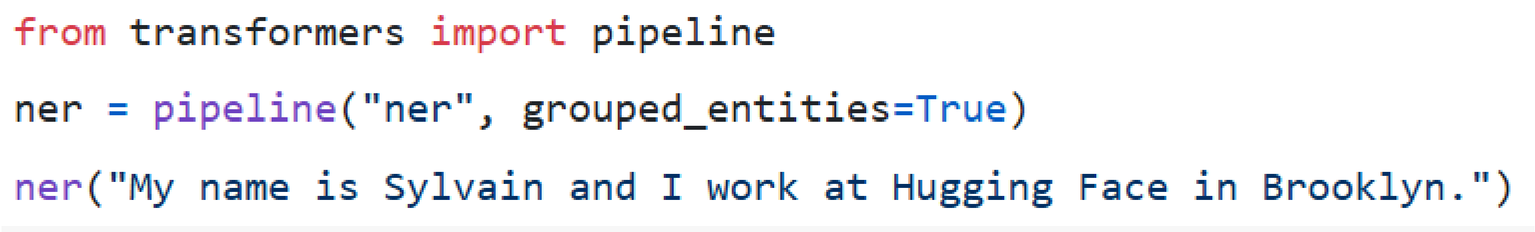
- Output:



```
{'sequence': 'This is a course about the Transformers library.',
 'labels': ['education', 'business', 'politics'],
 'scores': [0.8719877600669861, 0.09406538307666779, 0.03394686803217026]}
```

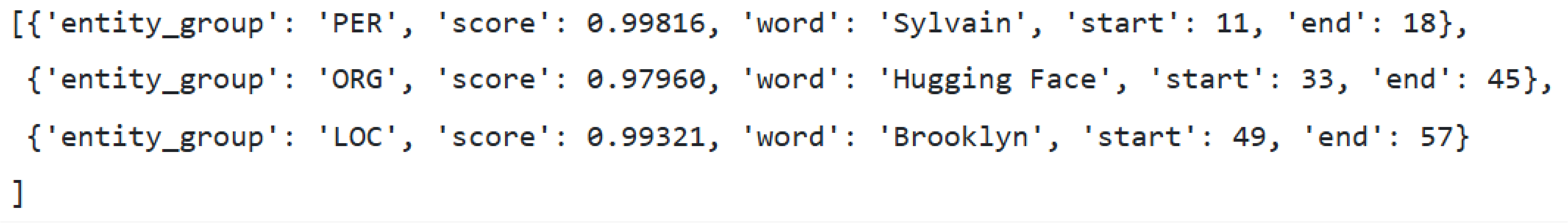
Named entity recognition:

- Code:



```python
from transformers import pipeline
ner = pipeline("ner", grouped_entities=True)
ner("My name is Sylvain and I work at Hugging Face in Brooklyn.")
```

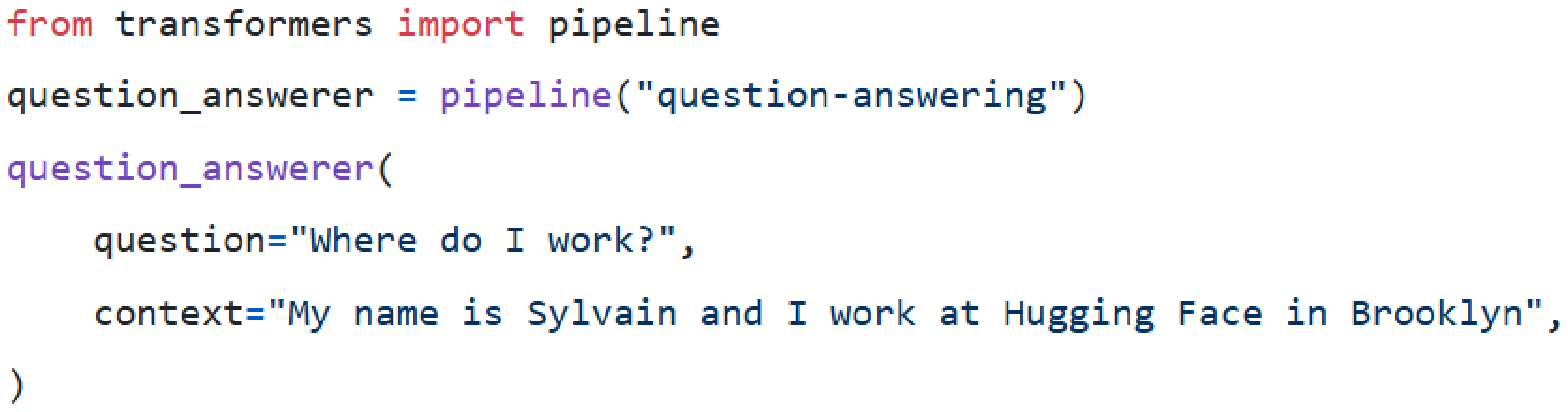
- Output:



```
[{'entity_group': 'PER', 'score': 0.99816, 'word': 'Sylvain', 'start': 11, 'end': 18},
 {'entity_group': 'ORG', 'score': 0.97960, 'word': 'Hugging Face', 'start': 33, 'end': 45},
 {'entity_group': 'LOC', 'score': 0.99321, 'word': 'Brooklyn', 'start': 49, 'end': 57}
]
```

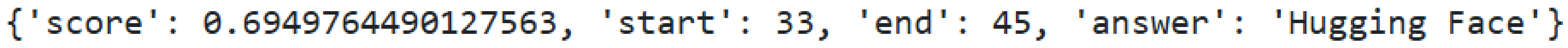
Question answering:

- Code:



```python
from transformers import pipeline
question_answerer = pipeline("question-answering")
question_answerer(
    question="Where do I work?",
    context="My name is Sylvain and I work at Hugging Face in Brooklyn",
)
```

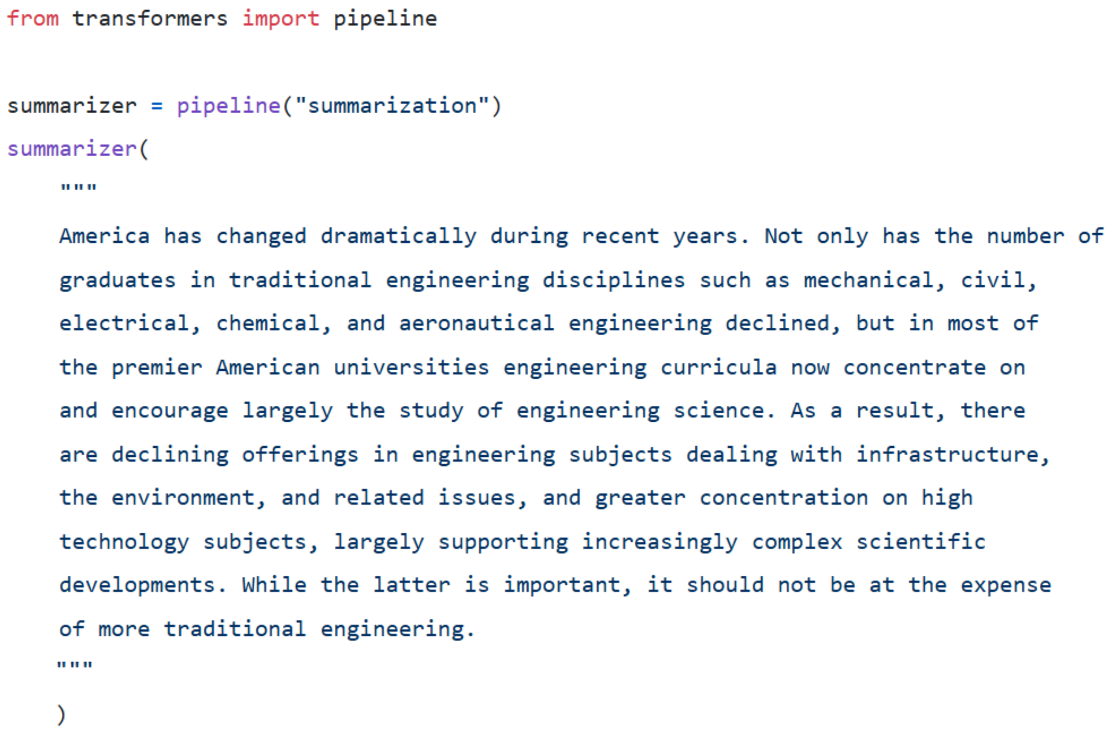
- Output:



```
{'score': 0.6949764490127563, 'start': 33, 'end': 45, 'answer': 'Hugging Face'}
```

Summarization:

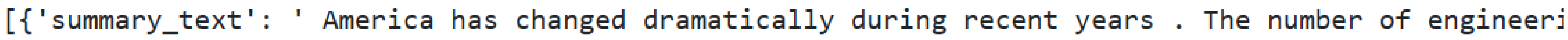
- Code:



```python
from transformers import pipeline

summarizer = pipeline("summarization")
summarizer(
    """
    America has changed dramatically during recent years. Not only has the number of
    graduates in traditional engineering disciplines such as mechanical, civil,
    electrical, chemical, and aeronautical engineering declined, but in most of
    the premier American universities engineering curricula now concentrate on
    and encourage largely the study of engineering science. As a result, there
    are declining offerings in engineering subjects dealing with infrastructure,
    the environment, and related issues, and greater concentration on high
    technology subjects, largely supporting increasingly complex scientific
    developments. While the latter is important, it should not be at the expense
    of more traditional engineering.
    """
    )
```

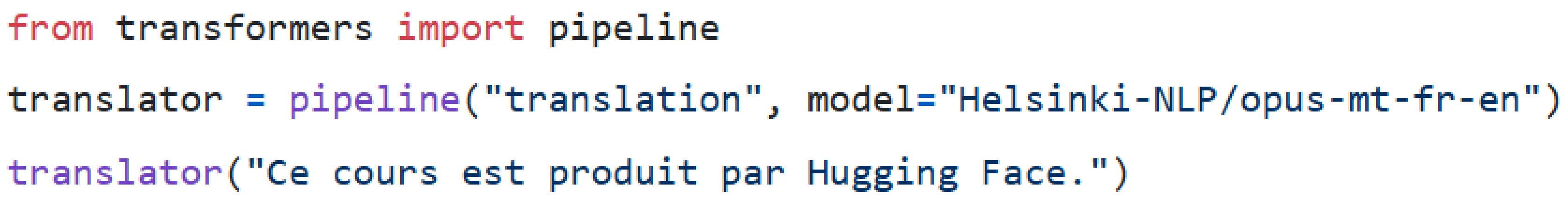
- Output:



```
[{'summary_text': ' America has changed dramatically during recent years . The number of engineeri
```

Translation:

- Code:

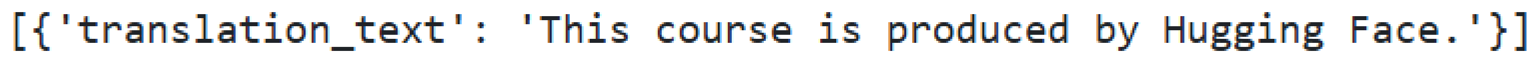


```
from transformers import pipeline

translator = pipeline("translation", model="Helsinki-NLP/opus-mt-fr-en")

translator("Ce cours est produit par Hugging Face.")
```

- Output:

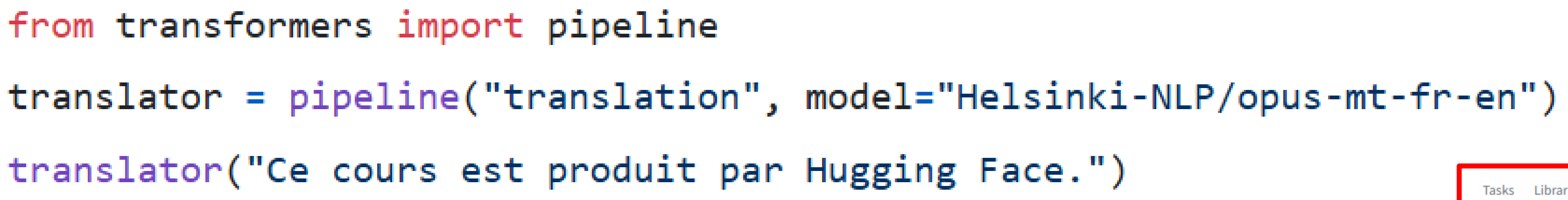


```
[{'translation_text': 'This course is produced by Hugging Face.'}]
```

How to deal with non-English texts?

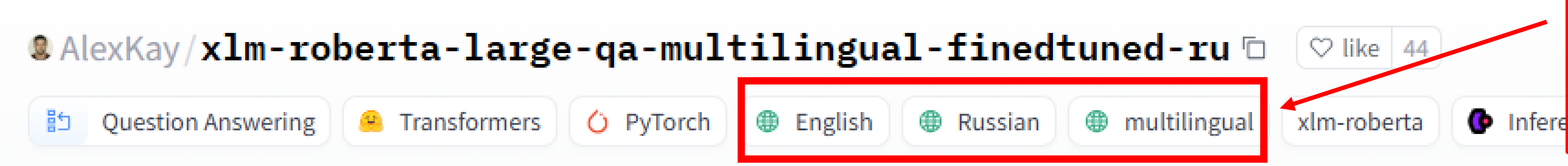
Two options:

1. Translate your current text to, for example, English, with pre-trained models designed for translation

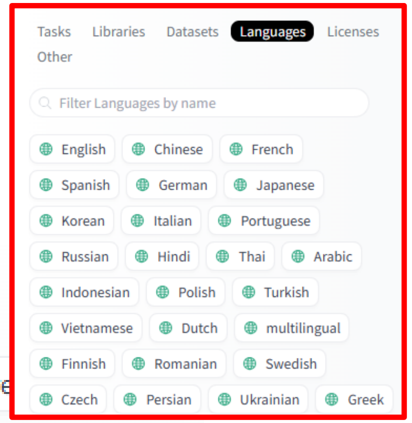


```
from transformers import pipeline
translator = pipeline("translation", model="Helsinki-NLP/opus-mt-fr-en")
translator("Ce cours est produit par Hugging Face.")
```

2. Use pre-trained models that can handle multilingual texts



AlexKay / xlm-roberta-large-qa-multilingual-finedtuned-ru



Think out-of-the-box & surprise us @ the datathon!

HAVE FUN!!!!