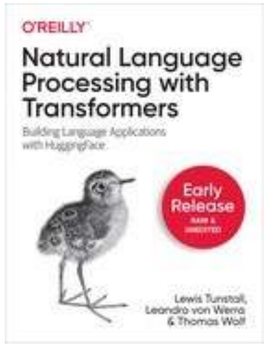
A Guided Tour Through the Transformers Landscape

SwissText – 15 June 2021

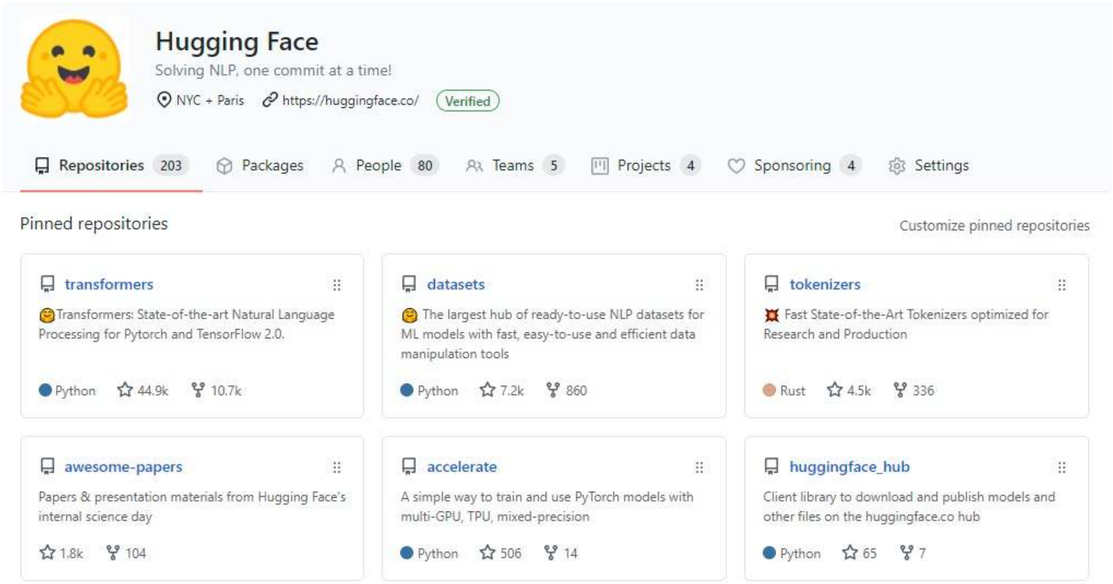
Lewis Tunstall | Open-source @ Hugging Face | lewis@huggingface.co

huggingface.co/course/



[NLP with Transformers]

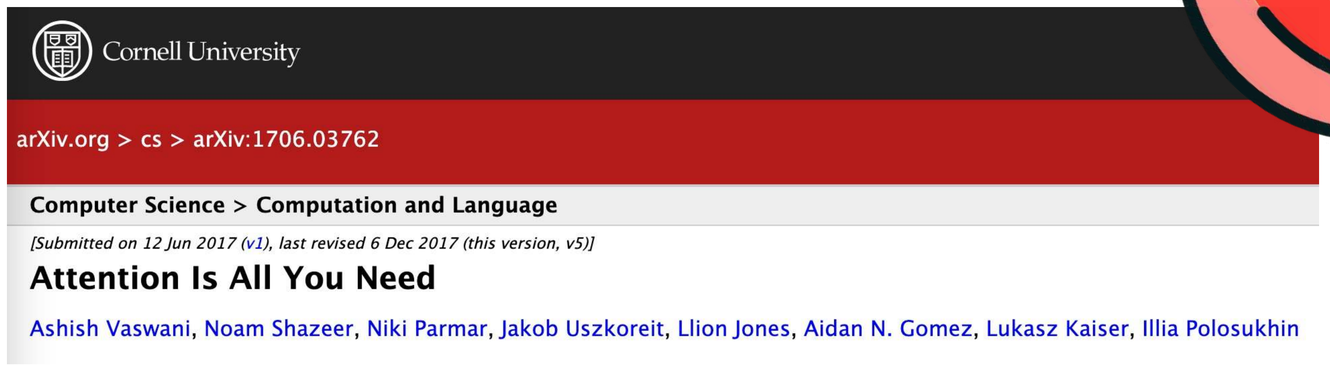
Education



Open Source

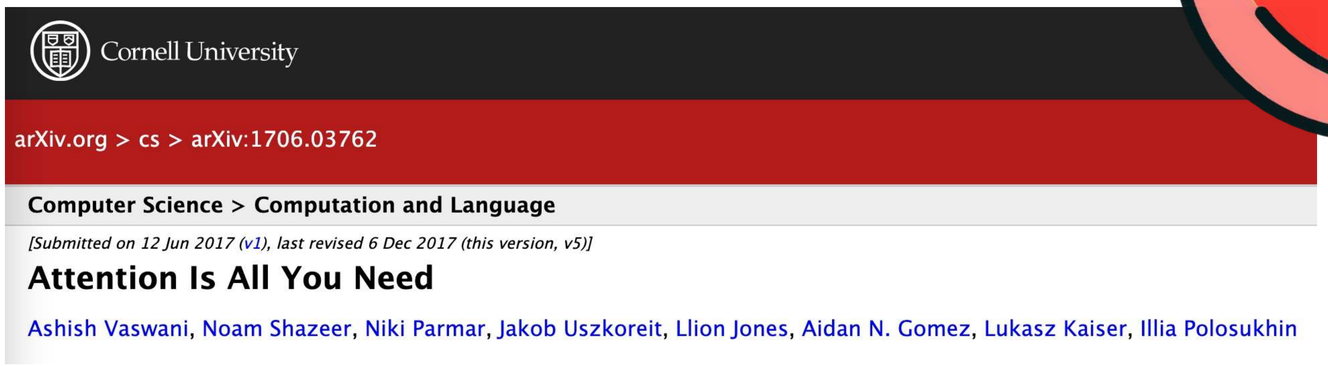
A brief history

4 years old
this month

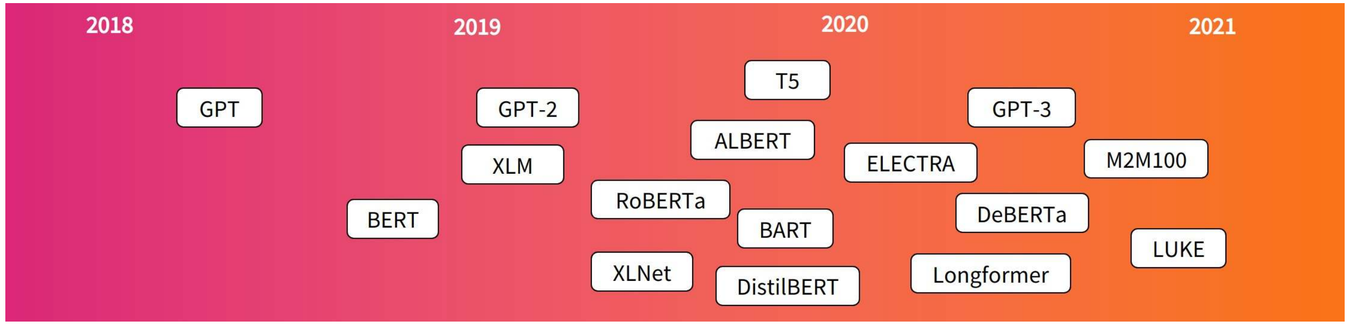


A brief history

4 years old this month



A "Cambrian explosion"



Main ingredients

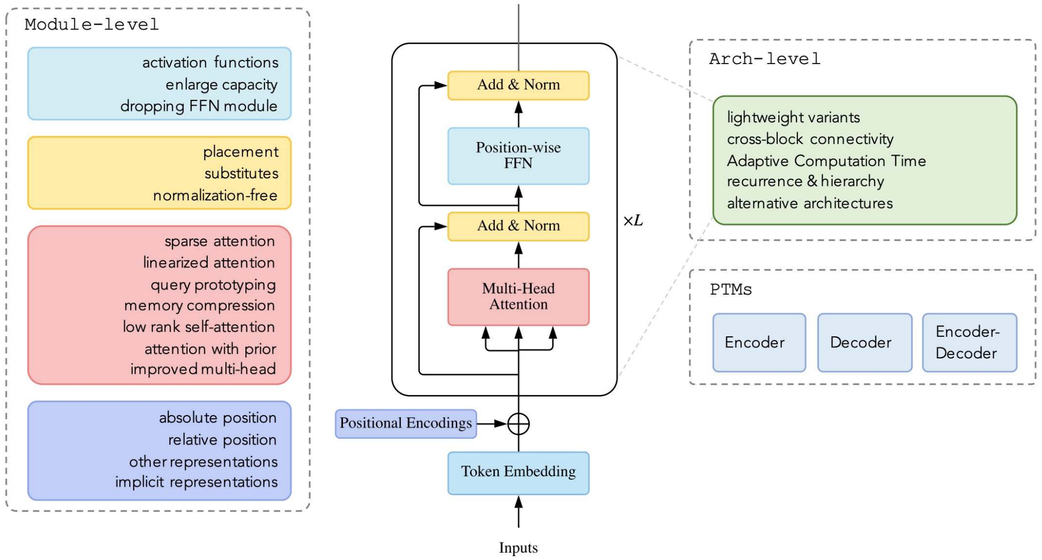


Image credit: [Lin et al (2021)]

Transformer Blocks

Main ingredients

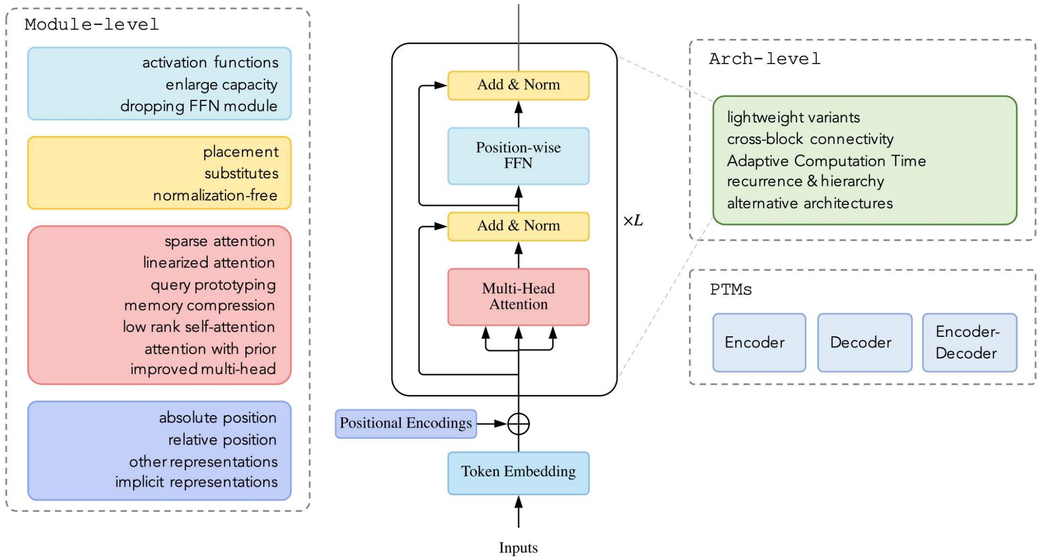
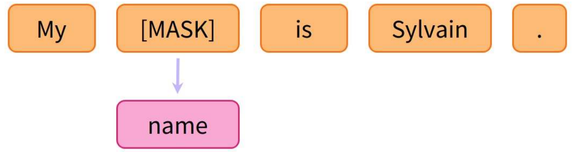
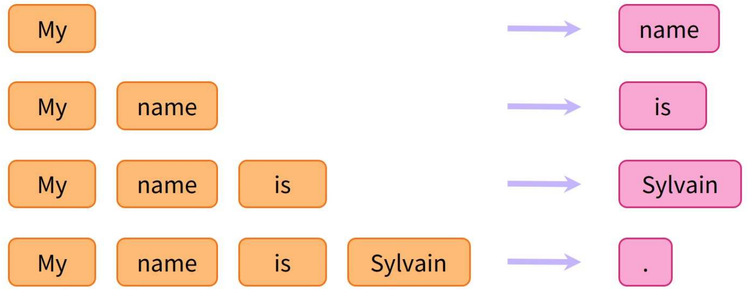


Image credit: [Lin et al (2021)]



Masked LM

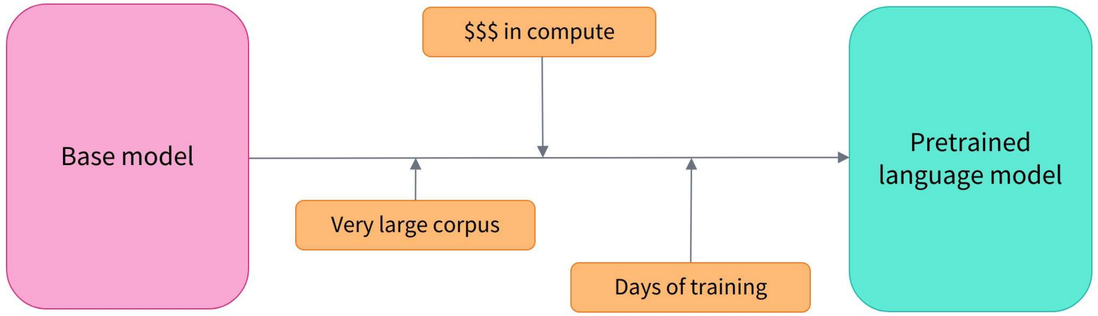


Autoregressive LM

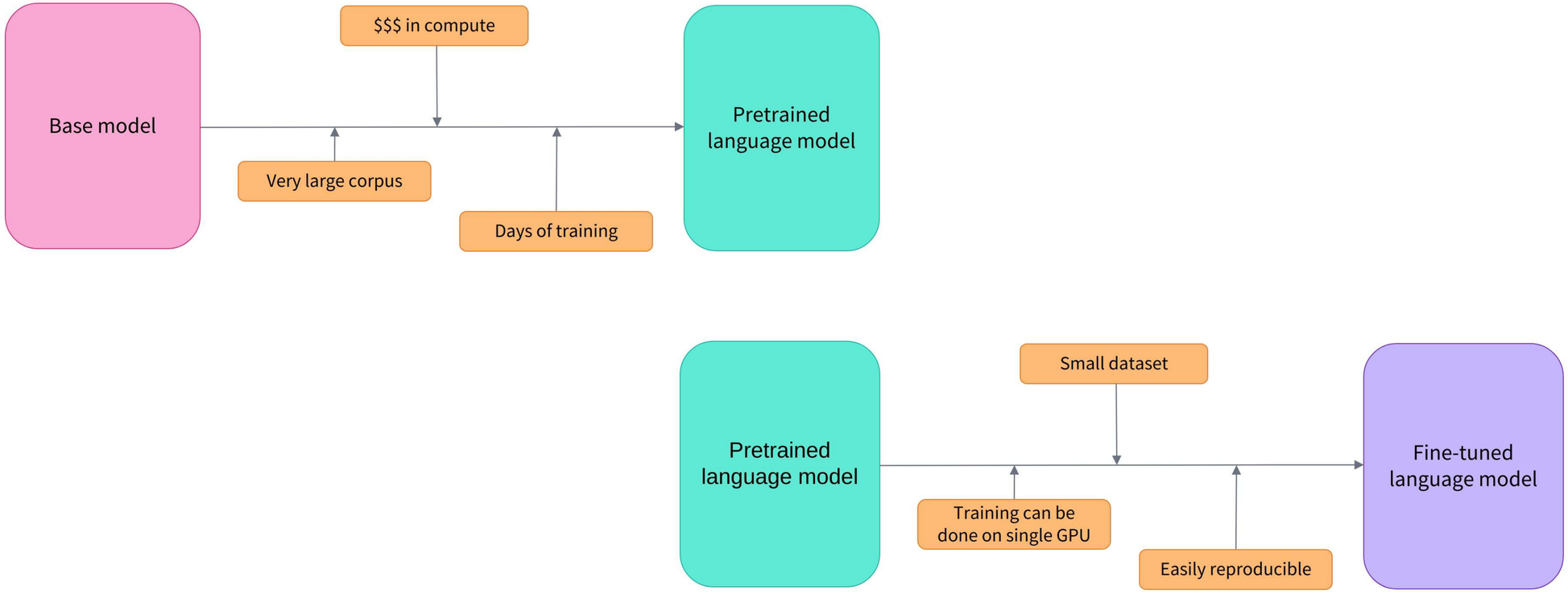
Transformer Blocks

Language Modelling

The modern paradigm



The modern paradigm



Works for vision too 🤯

Image GPT (iGPT)

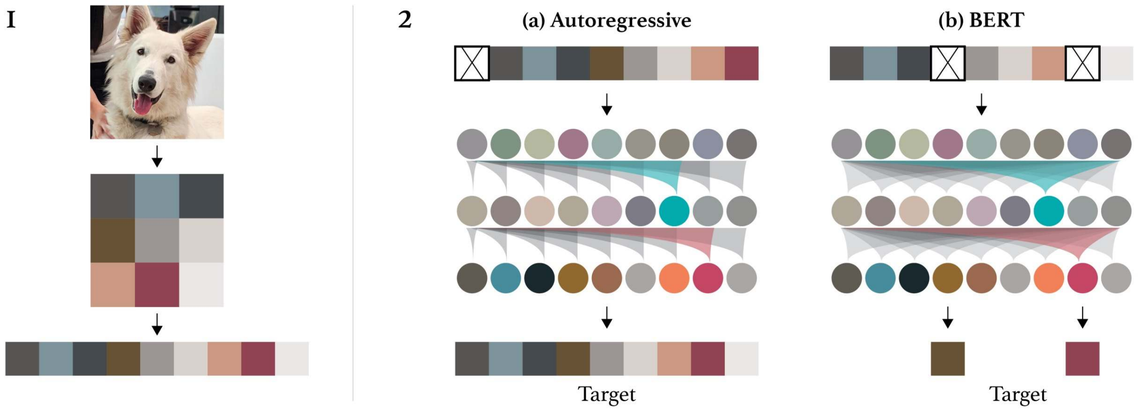


Image credit: Chen et al (2020)

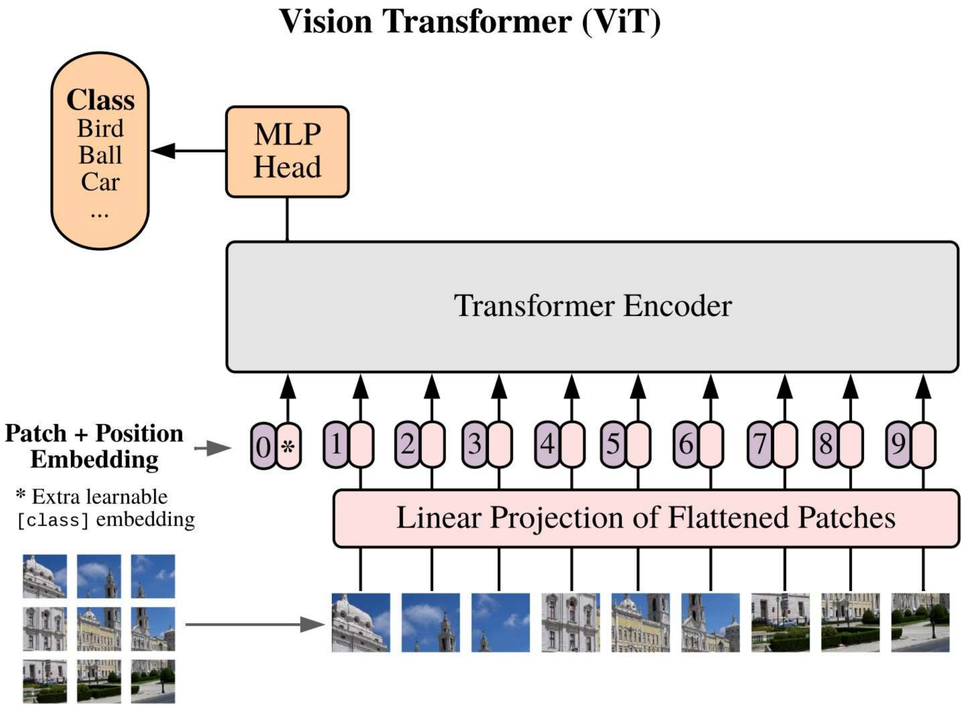


Image credit: Dosovitskiy et al (2020)

Including multiple modalities

TEXT PROMPT an illustration of a baby daikon radish in a tutu walking a dog

AI-GENERATED IMAGES

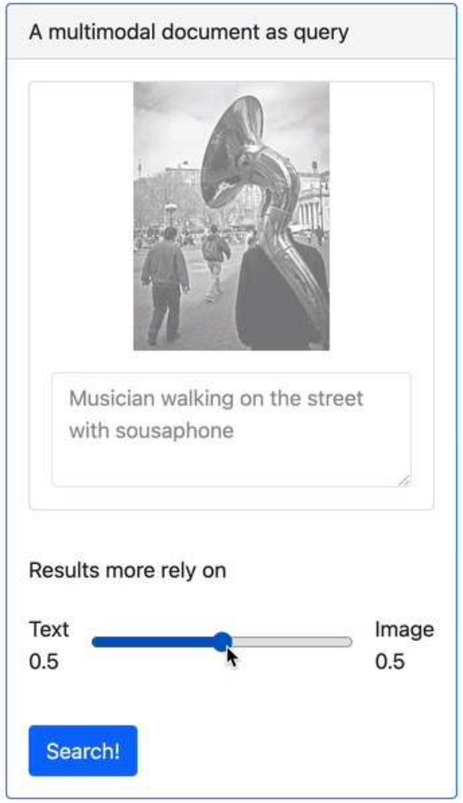
Edit prompt or view more images↓

TEXT PROMPT an armchair in the shape of an avocado. . . .

AI-GENERATED IMAGES

Edit prompt or view more images↓

[DALL-E by OpenAI]



Multimodal document search by [Jina]

Transformers are now everywhere

NLP

Speech

Vision

BioChem

Time Series

RL

Transformers are now everywhere

NLP

Speech

Vision

Domain X

BioChem

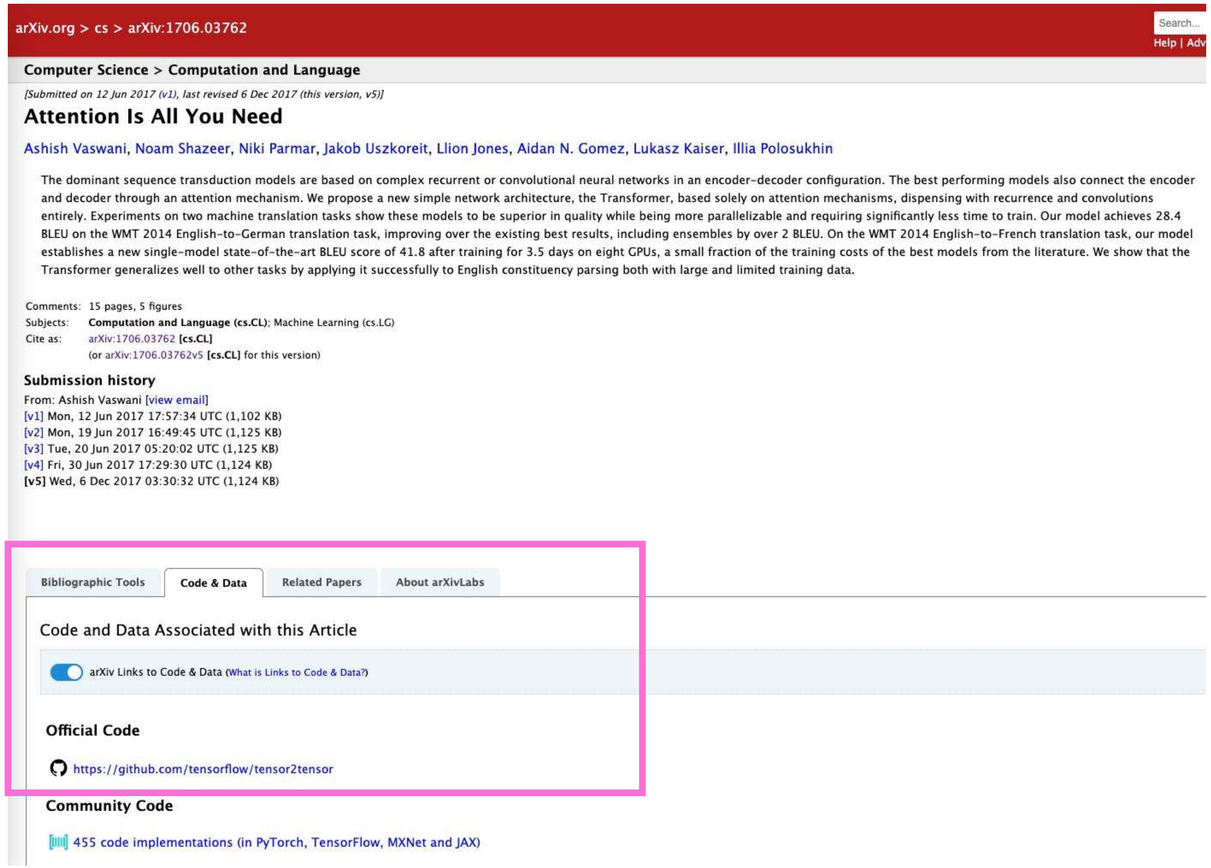
Time Series

RL

Bridging the science – industry divide

Humble Data Scientist

Get the code and model weights?



arXiv.org > cs > arXiv:1706.03762

Search... Help | Adv

Computer Science > Computation and Language

[Submitted on 12 Jun 2017 (v1), last revised 6 Dec 2017 (this version, v5)]

Attention Is All You Need

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks in an encoder-decoder configuration. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

Comments: 15 pages, 5 figures
Subjects: **Computation and Language (cs.CL)**; Machine Learning (cs.LG)
Cite as: arXiv:1706.03762 **[cs.CL]**
(or arXiv:1706.03762v5 **[cs.CL]** for this version)

Submission history

From: Ashish Vaswani [view email]
[v1] Mon, 12 Jun 2017 17:57:34 UTC (1,102 KB)
[v2] Mon, 19 Jun 2017 16:49:45 UTC (1,125 KB)
[v3] Tue, 20 Jun 2017 05:20:02 UTC (1,125 KB)
[v4] Fri, 30 Jun 2017 17:29:30 UTC (1,124 KB)
[v5] Wed, 6 Dec 2017 03:30:32 UTC (1,124 KB)

Bibliographic Tools | **Code & Data** | Related Papers | About arXivLabs

Code and Data Associated with this Article

arXiv Links to Code & Data (What is Links to Code & Data?)

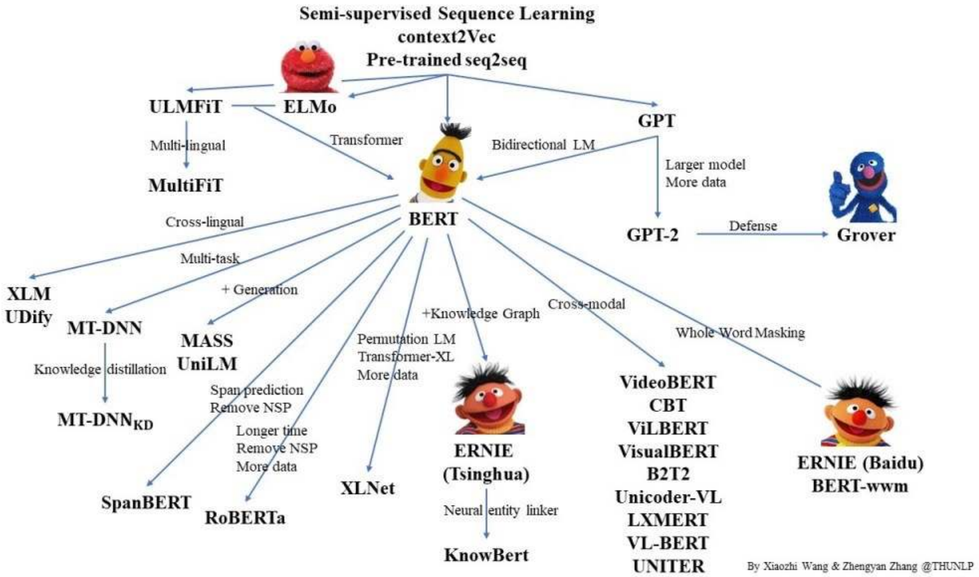
Official Code

https://github.com/tensorflow/tensor2tensor

Community Code

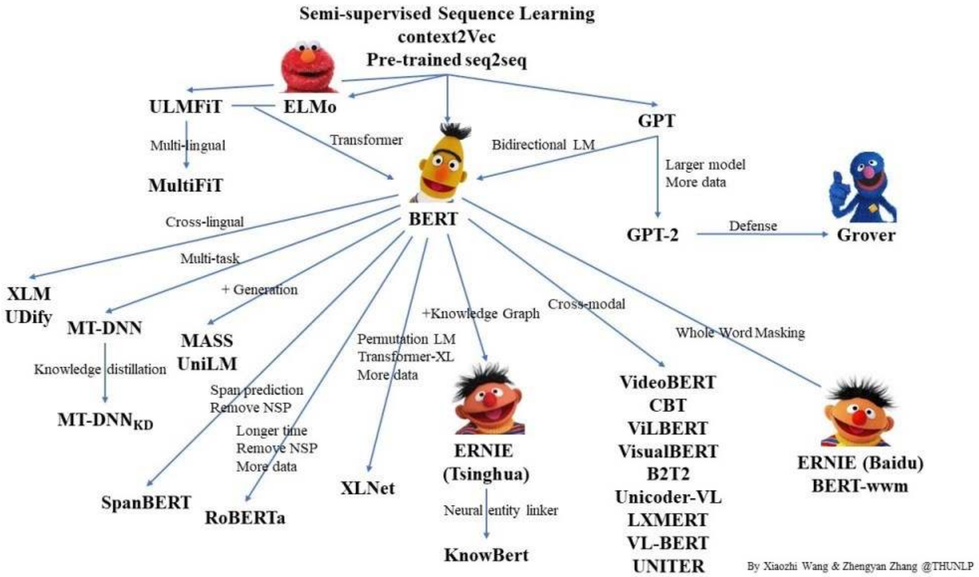
455 code implementations (in PyTorch, TensorFlow, MXNet and JAX)

The wild west of open-source



Explosion of pretrained models:
which one do I choose?

The wild west of open-source



Explosion of pretrained models:
which one do I choose?

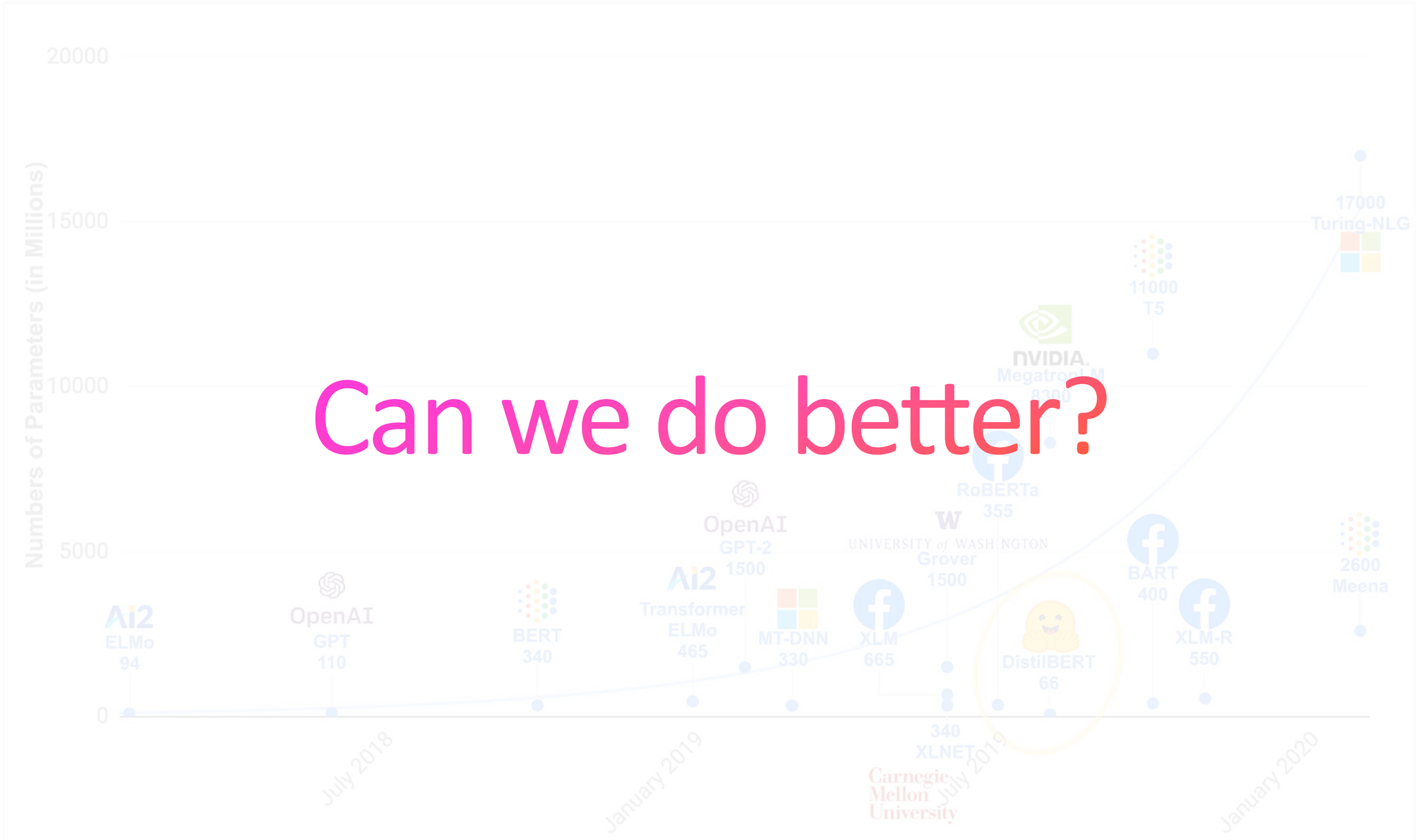
Python 2? Really? #8

Open impredicative opened this issue on 11 Jul 2019 · 3 comments

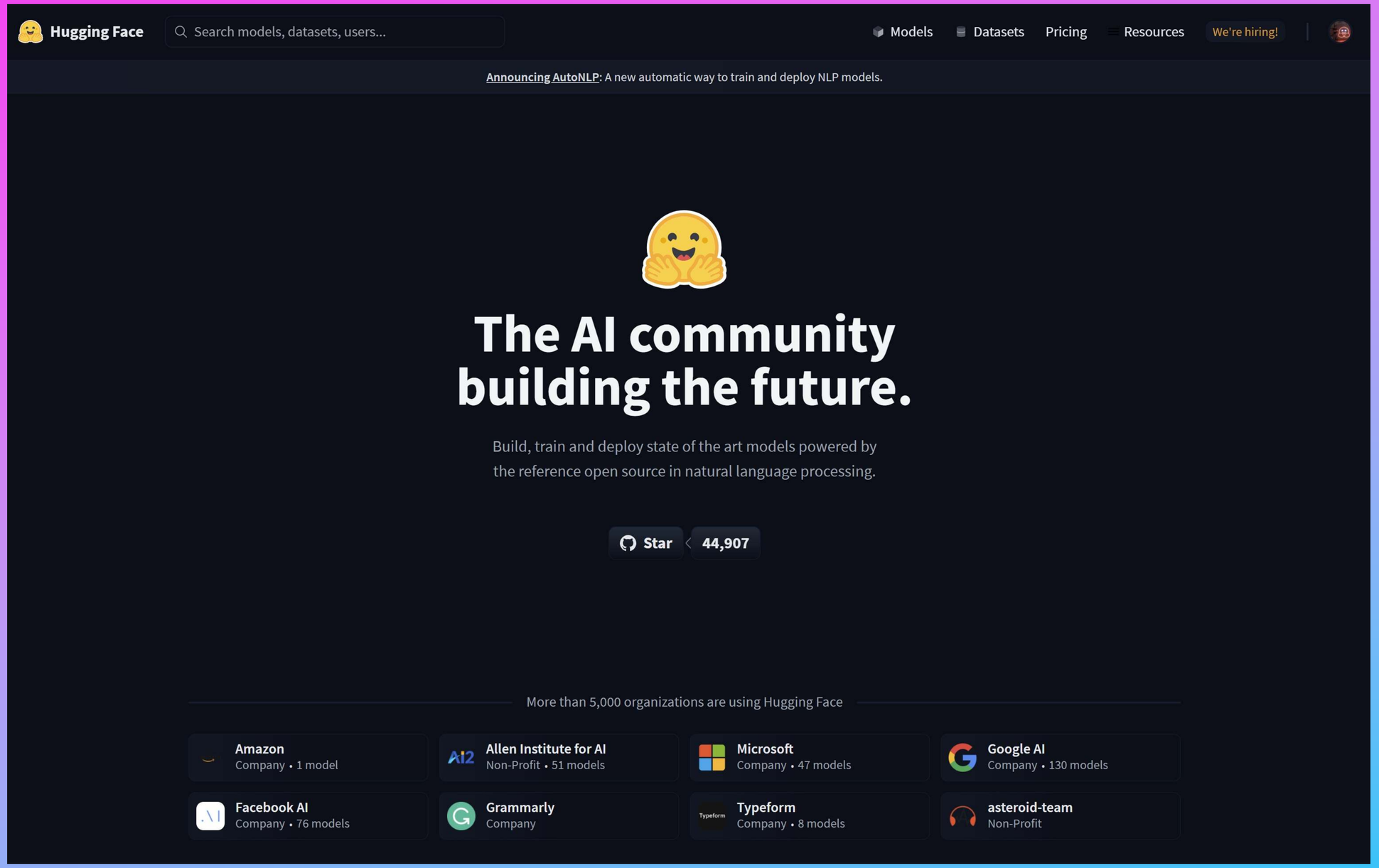
Oh c'mon you guys... #2

Open kcrosley-leisurelabs opened this issue on 18 Jun 2020 · 15 comments

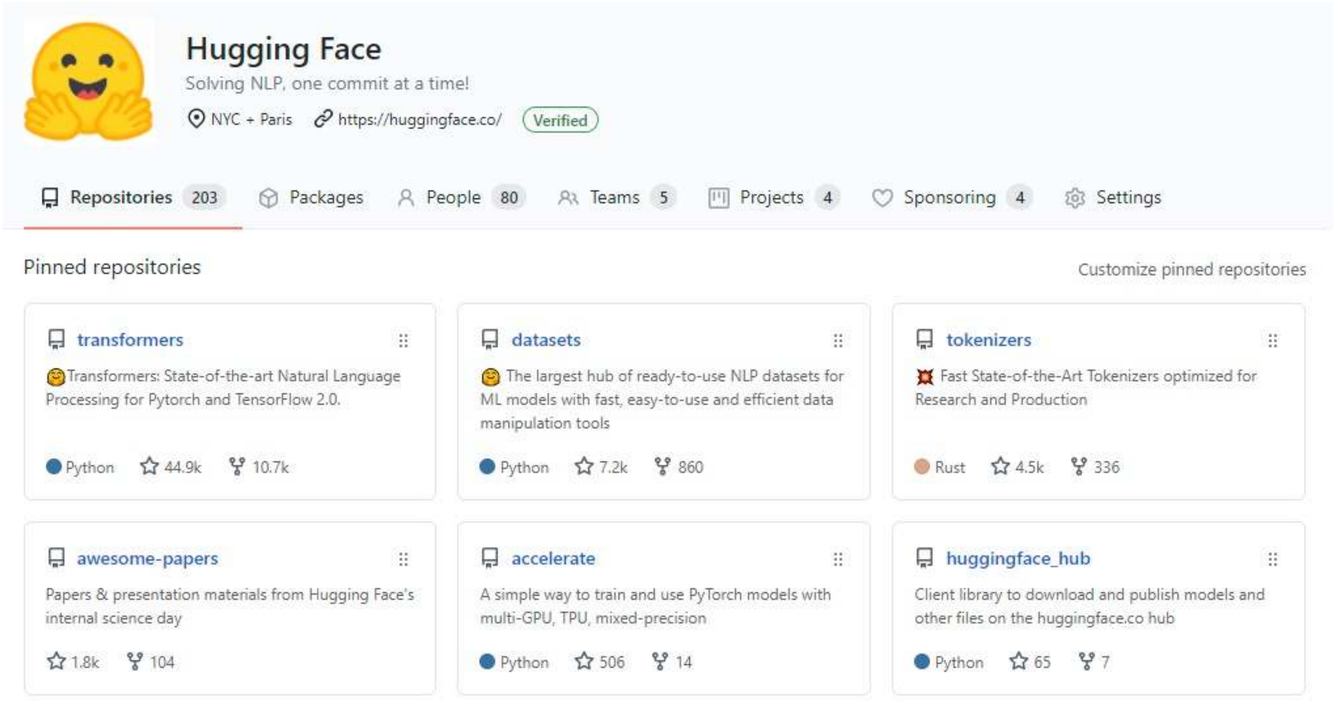
Different APIs, missing docs,
reproducibility issues, ...



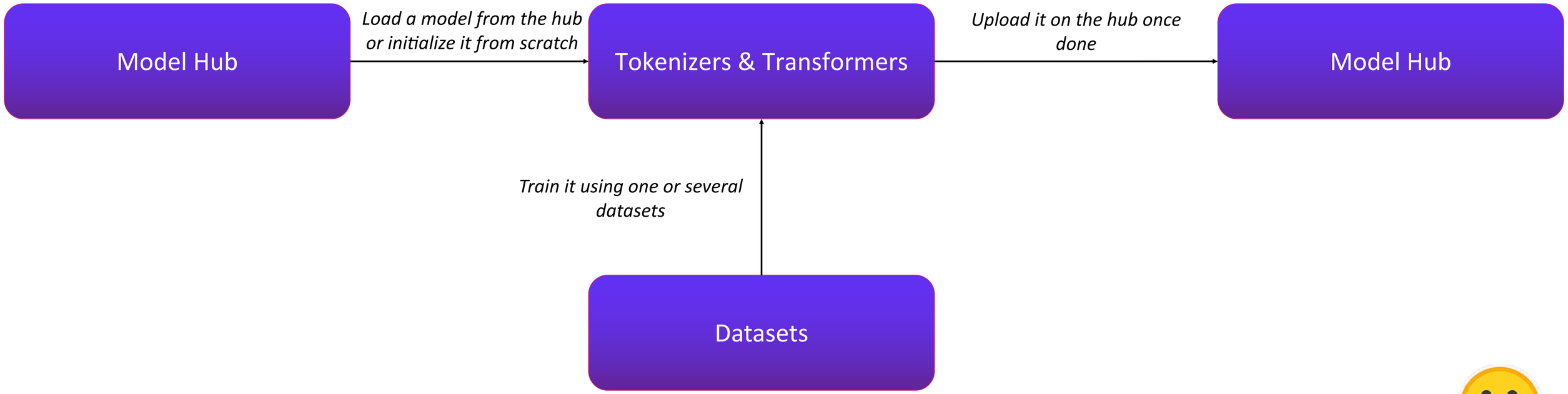
Can we do better?


Hugging Face
Search models, datasets, users...
Models
Datasets
Pricing
Resources
We're hiring!
Announcing AutoNLP: A new automatic way to train and deploy NLP models.
The AI community building the future.
Build, train and deploy state of the art models powered by the reference open source in natural language processing.
Star
44,907
More than 5,000 organizations are using Hugging Face
Amazon
Company • 1 model
Allen Institute for AI
Non-Profit • 51 models
Microsoft
Company • 47 models
Google AI
Company • 130 models
Facebook AI
Company • 76 models
Grammarly
Company
Typeform
Company • 8 models
asteroid-team
Non-Profit

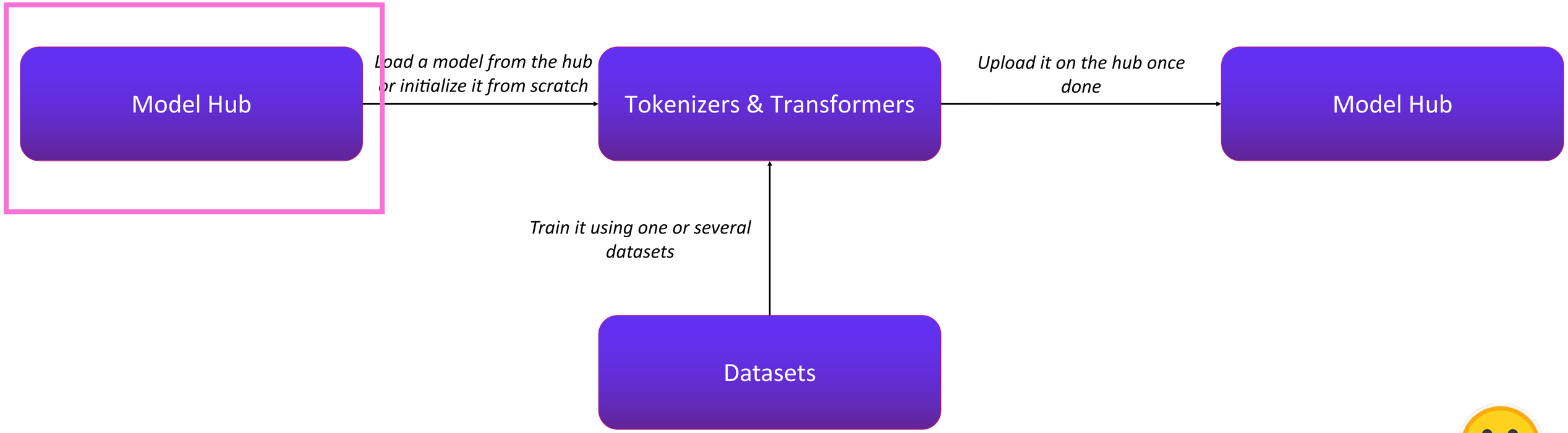
Open-source @ Hugging Face



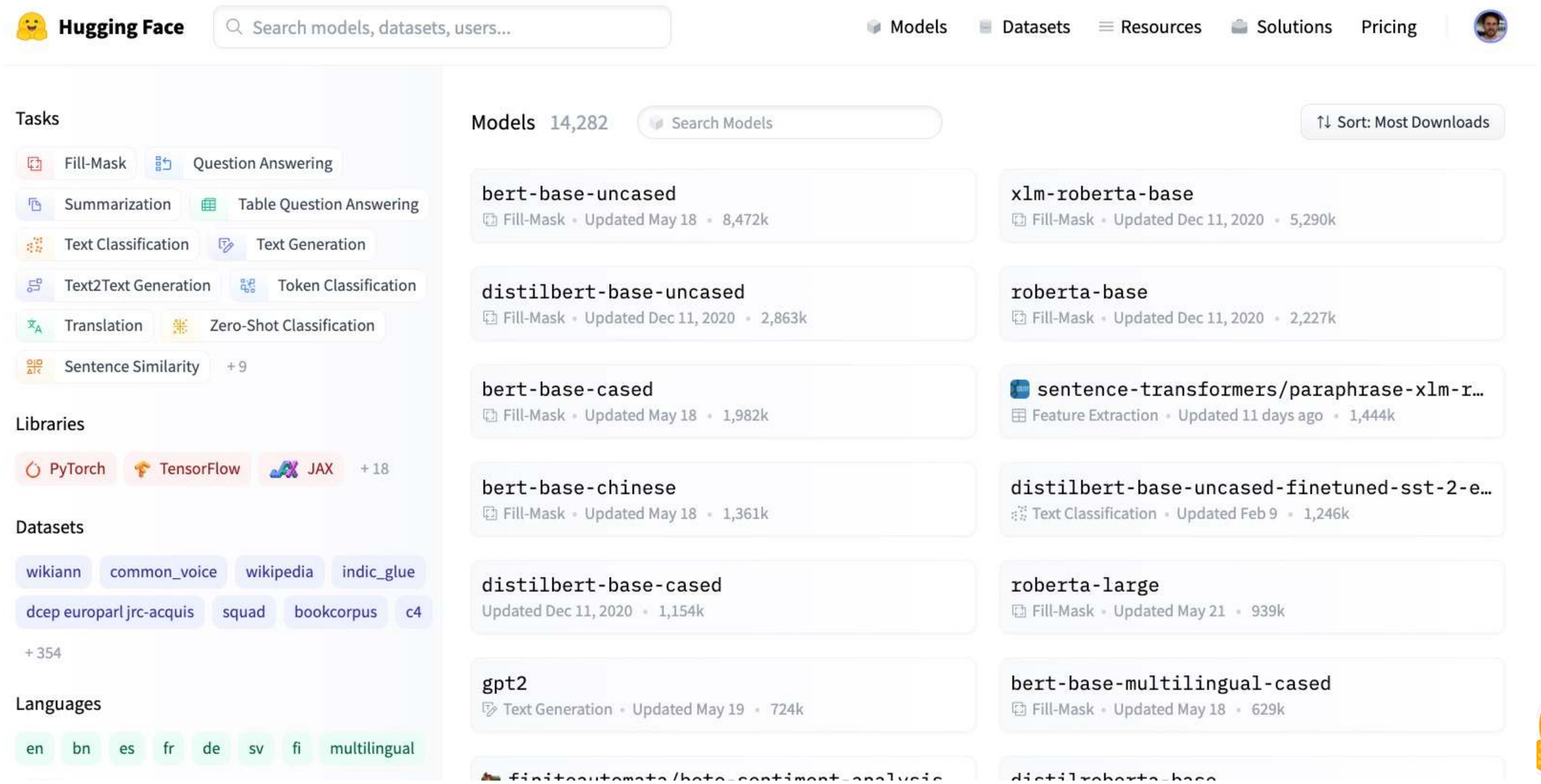
The Hugging Face ecosystem



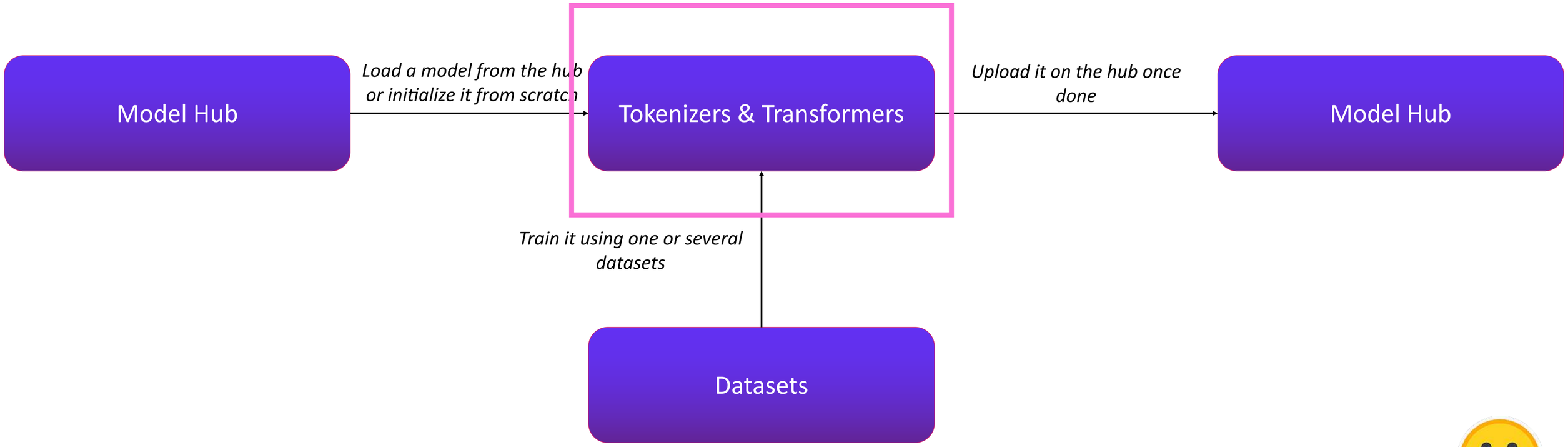
The Hugging Face ecosystem



The Model Hub



The Hugging Face ecosystem



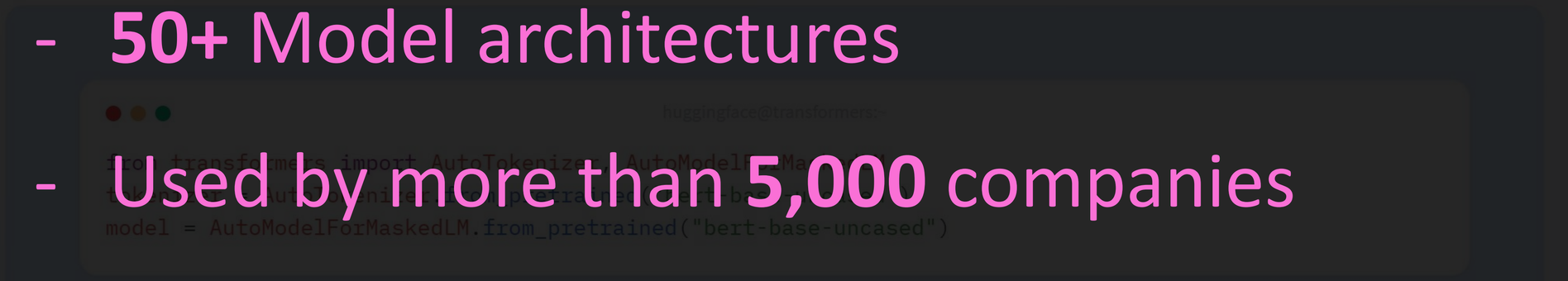
Tokenizers and Transformers

huggingface@transformers:~

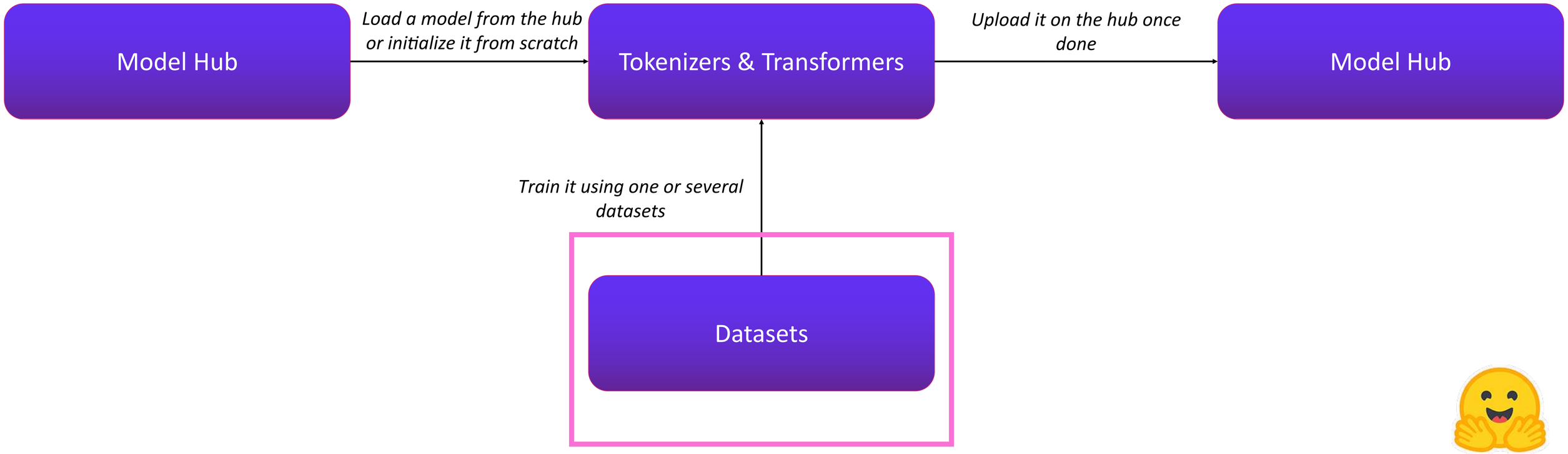
```python
from transformers import AutoTokenizer, AutoModelForMaskedLM
tokenizer = AutoTokenizer.from_pretrained("bert-base-uncased")
model = AutoModelForMaskedLM.from_pretrained("bert-base-uncased")
```

Tokenizers & Transformers

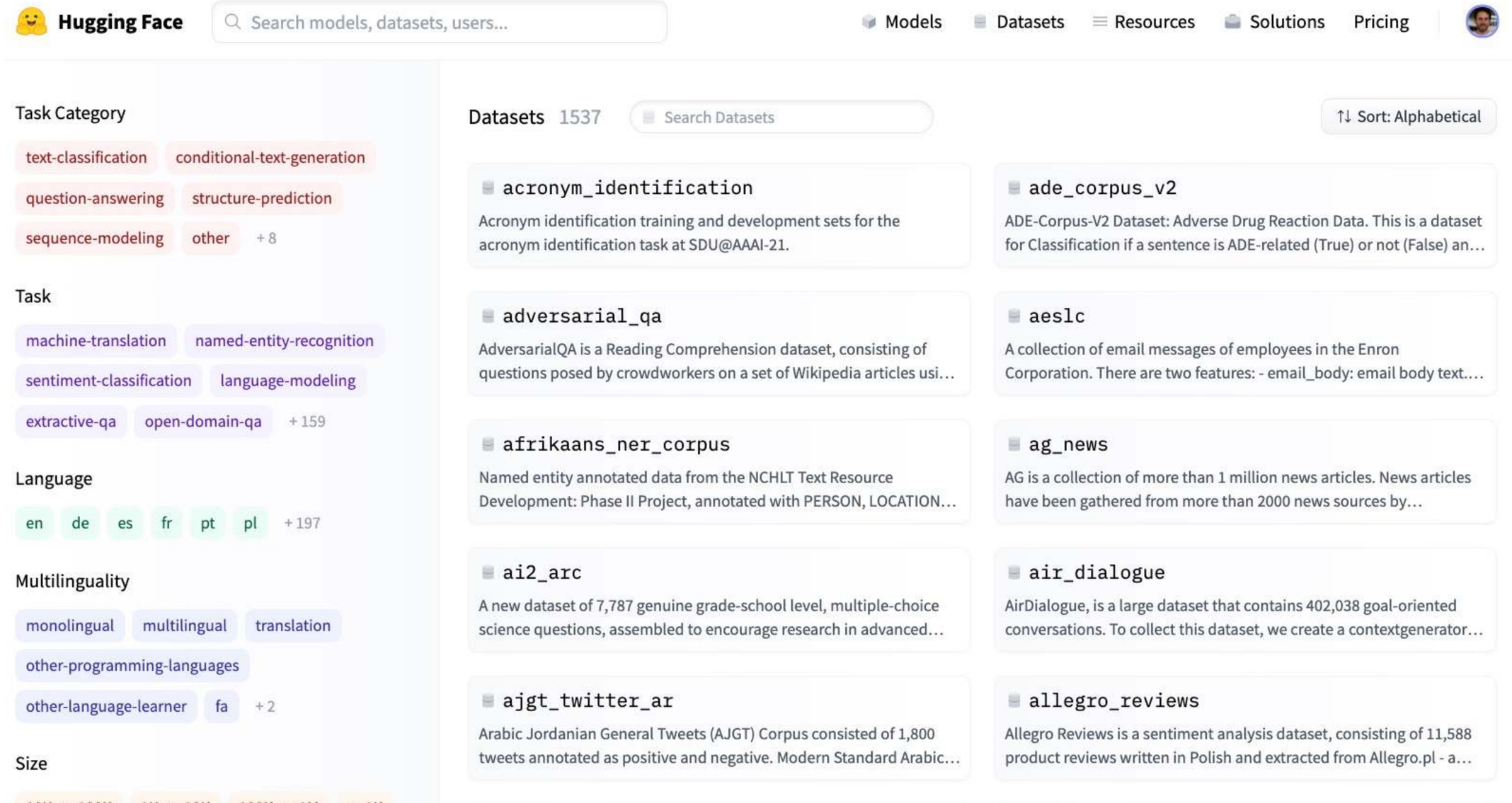
- **50+** Model architectures
- Used by more than **5,000** companies
- Simple API across all architectures



The Hugging Face ecosystem



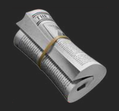
The Datasets Hub



Datasets

The Hub:

- Largest hub of ready-to-use datasets
- **1000+** Datasets available
- **450+** languages and dialects supported

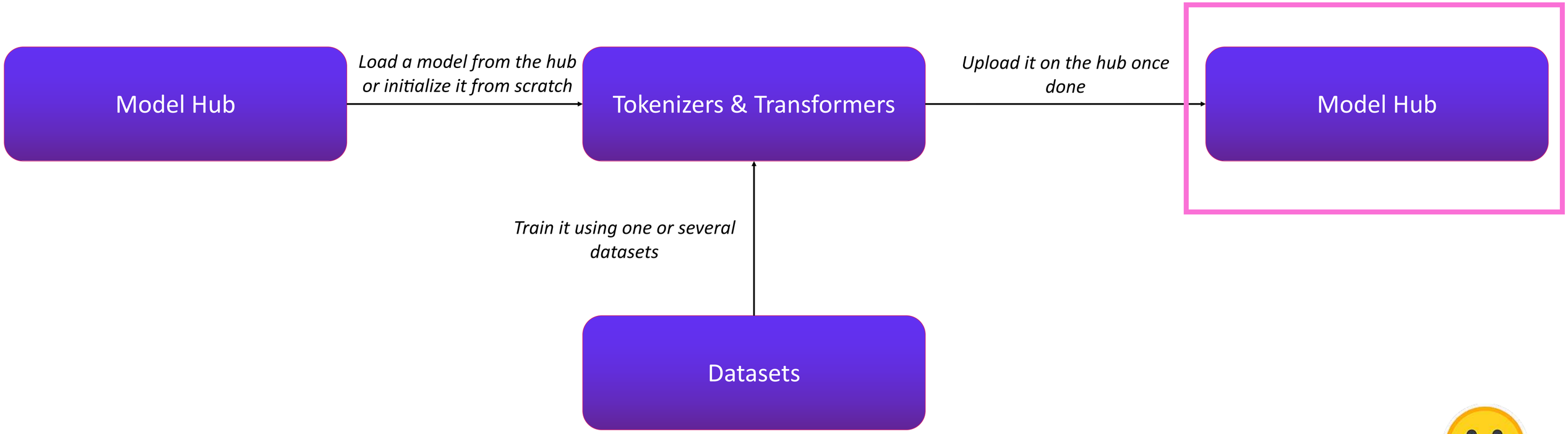


Datasets

The python library:

- Load any dataset in **one line**
- Supports **huge** datasets without RAM limitations
- **Fast** iterations and querying

The Hugging Face ecosystem

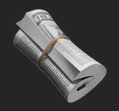


Hugging Face Hub

```python
from huggingface_hub import HfApi()

api = HfApi()
```

Hugging Face Hub

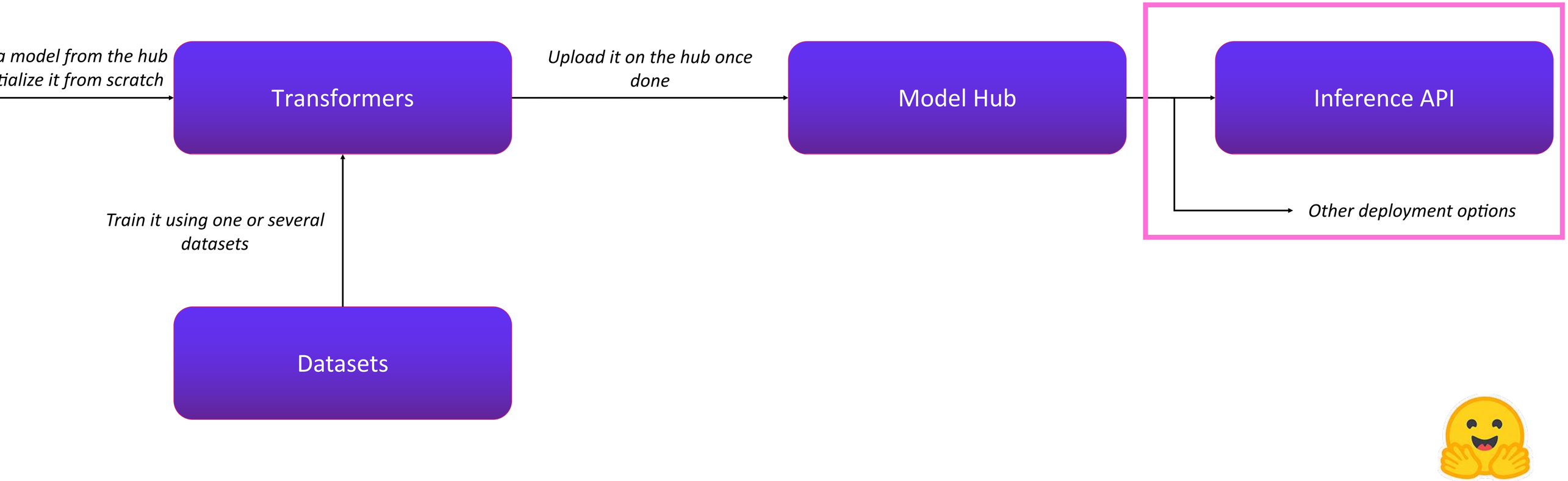
```python
class DummyModel(nn.Module, ModelHubMixin):

    def __init__(self, **kwargs):
        super().__init__()
        self.config = kwargs.pop("config", None)
        self.l1 = nn.Linear(2, 2)

    def forward(self, x):
        return self.l1(x)

model = DummyModel()
model.save_pretrained("my-dummy-model")
model.push_to_hub("my-dummy-model", organization="huggingface")

# Reload it from any device!

model = DummyModel.from_pretrained("huggingface/my-dummy-model")
```



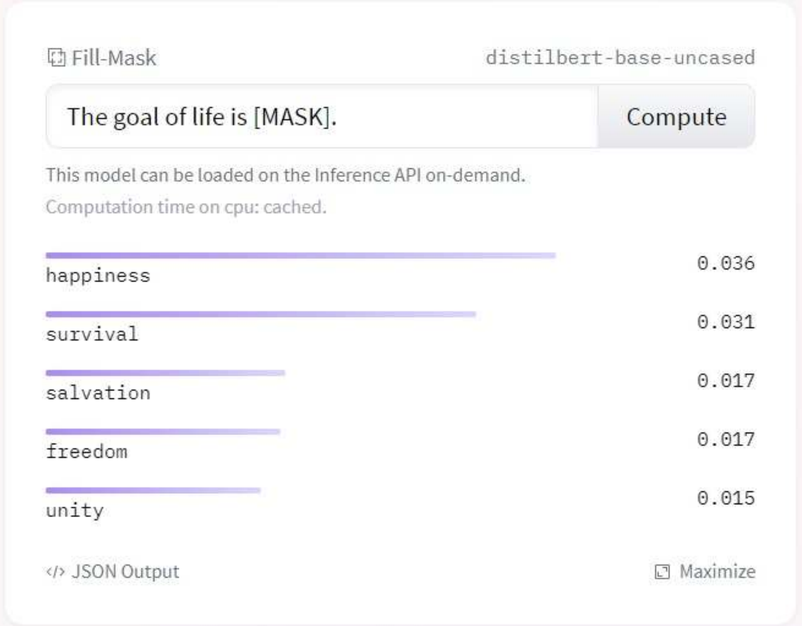
HuggingFace Hub

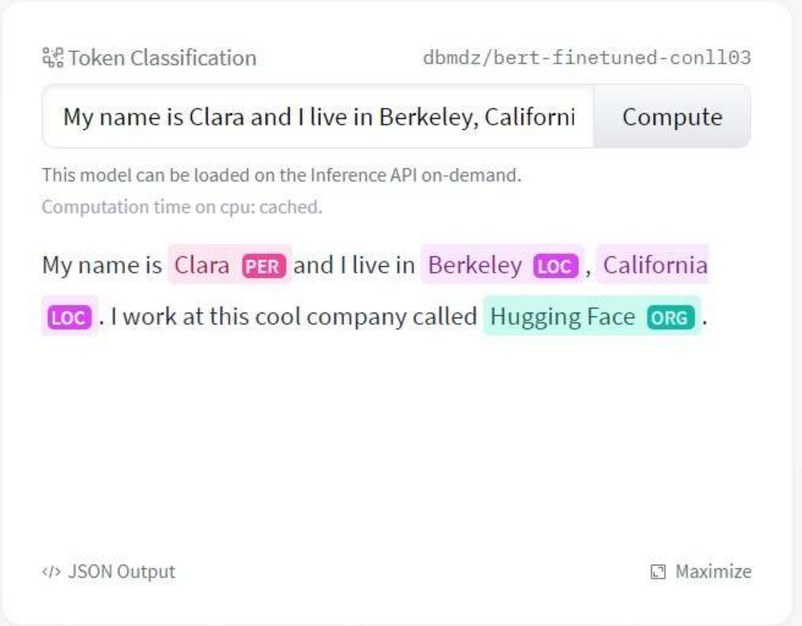
- Python API
- Simple mixin for your PyTorch Module
- Supports Transformers, AllenNLP, Asteroid, Spacy, Timm …

The Hugging Face ecosystem

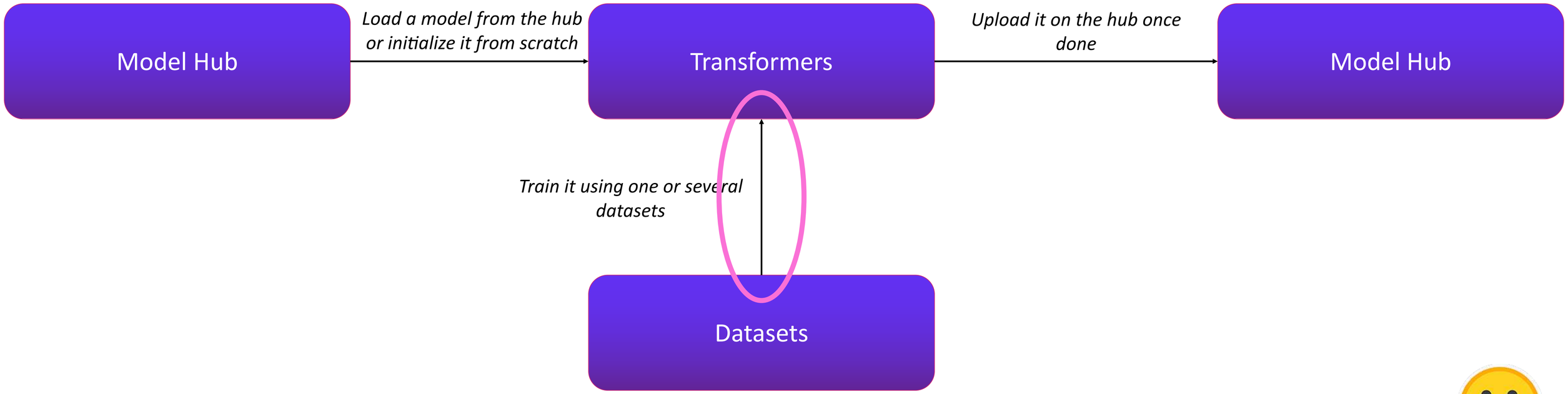


Inference API





The Hugging Face ecosystem



Three ways to train a model

Trainer

```python
from transformers import AutoModelForSequenceClassification, Trainer, TrainingArguments
model = AutoModelForSequenceClassification.from_pretrained("bert-base-uncased")
training_args = TrainingArguments(
    output_dir="./my-finetuned-model",  # output directory
    num_train_epochs=3,                 # total number of training epochs
    per_device_train_batch_size=16,     # batch size per device during training
    per_device_eval_batch_size=64,      # batch size for evaluation
    evaluation_stategy="epoch",         # evaluate every epoch
    weight_decay = 0.01,                # strength of weight decay
)
trainer = Trainer(
    model=model,                        # model to be trained
    args=training_args,                 # training arguments, defined above
    train_dataset=train_dataset,        # the training set
    eval_dataset=test_dataset,          # the evaluation set
    tokenizer=tokenizer,                # the tokenizer used for preporcessing
)
train.train()
train.push_to_hub()
```

Accelerate

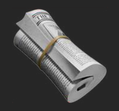
```
  import torch
  import torch.nn.functional as F
  from datasets import load_dataset
+ from accelerate import Accelerator

+ accelerator = Accelerator()
- device = 'cpu'
+ device = accelerator.device

  model = torch.nn.Transformer().to(device)
  optim = torch.optim.Adam(model.parameters())

  dataset = load_dataset('my_dataset')
  data = torch.utils.data.DataLoader(dataset, shuffle=True)

+ model, optim, data = accelerator.prepare(model, optim, data)
```

```
  model.train()
  for epoch in range(10):
      for source, targets in data:
          source = source.to(device)
          targets = targets.to(device)

          optimizer.zero_grad()

          output = model(source)
          loss = F.cross_entropy(output, targets)

-         loss.backward()
+         accelerator.backward(loss)

          optimizer.step()
```

Accelerate: handling devices

```
  import torch
  import torch.nn.functional as F
  from datasets import load_dataset
+ from accelerate import Accelerator

+ accelerator = Accelerator()
- device = 'cpu'

- model = torch.nn.Transformer().to(device)
+ model = torch.nn.Transformer()
  optim = torch.optim.Adam(model.parameters())

  dataset = load_dataset('my_dataset')
  data = torch.utils.data.DataLoader(dataset, shuffle=True)

+ model, optim, data = accelerator.prepare(model, optim, data)
```

```
  model.train()
  for epoch in range(10):
      for source, targets in data:
-         source = source.to(device)
-         targets = targets.to(device)

          optimizer.zero_grad()

          output = model(source)
          loss = F.cross_entropy(output, targets)

-         loss.backward()
+         accelerator.backward(loss)

          optimizer.step()
```



Accelerate

Currently supported:

- Training
- Evaluation
- CPU, GPU, Multi GPU, TPU, Mixed precision
- AWS Sagemaker

Upcoming:

- Horovod
- FairScale
- DeepSpeed
- Others? 🤗

auto NLP

```
# Upload your model data
autonlp upload --project sentiment_detection --split train
               --col_mapping review:text,sentiment:target
               --files ~/datasets/train.csv

# Train your model
autonlp train --project sentiment_detection

# Use your model
curl -X POST
        -H "Authorization: Bearer API api_jeZrkpoqfjzioaRaerjlbRQeKykrop"
        -H "Content-Type: application/json"
        -d '{"inputs":"The goal of life is [MASK]"}'
        https://api-inference.huggingface.co/models/sentiment_detection
```

Big Science

bigscience.huggingface.co/

What is Big Science?

One-year research workshop on large multilingual datasets and large language models

Analogy with the Large Hadron Collider at CERN:

- has involved 10,000+ researchers
- from 100+ countries
- lead to the discovery of 59 hadrons
- publication of more than 2,800 papers (😱)

In many scientific fields (epidemiology, space, fusion…), large-scale and worldwide research collaborations create tools useful for the entire research community, like the LHC, ITER, ISS…

Isn't it time to build similar large, diverse, open research collaborations in AI/NLP as well?

Why do this?

Research

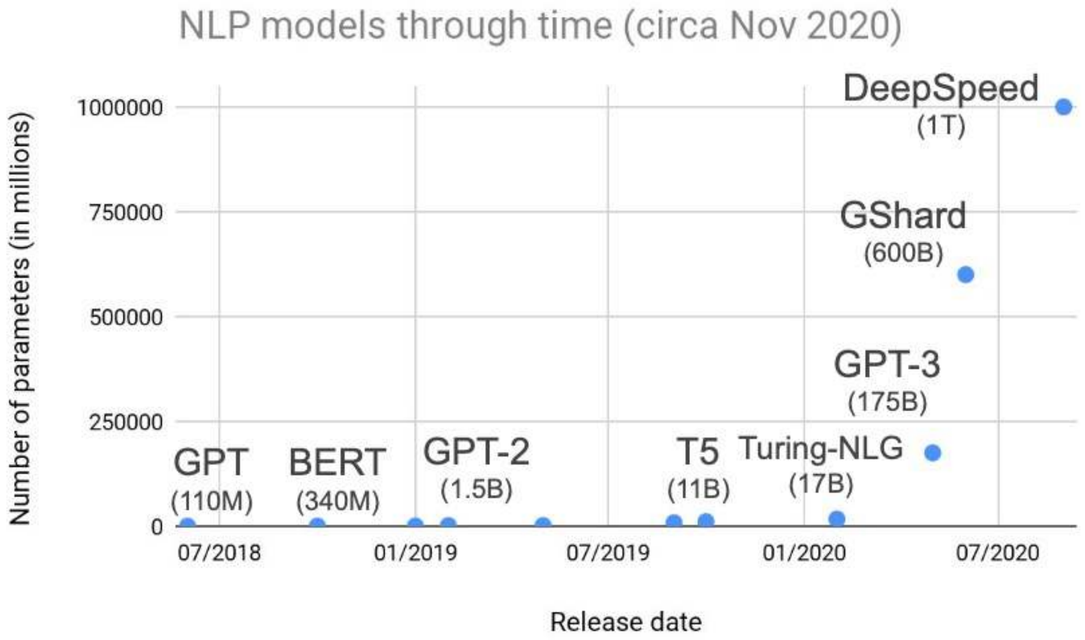
- Models not designed as general research tools
- Difficult involvement of academic researchers
- Lack of fields diversity of the research teams building them

Environmental

- Training parallel models in private setting => duplication of energy requirements
- Carbon footprint not documented/taken into account

Ethical and societal

- Shortcomings in the text corpora used to train these models
- Ethical/bias/usage question are usually asked a-posteriori



But large-scale public compute exists

Accelerated partition (or GPU partition)

- 261 four-GPU accelerated compute nodes with:
  - 2 Intel Cascade Lake 6248 processors (20 cores at 2.5 GHz), namely 40 cores per node
  - 192 GB of memory per node
  - 4 Nvidia Tesla V100 SXM2 GPUs (32 GB)
- 31 eight-GPU accelerated compute nodes, currently dedicated to the AI community with:
  - 2 Intel Cascade Lake 6226 processors (12 cores at 2.7 GHz), namely 24 cores per node
  - 20 nodes with 384 GB of memory and 11 nodes with 768 GB of memory
  - 8 Nvidia Tesla V100 SXM2 GPUs (32 GB)
- Extension in the summer of 2020, 351 four-GPU accelerated compute nodes with:
  - 2 Intel Cascade Lake 6248 processors (20 cores at 2.5 GHz), namely 40 cores per node
  - 192 GB of memory per node
  - 4 Nvidia Tesla V100 SXM2 GPUs (16 GB)

• Cumulated peak performance of 28 Pflop/s with a total of **2696 Nvidia V100 GPUs**

Jean Zay supercomputer in France

A brief history

- 🐣 Early 2021: Discussions between Thomas Wolf (HuggingFace), Stéphane Requena (GENCI) and Pierre-François Lavallée (IDRIS)
- 👨‍👧 Very quickly: Science team of HF + many members of the French academic and industrial AI and NLP research communities joined the discussion to further develop the project leading to the grant application
- February 2021: Grant application for 5 million GPU hours
- 🌐 Following the grant submission: open/extend to international research community
- 🛠️ When the project reached 200+ participants: the organization of the project started to take shape and to adopt the structure of a research workshop
- 🚀 19/04: Grant accepted -- first half of the project
- 🌸 28/04: Kickoff event + project becomes public

Core research questions

- Large models + large datasets: exhibit intriguing and quite surprising behaviors from a research point of view

- Raise many research questions across many fields/subfields of AI/NLP:
  - Fundamental:
    - limits of what can be done with purely statistical and text-based approaches?
    - notion of what is an NLP task and what is the relation between a task and a dataset
  - Bias/fairness:
    - notion of bias and its relation to the dataset and training objectives
    - representativeness and stereotypes
    - memorization versus generalization and personal information memorization
  - Environmental impact and carbon footprint
  - And so many others (interpretability, relation to cognitive processes, use in linguistics…)

How can I participate?

- **General Advisor** (Steering Committee member):
  - *role* - give general scientific/organization advice - everyone here is in the SC by default
  - *time commitment* - light - reading a newsletter every 2 weeks - giving feedback/advice
- **Join** a **Working Group** (Organizing Committee member)
  - *role* - join one of the Working Groups to advise or participate (code, research...)
  - *time commitment* - medium - depends on the chosen WG
- **Chair/co-Chair** a **Working Group** (Organizing Committee member)
  - *role* - the chairs are responsible for providing the minimal amount of work necessary for having a barebone version of the task. If WG members are active, the chairs can mostly coordinate the effort and organise the decision process.
  - *time commitment* - more significant - depends on the chosen WG
- A **Workshop attendant** joining live events or some community events (tbd)
  - *role* - participating in the collaborative task following guidelines by the WG
  - *time commitment* - free - up to the attendant - open to anyone, beginners, people outside of the research fields, etc… - very accessible

Numbers of Parameters (in Millions)
20000
15000
10000
5000
0
July 2018
January 2019
July 2019
January 2020
AI2
ELMo
94
OpenAI
GPT
110
BERT
340
AI2
Transformer
ELMo
465
OpenAI
GPT-2
1500
MT-DNN
330
XLM
665
Grover
1500
340
XLNET
Carnegie Mellon University
RoBERTa
355
DistilBERT
66
NVIDIA.
MegatronLM
8300
11000
T5
17000
Turing-NLG
BART
400
XLM-R
550
2600
Meena

Questions!