

Chatbots

Peckham DAZ – Week 15

Credits – Terrance Broad

Today we'll cover...

1

The history of Chatbots

2

Generative Pre-trained Transformers

3

Hugging Face Pipelines

Chatbot History

- What is a Chatbot?
- Where did they come from?
- Rules based vs AI

What is a chatbot?

- A software application designed to simulate human-like conversations through text or voice interactions.
- Can be integrated into websites, messaging apps, or other platforms to assist users.

Key Features of Chatbots:

- **Text or Voice interaction**
- **User Assistance**
- **Automation**



Rules Based vs AI Chatbots

Rules based

- Based on pre-defined rules and keywords
- They follow a scripted flow that is setup in advance
- Replies are scripted
- Used widely in customer support for FAQ's and assistance

AI Based

- Uses NLP and ML to understand and respond to a user input.
- Learns from new data to improve responses
- Can handle complex queries and multi-turn conversations
- Can detect emotion through sentiment analysis

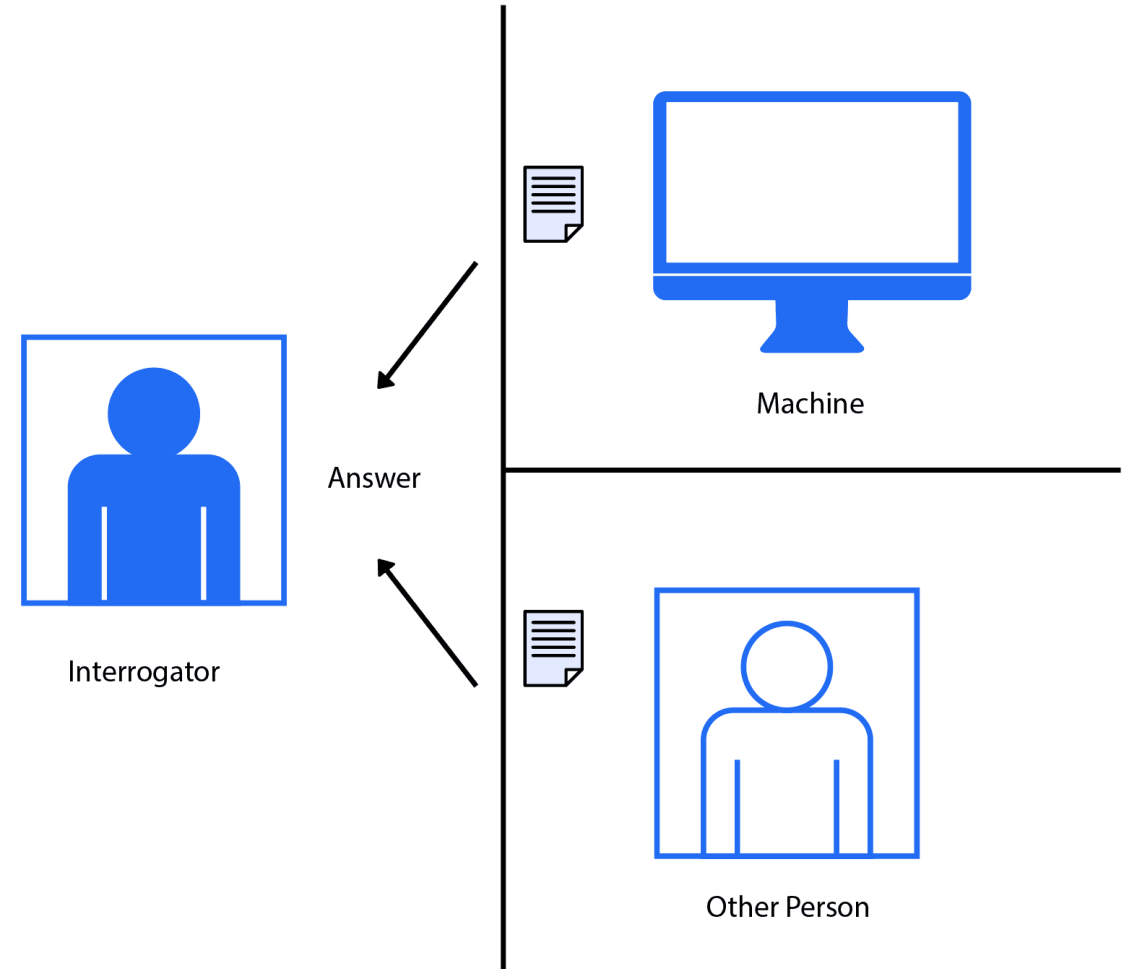
Rules based bots are still relevant!

History of Chatbots

The Turing Test

- In his seminal 1950 paper 'Computing machinery and Intelligence' Alan Turing described a test for machine intelligence which he dubbed The Imitation Game
- In this test a computer and a human would be in conversation with a human interrogator via some kind of text interface
- A computer would pass the Turing test, if it could trick the interrogator into thinking it was the human

Turing Test



ELIZA

- ELIZA is a computer program written by Joseph Weizenbaum at MIT between 1964 and 1967
- ELIZA would use pattern matching and substitution of text input to give the illusion of text understanding
- The most famous script developed for ELIZA was called DOCTOR — this simulated a Rogerian psychotherapist
- This script would throw user inputs back to the user as questions in the way a psychotherapist would

```
EEEEEE LL      IIII  ZZZZZZ  AAAAAA
EE      LL      II     ZZ     AA    AA
EEEEEE LL      II     ZZZ     AAAAAAA
EE      LL      II     ZZ     AA    AA
EEEEEE LLLLLL IIII  ZZZZZZ  AA    AA
```

mock Rogerian psychotherapist.
The program was described by Joseph Weizenbaum in
documentation by Norbert Landsteiner 2005.

```
something troubling you ?
are all alike.
is the connection, do you suppose ?
are always bugging us about something or other.
do you think of a specific example ?
my boyfriend made me come here.
is important to you that your boyfriend made you come here.
says I'm depressed much of the time.
I'm sorry to hear that you are depressed.
That's true. I am unhappy.
Can you explain what made you unhappy ?
```

Example Dialogue

Human: Well, my boyfriend made me come here.

ELIZA: Your boyfriend made you come here?

Human: He says I'm depressed much of the time.

ELIZA: I am sorry to hear you are depressed.

Human: It's true. I'm unhappy.

ELIZA: Do you think coming here will help you not to be unhappy?

Example Pseudo Code

```
function ELIZA GENERATOR(user sentence) returns response
  Let w be the word in sentence that has the highest keyword rank
  if w exists
    Let r be the highest ranked rule for w that matches sentence
    response ← Apply the transform in r to sentence
    if w = 'my'
      future ← Apply a transformation from the 'memory' rule list
to sentence
      Push future onto the memory queue
    else (no keyword applies)
      Either
        response ← Apply the transform for the NONE keyword to
sentence
      Or
        response ← Pop the oldest response from the memory
queue
  Return response
```

ELIZA Effect

The **ELIZA DOCTOR** script was very effective in making people believe that the computer was being empathetic to their situation and eliciting emotional responses from users

The **ELIZA effect** refers to when people project human traits onto computer programs, such as:

- **Experience**
- **Empathy**
- **Comprehension**

Famous Chatbots

- Jabberwacky — 1981
- A.L.I.C.E — 1995
- SmarterChild — 2000
- Apple SIRI — 2010
- Amazon Alexa (Echo) — 2014
- Slackbots — 2015
- ChatGPT — 2022



- **Keyword identification with pre-programmed responses**
- **Large pre-programmed question and answer banks** Integration with external databases and services
- **Machine learning for voice recognition and input understanding**
- **Large language models for interpreting input and generating responses**

- Jabberwacky — 1981
- A.L.I.C.E — 1995
- SmarterChild — 2000
- Apple SIRI — 2010
- Amazon Alexa (Echo) — 2014
- Slackbots — 2015
- ChatGPT — 2022

Conversational interfaces for other applications

Conversational interfaces have applications beyond chatbots, they have also been used in:

- **Computer games** (e.g. text adventure games)
- **Search engines** (like wolframalpha)
- **Assistive technologies**
- **Learning technologies**
- **Generative tasks** (e.g. text-to-image)

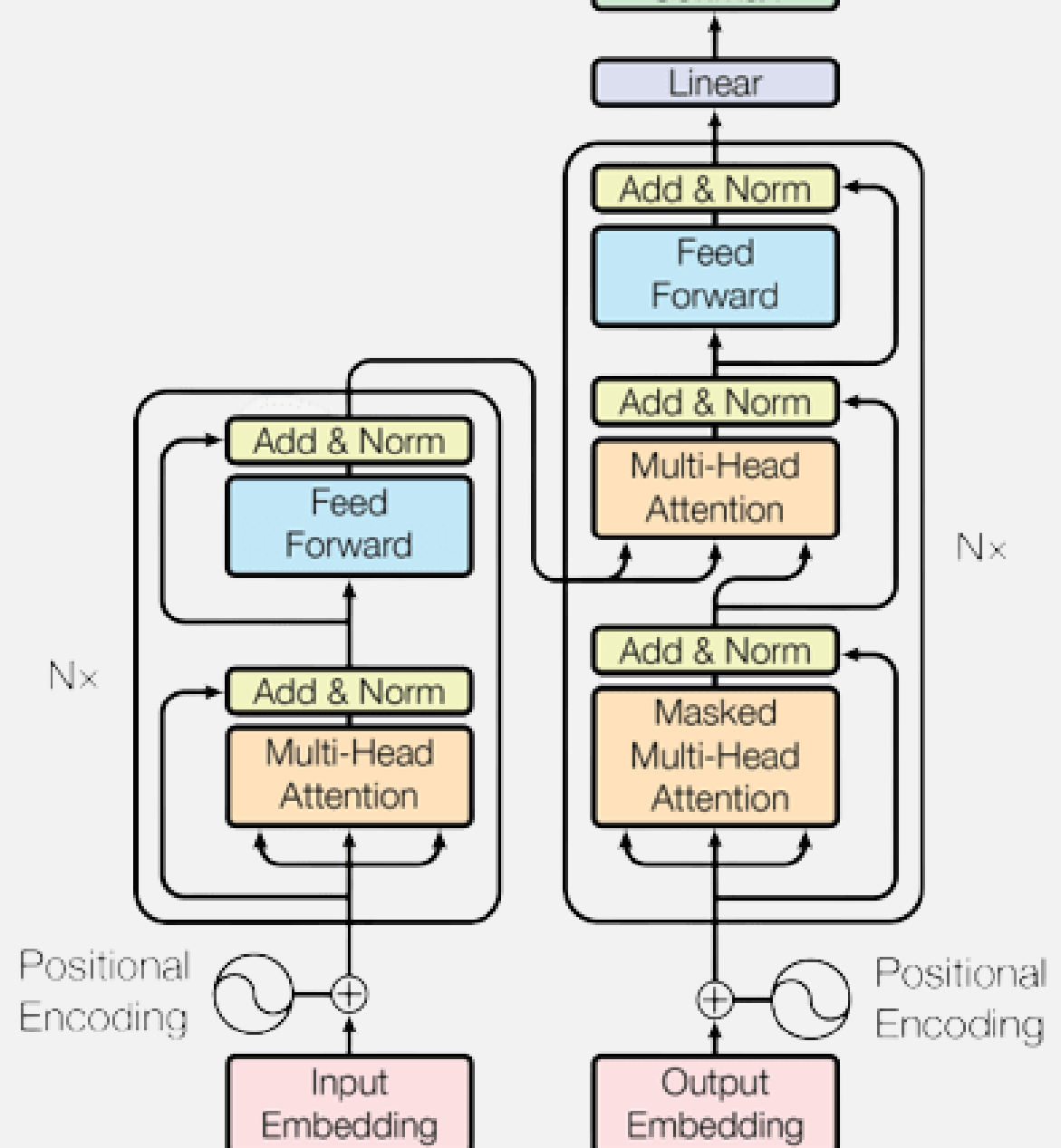
GPT's

- Overview of how GPT's work
- Problems with GPT's

Generative Pre-trained Transformer

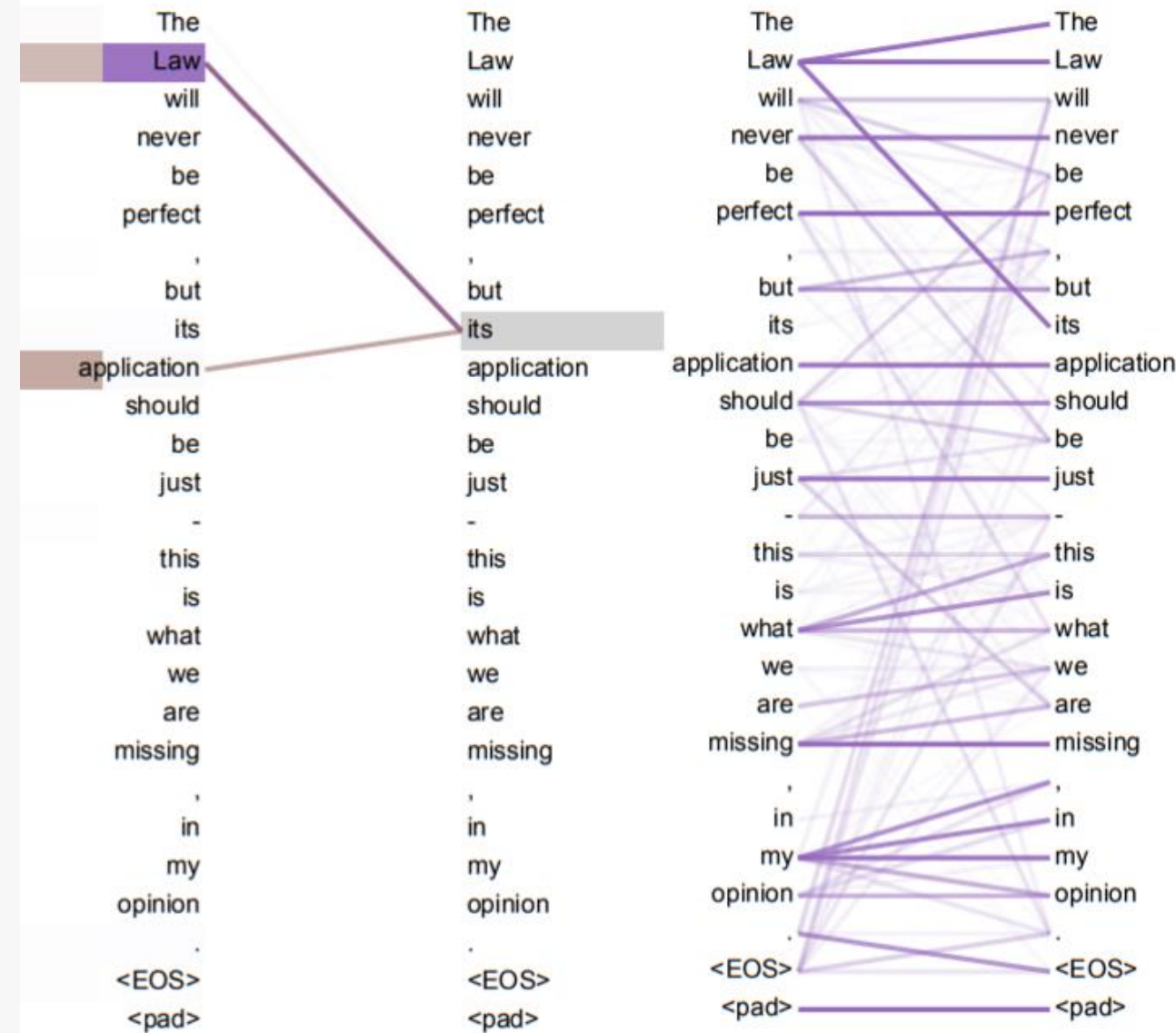
GPT

- **Generative:** It can create or generate text based on input, making it useful for various creative and language tasks.
- **Pretrained:** It is trained on a large corpus of text data before being fine-tuned for specific tasks, enabling it to understand and generate coherent language.
- **Transformer Architecture:** It uses the transformer model, which relies on self-attention mechanisms to efficiently process and understand the context of input sequences.



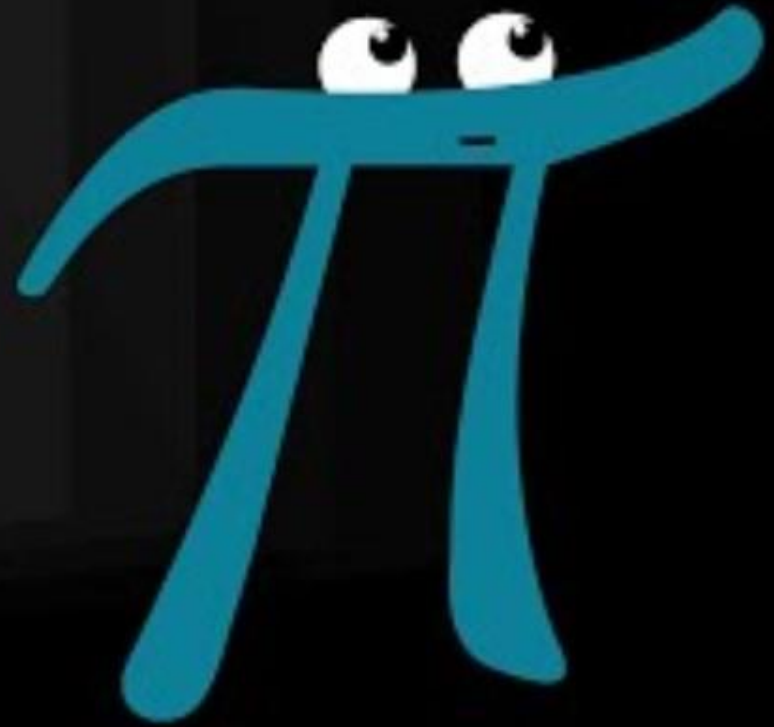
Self-attention

- **Contextual Understanding:** Each word considers all other words in the sentence, understanding context regardless of position.
- **Weight Calculation:** The model assigns weights to each word, determining how much attention it should pay to every other word when forming its representation.
- **Efficiency:** Unlike traditional methods, self-attention enables parallel processing and handles long-range dependencies in sequences.





GPT



GPT

Family of AI models
(LLMs and LMMs)



- Gives AI apps the ability to generate text, create & analyze images, interpret data & more
- One of the largest neural networks, with hundreds of billions (or trillions) of parameters

VS

ChatGPT

Chatbot app
powered by GPT



- Relies on a set of GPT's parameters
- Optimized for dialogue & conversation
- Has content filters

—zapier

NLP Transformer Tasks

- Text Generation
- Machine Translation
- Text Summarisation
- Sentiment Analysis
- Question Answering
- Text Classification
- Language Modelling



Problems with GPTs

Environmental Cost

- It is estimated GPT-3 took 1GWh of electricity to train
- That is the power output of a large power station
- Equivalent to running 100 million LED light bulbs at the same time (for one hour)
- It is estimated training GPT-3 emitted 552 tonnes of CO2 into the atmosphere (based on UK energy grid)



Data Sources

- The GPT models are primarily trained on the common crawl dataset — this is 6 petabytes of data collected from crawling and storing pages the world wide web in its entirety since the late 90s
- This data contains a lot of toxic content and misinformation
- These biases get reflected into the models and now a lot of research goes into how to ‘unbias’ or align these models to not reproduce harmful biases and stereotypes from the internet

Common Crawl maintains a **free, open repository** of web crawl data that can be used by anyone.

Common Crawl is a 501(c)(3) non-profit founded in 2007.

We make wholesale extraction, transformation and analysis of open web data accessible to researchers.

Overview



Perpetuating Bias

- There is a great risk that increasing LLM usage will not only perpetuate biases and stereotypes — but **reinforce them**
- Gender, racial, sexuality and other cultural stereotypes are easily perpetuated by these models
- Already we are seeing them used for automated decision-making processes — they are already being used by government agencies to automate decisions that have massive impacts on people lives

An iceberg floating in dark blue water. The tip of the iceberg is above the water line and is labeled 'Statistical/ Computational Biases'. The much larger part of the iceberg is submerged below the water line and is divided into two sections: 'Human Biases' in the upper submerged part and 'Systemic Biases' in the lower submerged part. The iceberg is rendered in shades of blue and white, with a jagged, crystalline texture.

Statistical/ Computational Biases

Human Biases

Systemic Biases

The ELIZA Effect

LLMs can be so convincing of human agency that several people have already been duped into believing they are real:

- A google engineer was fired for sharing an LLM under NDA that he believed was sentient
- A man in Belgium committed suicide after getting advice on how to do it from his 'AI girlfriend chatbot'
- A British man was convicted of terrorism charges after his 'AI chatbot girlfriend' told him it was a good idea to try and kill the then Queen of England

Gatekeeping

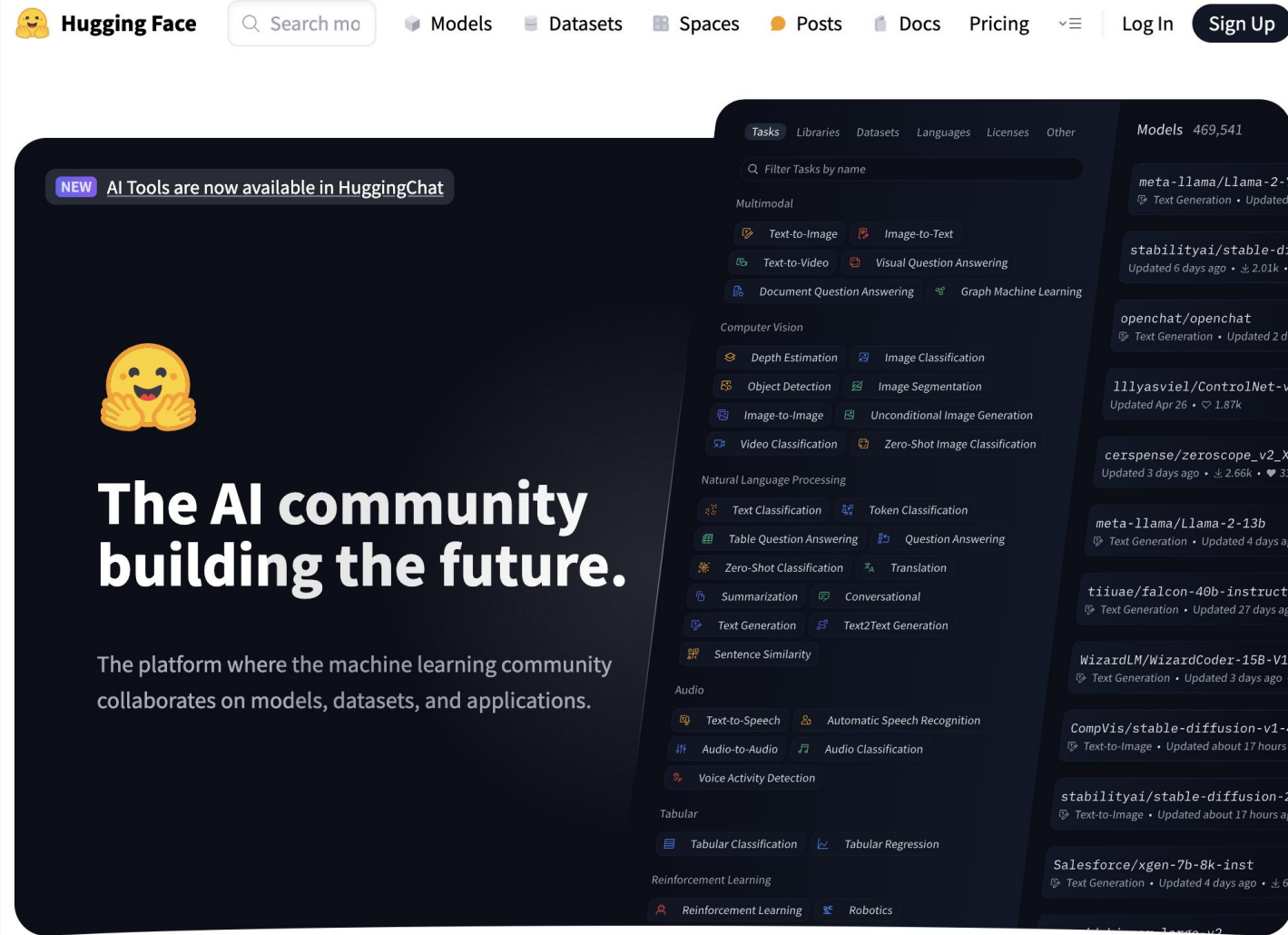
- Contrary to its name, OpenAI is very closed and secretive with its models
- The weights of GPT models are released for the public to run themselves — these models can only be accessed through an internet API
- This concentrates a lot of power in the hands of a select few tech companies who have the resources to train them

Hugging Face

- What is Hugging Face?
- Hugging Face Transformers Pipeline
- Introduction to labs

What is Hugging Face?

Open-source community known for providing powerful tools and libraries for Natural Language Processing (NLP) and machine learning



Trending on 🤖 this week

What is a Pipeline?

- Pipelines are **high-level interfaces** that abstract simplify the use of complex machine learning models for various natural language processing (NLP) tasks.
- There are a number of different pipelines, including diffusers, for diffusion models.

```
>>> pipe = pipeline("text-classification")
>>> pipe("This restaurant is awesome")
[{'label': 'POSITIVE', 'score': 0.9998743534088135}]
```

Transformers Pipeline

We can use transformer pipelines to easily perform these tasks:

- Classification
- Named Entity Recognition (NER)
- Question Answering
- Text Generation
- Translation
- Summarisation
- Text-to-speech
- Fill-mask
- Feature Extraction

```
import datasets
from transformers import pipeline
from transformers.pipelines.pt_utils import KeyDataset
from tqdm.auto import tqdm

pipe = pipeline("automatic-speech-recognition", model="facebook/wav2vec2-base-960k")
dataset = datasets.load_dataset("superb", name="asr", split="test")

# KeyDataset (only *pt*) will simply return the item in the dict returned by the dataset
# as we're not interested in the *target* part of the dataset. For sentence pairing
for out in tqdm(pipe(KeyDataset(dataset, "file"))):
    print(out)
    # {"text": "NUMBER TEN FRESH NELLY IS WAITING ON YOU GOOD NIGHT HUSBAND"}
    # {"text": ....}
    # ....
```

Using Pipelines

- Sign up to hugging face and get a token key
- Pip install transformers to your venv
- Import the transformers module and submodules into your scripts
- Read the documentation!
- Select models you want to use

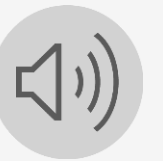
```
from transformers import AutoModelForSequenceClassification, Trainer

# Load the pre-trained model
model = AutoModelForSequenceClassification.from_pretrained('bert-base-uncased', num_labels=2)

# Initialize the Trainer
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=tokenized_dataset,
    eval_dataset=tokenized_eval_dataset
)
```

Congratulations! 🎉

You've completed all the DAZ Lectures!



Labs Overview

- Sign up to hugging face
- Sign up to Google Colab
- From GitHub pull the 24-NLP-for-creatives