An Open Source journey

THE LIMITS OF IMAGINATION

Aroma Rodrigues

AROMA RODRIGUES

Python Enthusiast, coding since 2014, it will be a decade next year. Yay! Multi-Pycon speaker engagements including Pycon US, Estonia, Lithuania, Hong Kong, Sweden, Europython, PyOhio, PyConZA, India and FOSSASIA. On the Natural Language Processing train since 2018. I do fun experiments, sometimes social with data, python and language models because I believe as do most multinational companies now, that the human condition is encoded in language, just as science in math, and it in inevitable that one day, we will be using computers to help us linguistically as they do mathematically.

STORIES

I

PyCon India 2018

Terms and Conditions Summarizer

II

PyCon-ZA 2019

NLP Fake News Detector

III

PyCon US 2024

Only Bad Demos in the Building

PyCon India 2018
Terms and Conditions Summarizer

TERMS AND CONDITIONS

🧾 Why: Drowning in unreadable legalese across emails, apps, and websites

🛠️ What: NLP pipeline to extract obligations, permissions, and risks from T&C docs

⚖️ Impact: Empowered users to skim legal agreements with context—not confusion

FIRST PRINCIPLES

🧾 Problem: No existing dataset; legalese buried in walls of unreadable text

🛠️ Approach: Built a labeled dataset from scratch using first principles of contract structure

⚖️ Outcome: NLP pipeline to extract obligations, risks, and user rights from legal docs

RESULTS

🧾 Problem: No usable dataset; legal text was dense, varied, and inaccessible

🛠️ Approach: Created an SNLI-style dataset from scratch using first principles

⚖️ Result: Early-stage NLP model with limited accuracy—but a crucial starting point; today's version could be vastly improved with synthetic data

PyCon US 2024

Only Bad Demos in the Building

“These are the things I wanted answers to…”

🕵️ Will forensic portrait artists keep their jobs?

✍️ Can AI write my travelogue better than me?

👗 Can I build a Clueless-style fashion matcher?

📚 How do I generate clean, non-duplicate synthetic data?

PyCon US 2024

Only Bad Demos in the Building

EXISTENTIAL CRISIS
STOICISM
PHILOSOPHY
PHILOSOPHICAL QUESTIONS

PyCon US 2024

Only Bad Demos in the Building

Who gets to keep their jobs in the post-ai world?

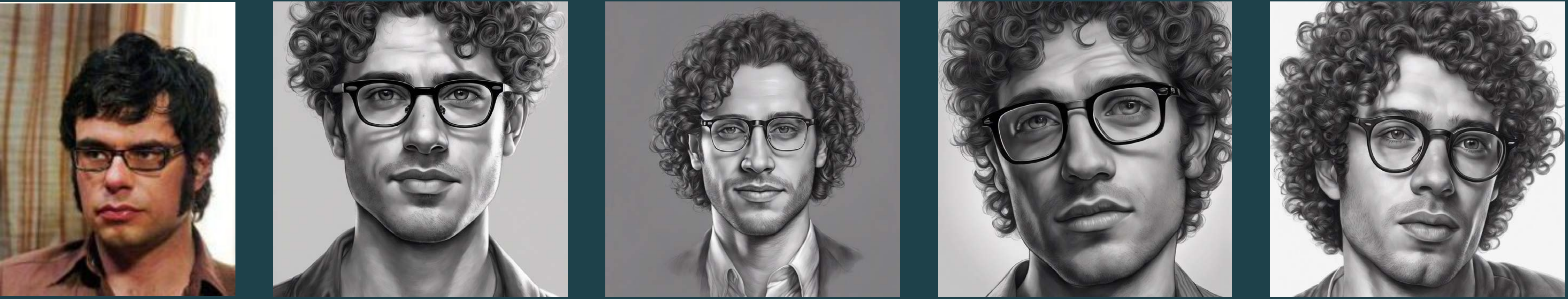
Will Forensic portrait makers keep their jobs or not?

PyCon US 2024

011447.jpg

A man with curly hair, dark eyes, small nose and thin lips wearing glasses

5_o_Clock_Shadow Bags_Under_Eyes Big_Nose Black_Hair Bushy_Eyebrows Eyeglasses Male Mustache Wavy_Hair Young

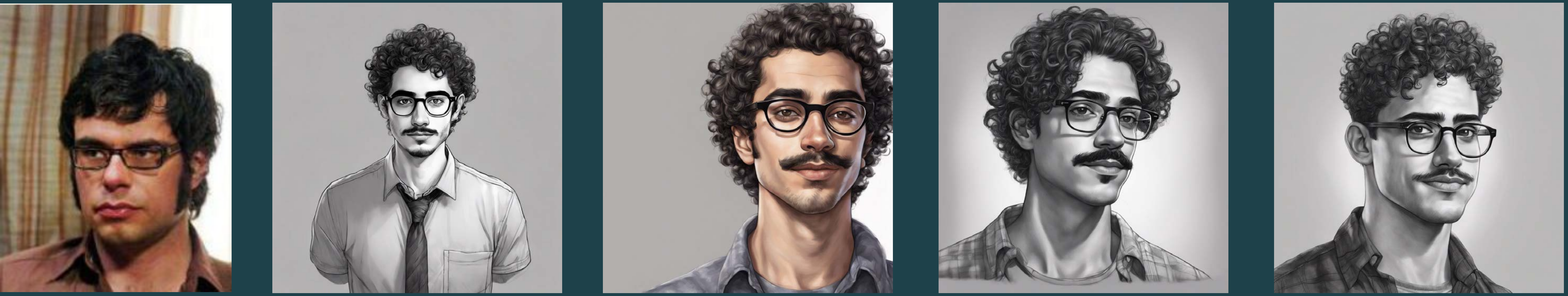


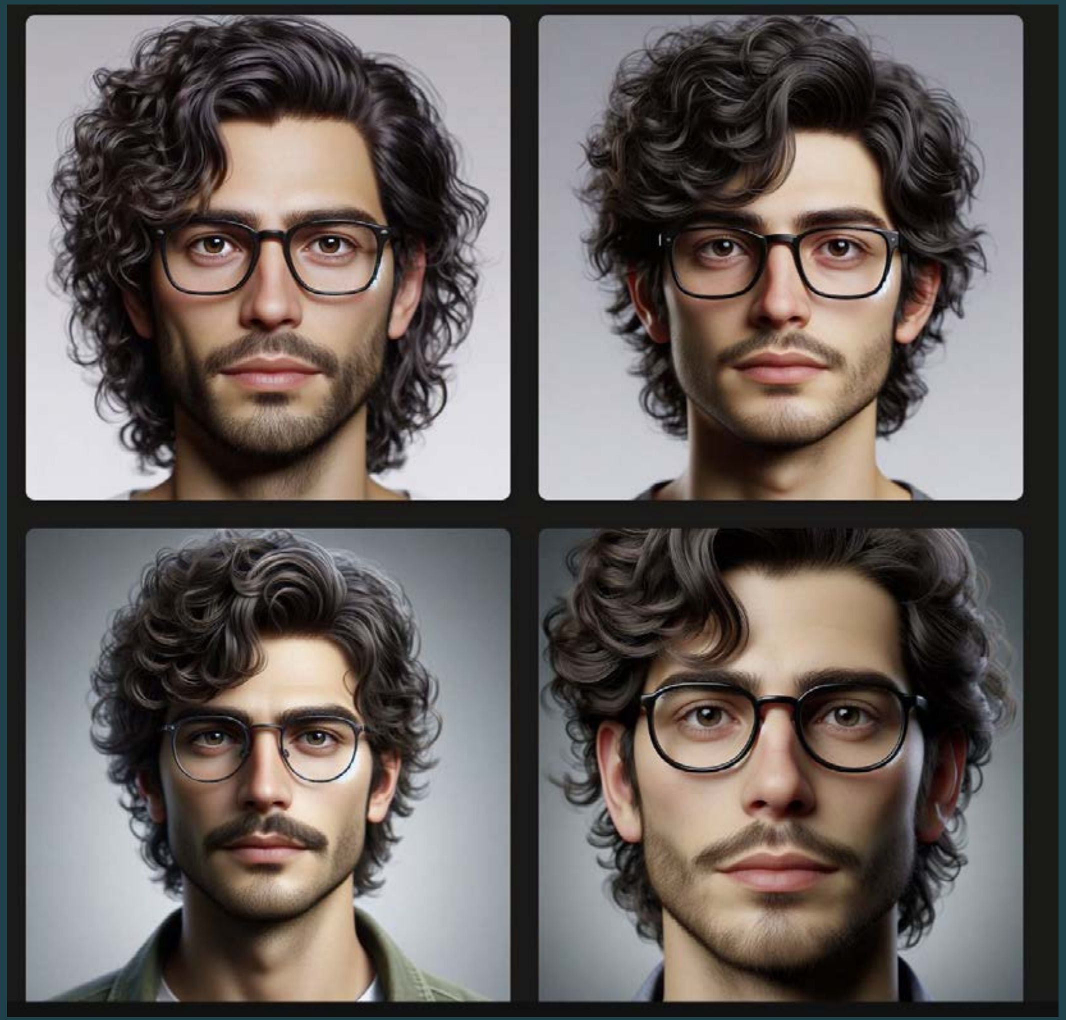
PyCon US 2024

011447.jpg

A man with curly hair, dark eyes, small nose and thin lips wearing glasses

5_o_Clock_Shadow Bags_Under_Eyes Big_Nose Black_Hair Bushy_Eyebrows Eyeglasses Male Mustache Wavy_Hair Young

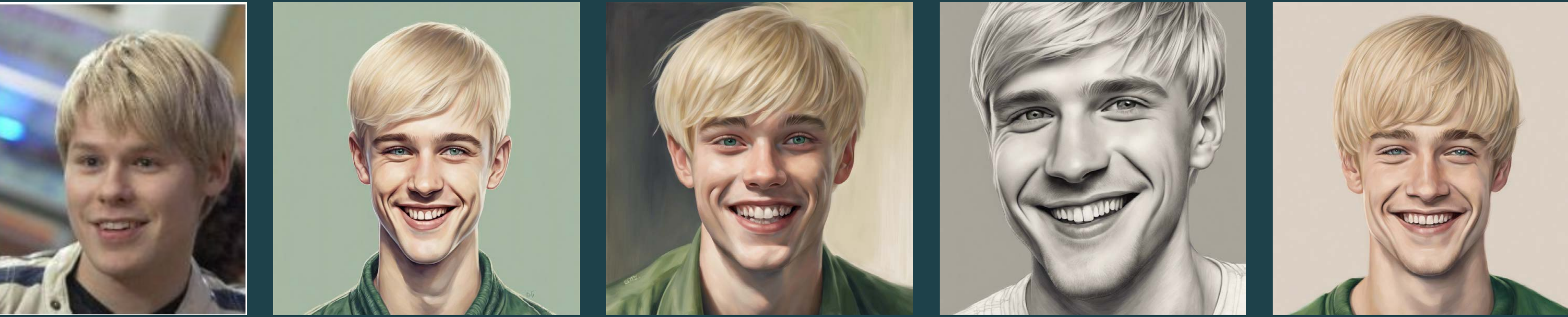




PyCon US 2024

049959.jpg

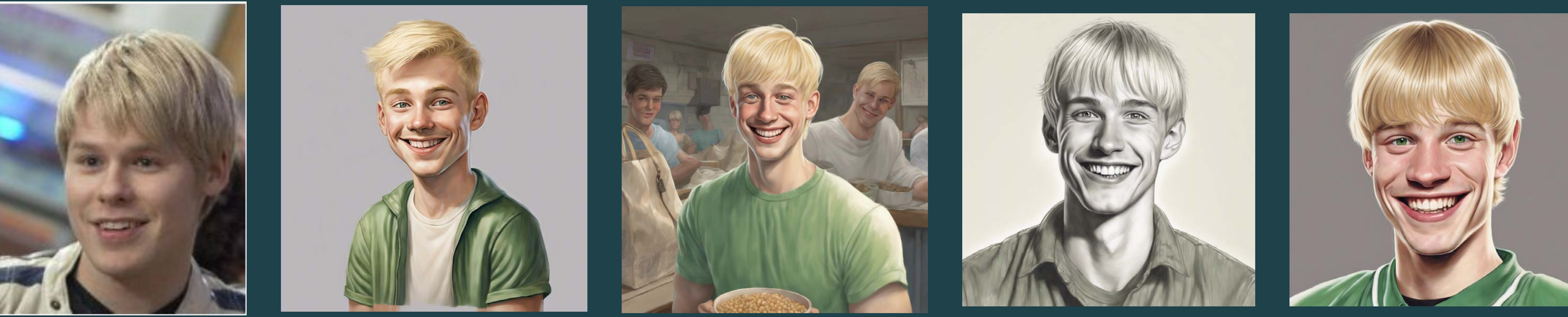
A man with light blonde hair, dark eyes and a side smile. A young boy with blonde hair and a fringe. Hie eyes are dark and small and his lips are thin. His upper teeth are visible. He is smiling and he seems excited. A smiling pale white man with a blonde bowl-haircut , green eyes



PyCon US 2024

049959.jpg

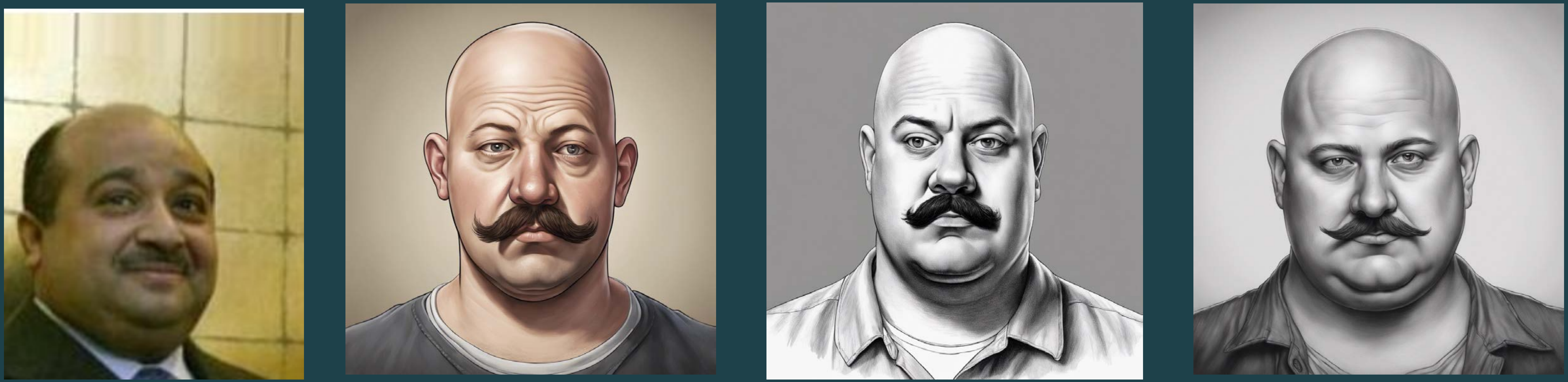
A man with light blonde hair, dark eyes and a side smile. A young boy with blonde hair and a fringe. Hie eyes are dark and small and his lips are thin. His upper teeth are visible. He is smiling and he seems excited. A smiling pale white man with a blonde bowl-haircut , green eyes



PyCon US 2024

057757.jpg

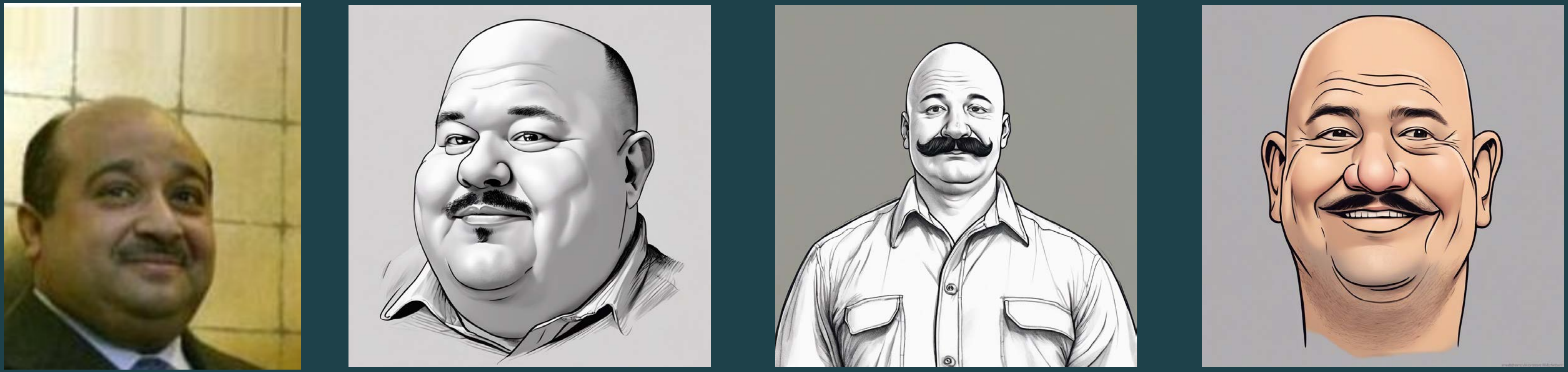
A bald man with hair on the sides of his head and a double chin. He has a round face, a wide nose, small ears and a moustache. A fat man with barely any hair and a mustache

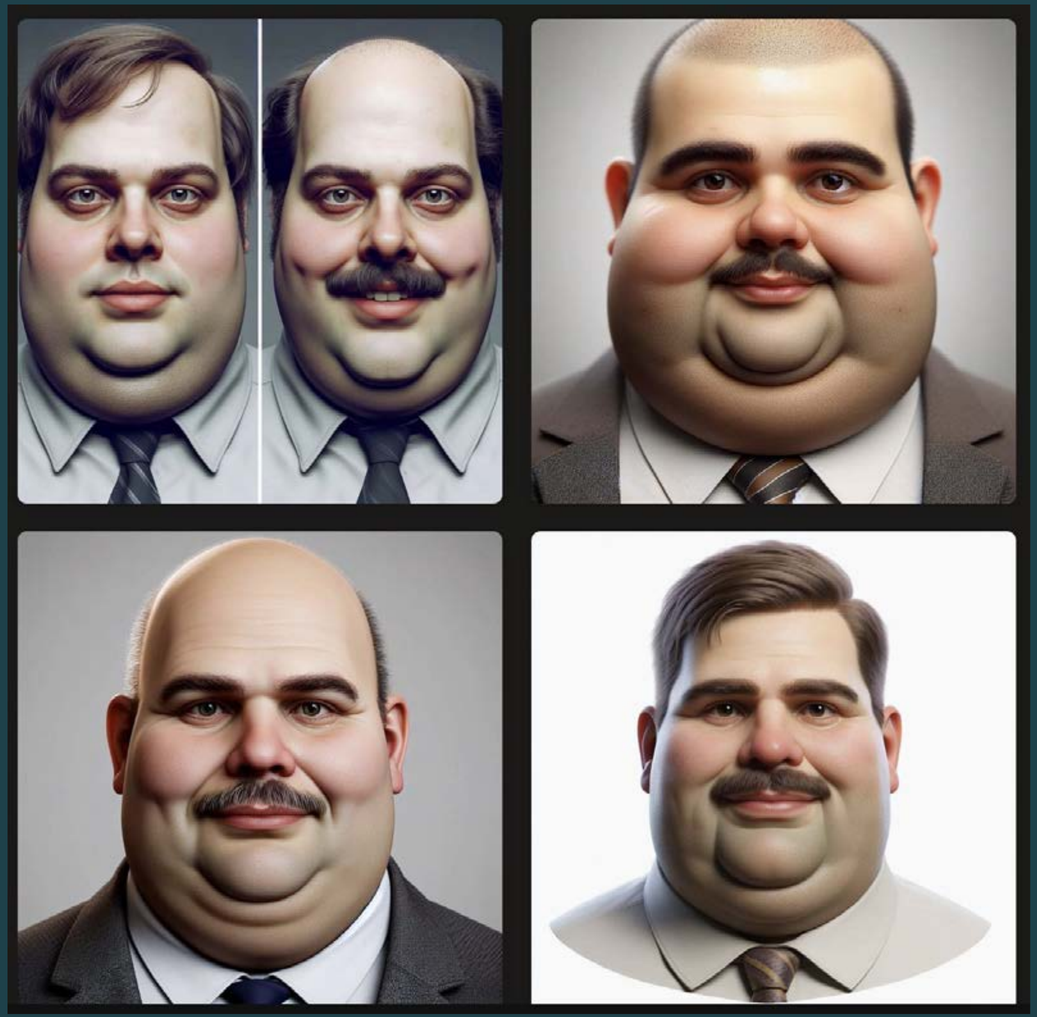


PyCon US 2024

057757.jpg

A bald man with hair on the sides of his head and a double chin. He has a round face, a wide nose, small ears and a moustache. A fat man with barely any hair and a mustache

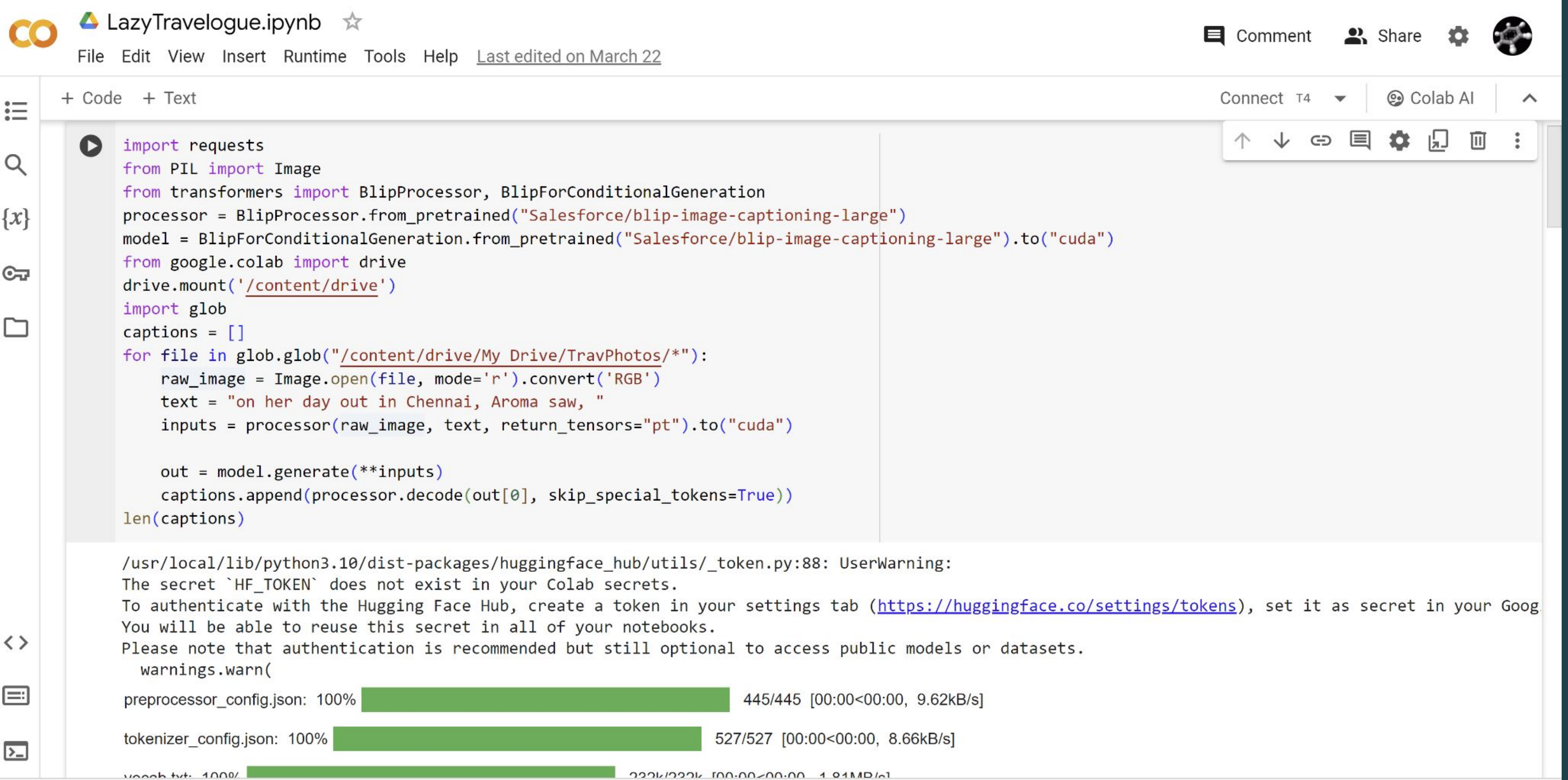




WHAT DO I DO THEN?

DREAM PROJECT : WRITE MY OWN TRAVELOGUE BUT - LOGISTICAL ISSUES FOR EG. TOOK 300 odd photos of a day out in CHENNAI

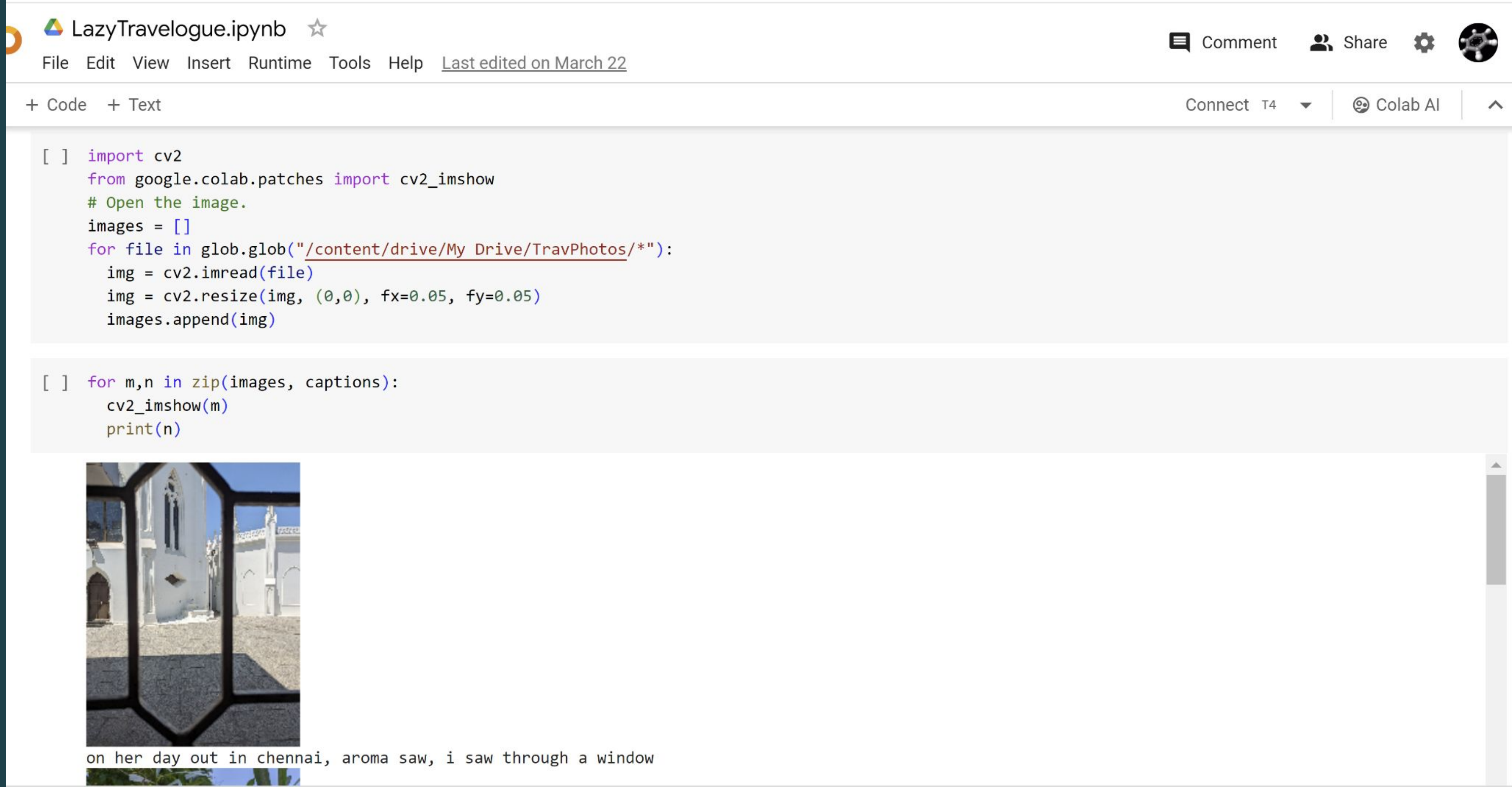
MADE AI DO IT!


LazyTravelogue.ipynb
File Edit View Insert Runtime Tools Help Last edited on March 22
Comment
Share
+ Code + Text
Connect T4
Colab AI

```python
import requests
from PIL import Image
from transformers import BlipProcessor, BlipForConditionalGeneration
processor = BlipProcessor.from_pretrained("Salesforce/blip-image-captioning-large")
model = BlipForConditionalGeneration.from_pretrained("Salesforce/blip-image-captioning-large").to("cuda")
from google.colab import drive
drive.mount('/content/drive')
import glob
captions = []
for file in glob.glob("/content/drive/My Drive/TravPhotos/*"):
    raw_image = Image.open(file, mode='r').convert('RGB')
    text = "on her day out in Chennai, Aroma saw, "
    inputs = processor(raw_image, text, return_tensors="pt").to("cuda")

    out = model.generate(**inputs)
    captions.append(processor.decode(out[0], skip_special_tokens=True))
len(captions)
```

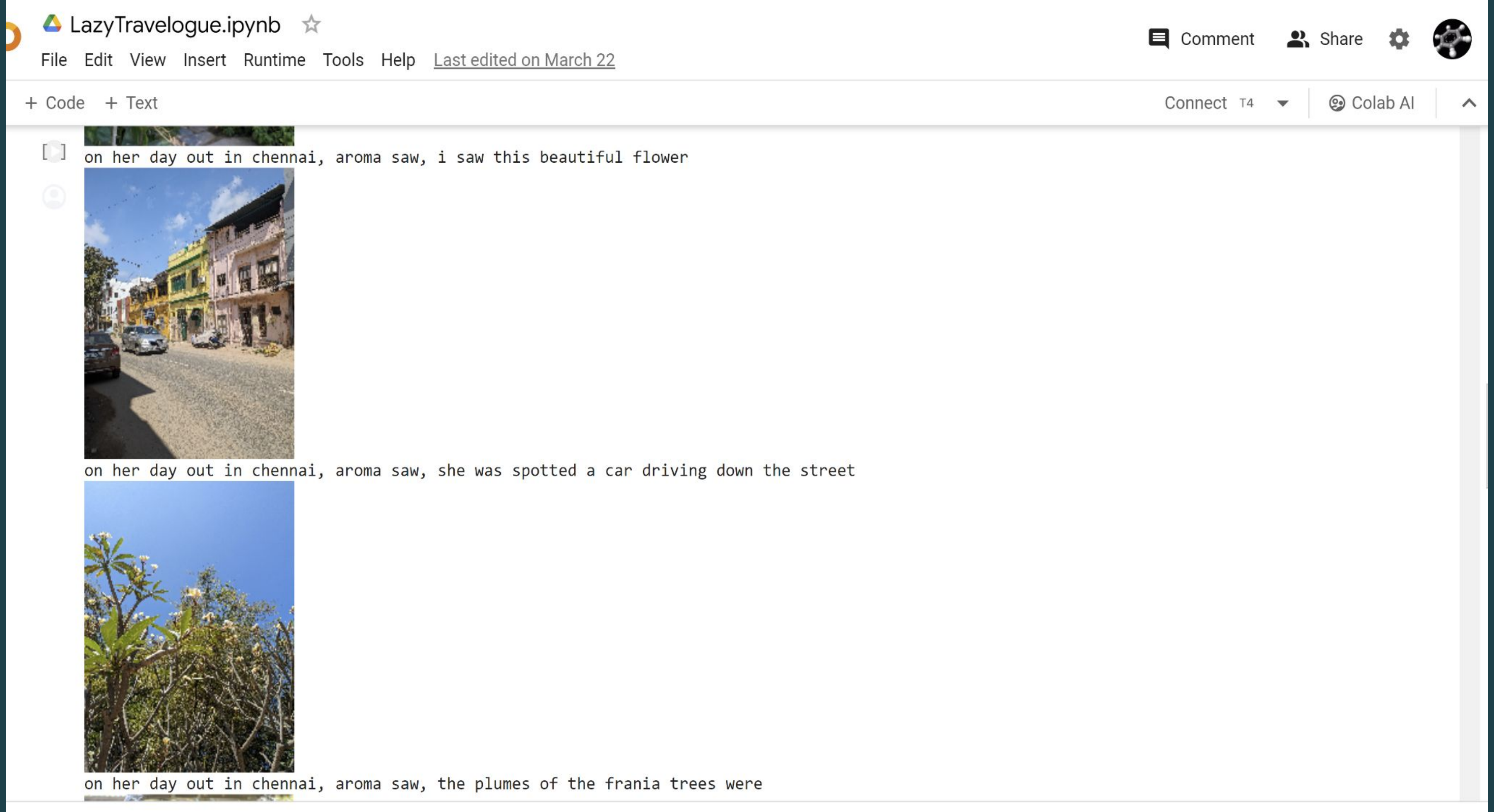
```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_token.py:88: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as secret in your Goog
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
preprocessor_config.json: 100%  445/445 [00:00<00:00, 9.62kB/s]
tokenizer_config.json: 100%  527/527 [00:00<00:00, 8.66kB/s]
```


LazyTravelogue.ipynb
File Edit View Insert Runtime Tools Help Last edited on March 22
Comment Share
+ Code + Text
Connect T4
Colab AI

```
import cv2
from google.colab.patches import cv2_imshow
# Open the image.
images = []
for file in glob.glob("/content/drive/My Drive/TravPhotos/*"):
  img = cv2.imread(file)
  img = cv2.resize(img, (0,0), fx=0.05, fy=0.05)
  images.append(img)
```

```
for m,n in zip(images, captions):
  cv2_imshow(m)
  print(n)
```

```
on her day out in chennai, aroma saw, i saw through a window
```

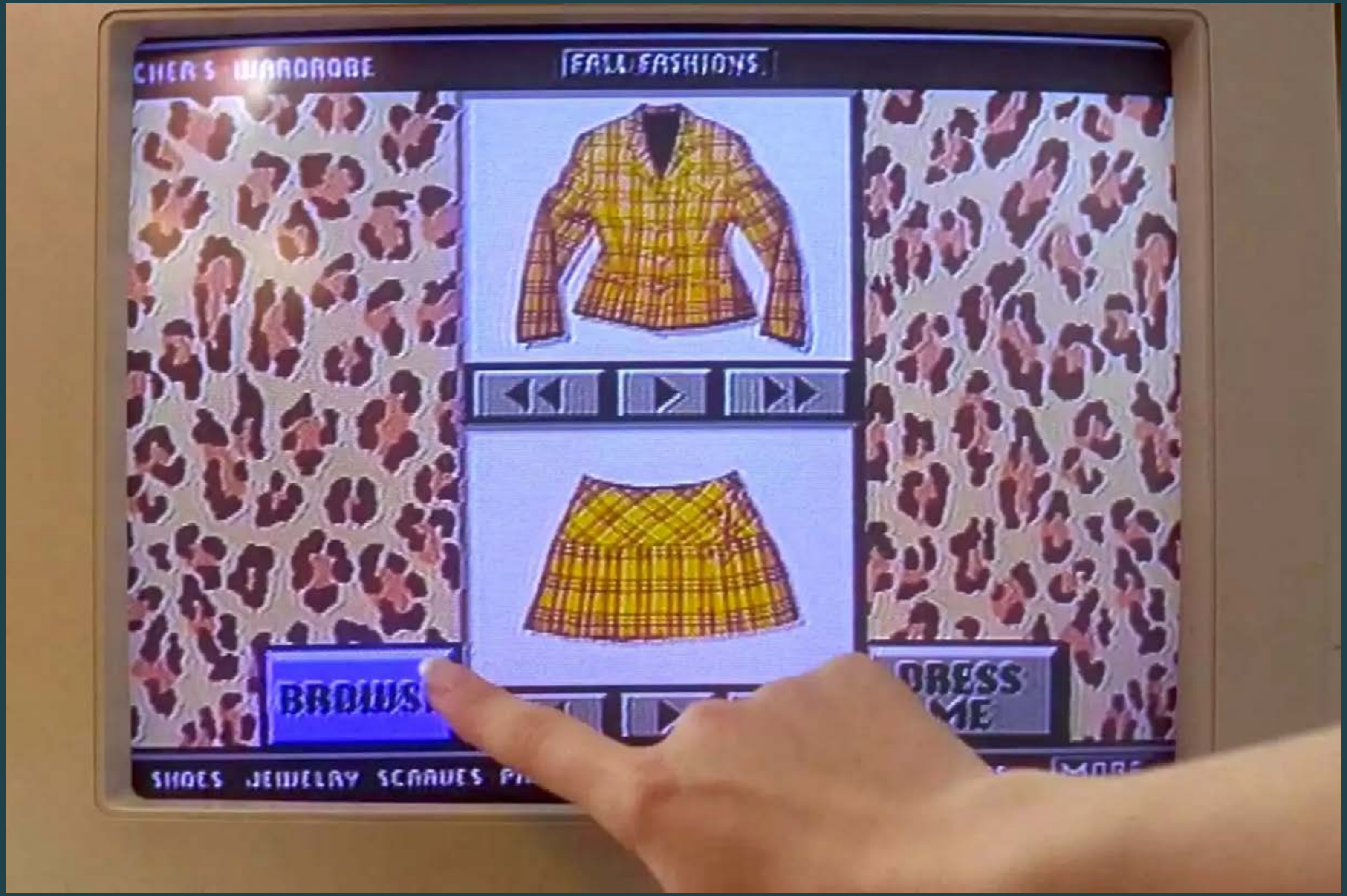


```
on her day out in chennai, aroma saw, i saw this beautiful flower
```

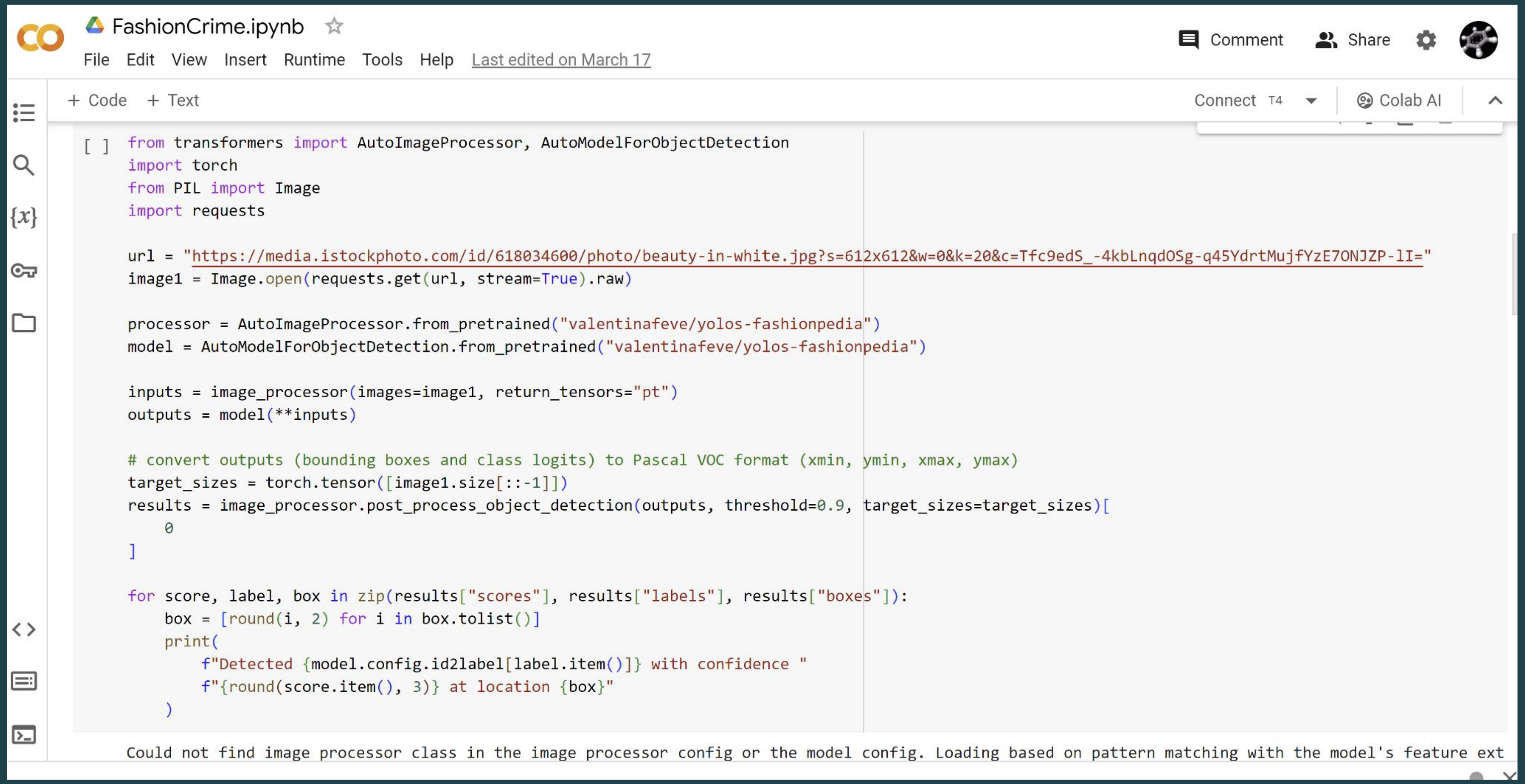
```
on her day out in chennai, aroma saw, she was spotted a car driving down the street
```

```
on her day out in chennai, aroma saw, the plumes of the frania trees were
```

Watch a movie
Clueless


CHER'S WARDROBE
FALL FASHIONS
BROWSE
DRESS ME
SHOES
JEWELRY
SCARVES
MORE

CAN AI DO IT?



```
from transformers import AutoImageProcessor, AutoModelForObjectDetection
import torch
from PIL import Image
import requests

url = "https://media.istockphoto.com/id/618034600/photo/beauty-in-white.jpg?s=612x612&w=0&k=20&c=Tfc9edS_-4kbLnqdOSg-q45YdrtMujfYzE7ONJZP-lI="
image1 = Image.open(requests.get(url, stream=True).raw)

processor = AutoImageProcessor.from_pretrained("valentinafeve/yolos-fashionpedia")
model = AutoModelForObjectDetection.from_pretrained("valentinafeve/yolos-fashionpedia")

inputs = image_processor(images=image1, return_tensors="pt")
outputs = model(**inputs)

# convert outputs (bounding boxes and class logits) to Pascal VOC format (xmin, ymin, xmax, ymax)
target_sizes = torch.tensor([image1.size[::-1]])
results = image_processor.post_process_object_detection(outputs, threshold=0.9, target_sizes=target_sizes)[
    0
]

for score, label, box in zip(results["scores"], results["labels"], results["boxes"]):
    box = [round(i, 2) for i in box.tolist()]
    print(
        f"Detected {model.config.id2label[label.item()]} with confidence "
        f"{round(score.item(), 3)} at location {box}"
    )
```

```
Could not find image processor class in the image processor config or the model config. Loading based on pattern matching with the model's feature ext
```

🧠 Goal: Trace who's being blamed in misinformation, not just flag what's fake

💬 Dataset: Real WhatsApp forwards—informal, noisy, multilingual

🛠️ Method: Combined SciPy- and NLTK-based event models with POS tagging & dependency parsing—minimal training data required

PyCon-ZA 2019

NLP Fake News Detector

🚀 Improvement: Today, LLMs offer richer context, better understanding of blame dynamics, and scalable solutions

BuzzFeed: 22M interactions on top 50 fake stories (2018)

Knight Foundation: 10M+ tweets from 700k+ accounts linked to fake/conspiracy news

Reuters (India): 52% get news from WhatsApp; rumor-fueled violence has led to deaths

PyCon-ZA 2019

Impact of Fake News

Keyword Extraction & Verification
rake_nltk for extracting key phrases
Use News API to cross-check real coverage of keyword-based claims
Reverse image search for detecting photoshopped images

Content Verification
Compare articles from spoof sites vs mainstream media
Use of fact-checking platforms like Alt News, SMHoaxSlayer

PyCon-ZA 2019

Techniques for Detection

Textual Clues
Grammar and spelling mistakes
Overly positive or negative sentiment
No sources, or suspicious sources
Repetition of certain keywords

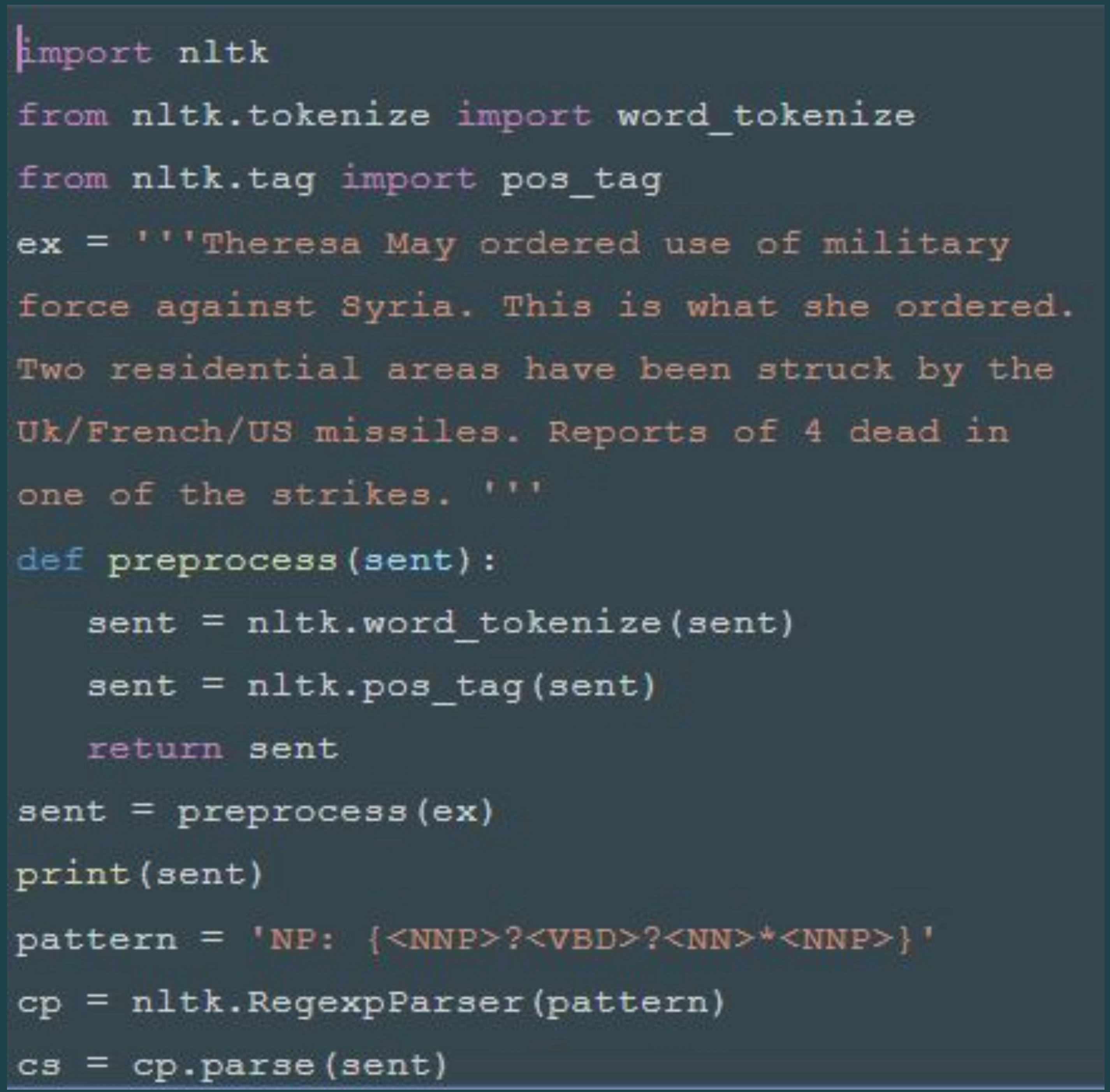
Syntactic Patterns: POS Tagging
Tagging statements to detect:
Blame assignment
Praise
Event causality
Active/Passive voice patterning using nltk.RegexpParser

Entity & Emotion Tracking
Track named politicians
Detect associated emotions: fear, hatred, sympathy

PyCon-ZA 2019

Techniques for Detection

🧱 Propaganda Structural Features
Path Model of Blame: Who did what, and why?
Causative Verbs: "ordered", "caused", "claimed"
Thresholding: % of sentences showing propaganda structure in the text



```
import nltk
from nltk.tokenize import word_tokenize
from nltk.tag import pos_tag
ex = '''Theresa May ordered use of military
force against Syria. This is what she ordered.
Two residential areas have been struck by the
Uk/French/US missiles. Reports of 4 dead in
one of the strikes. '''
def preprocess(sent):
   sent = nltk.word_tokenize(sent)
   sent = nltk.pos_tag(sent)
   return sent
sent = preprocess(ex)
print(sent)
pattern = 'NP: {<NNP>?<VBD>?<NN>*<NNP>}'
cp = nltk.RegexpParser(pattern)
cs = cp.parse(sent)
```

STORIES

IV

PyCon Estonia 2023
If your friends are bullshitting using LLMs

V

EuroPython 2022, PyCon Hong Kong 2021, PyCon Sweden 2021
How we're conditioned to believe the news is polarized

VI

PyOhio 2020 Analyzing bias in children's educational materials

PyCon Estonia 2023

If your friends are bullshitting using SNLI

1

🧠 Goal: Use NLP to spot contradictions in statements, proving when your friends are being inconsistent

2

🧩 Models: Compare BERT and GPT-2 embeddings for detecting contradiction

3

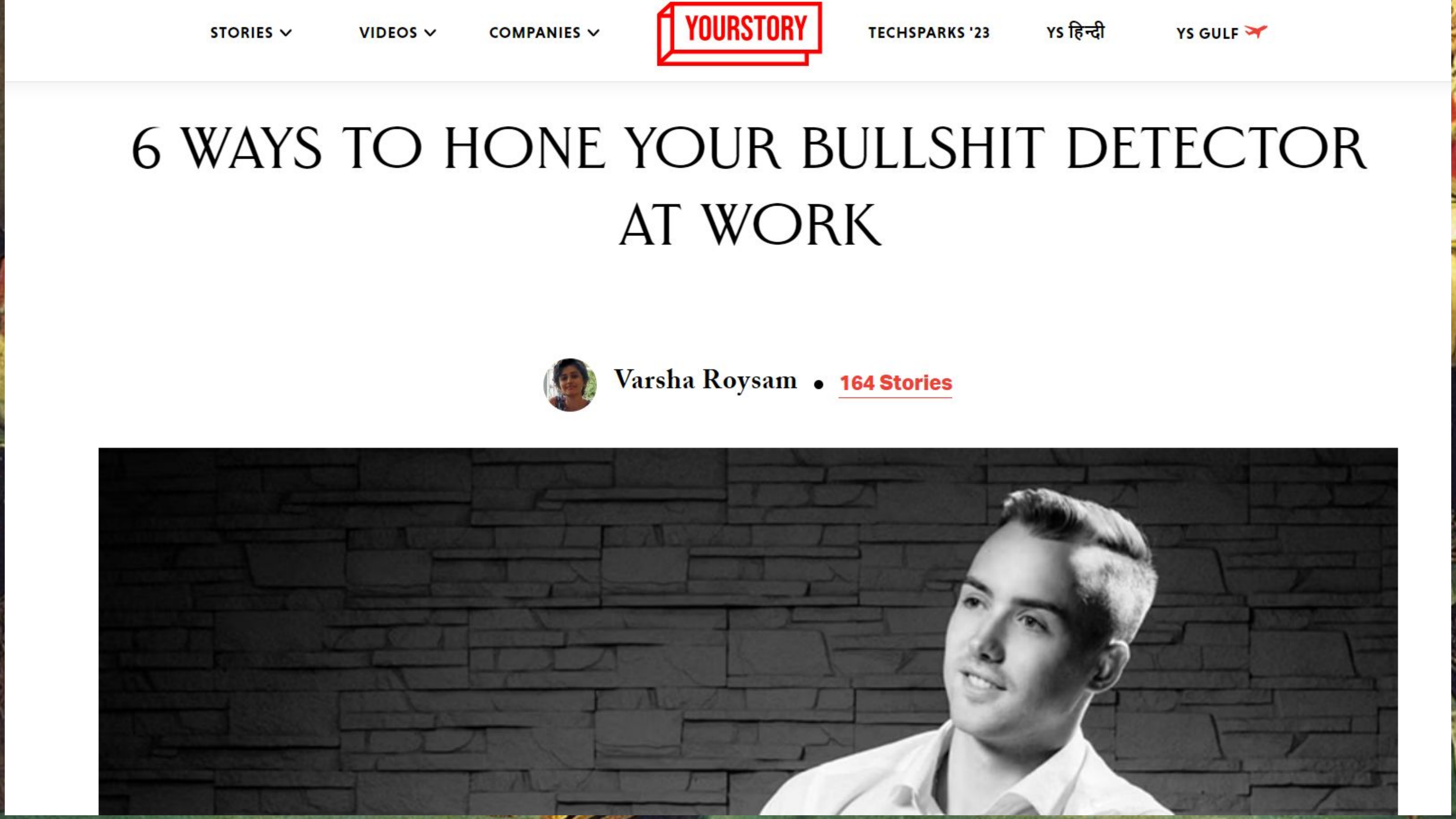
📝 Dataset: Leverage SNLI for building and training contradiction models

4

🌐 Impact: Improve LLM coherency—tackle hallucinations and filter fake/contradictory news

PyCon Estonia 2023

If your friends are bullshitting using SNLI

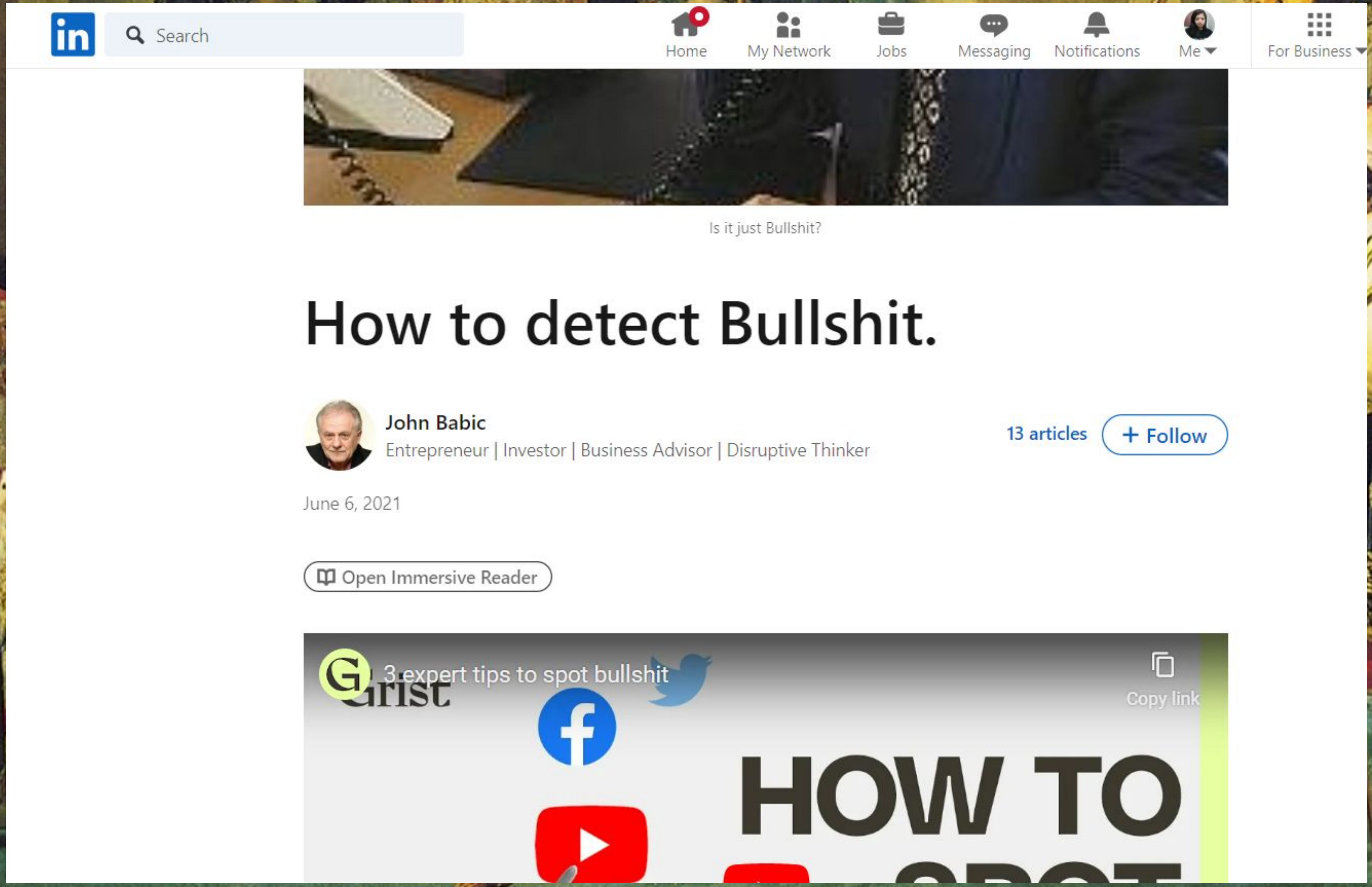


6 WAYS TO HONE YOUR BULLSHIT DETECTOR AT WORK

Varsha Roysam • 164 Stories

PyCon Estonia 2023

If your friends are bullshitting using SNLI



Is it just Bullshit?

How to detect Bullshit.

John Babic
Entrepreneur | Investor | Business Advisor | Disruptive Thinker

13 articles
+ Follow

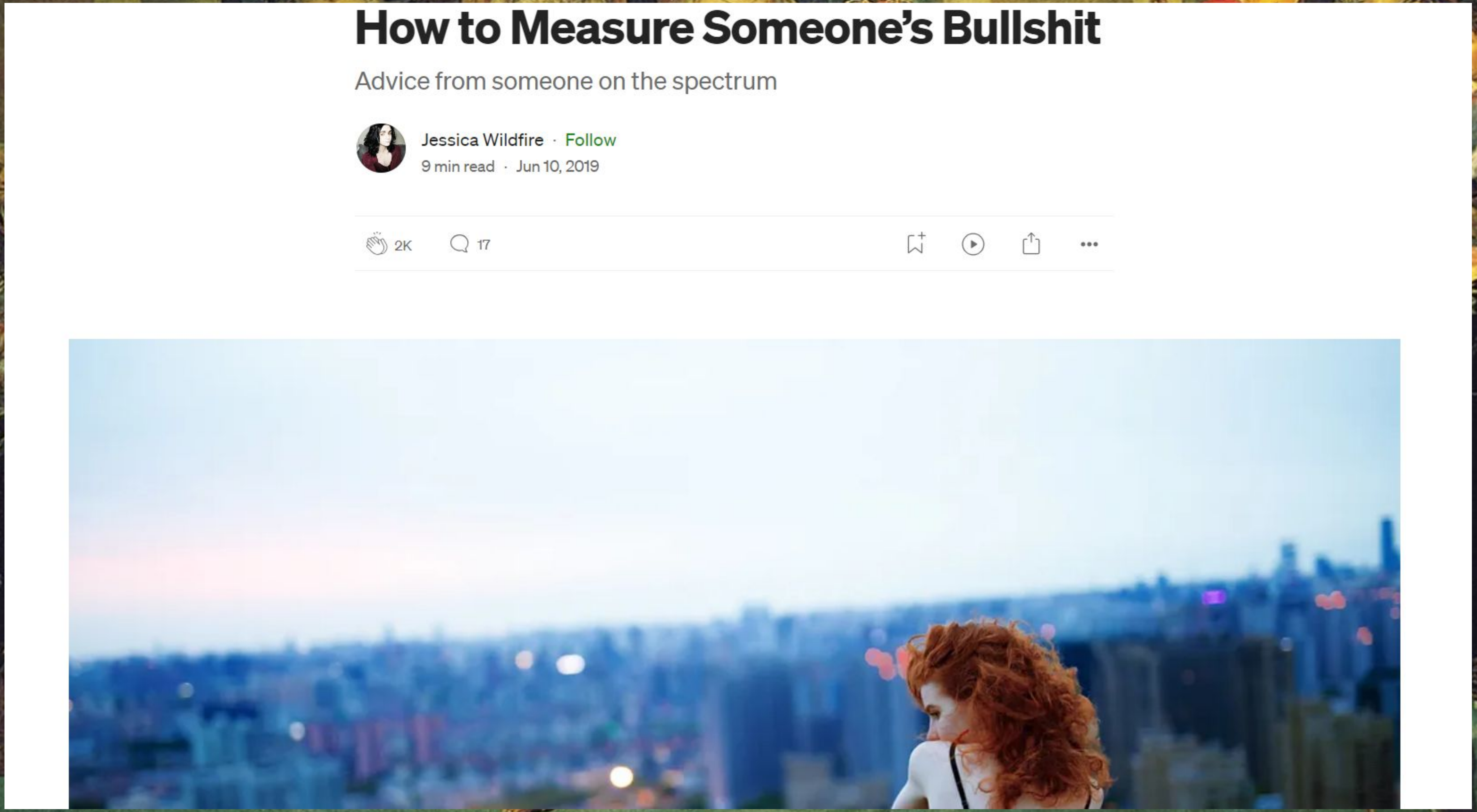
June 6, 2021

Open Immersive Reader

3 expert tips to spot bullshit

PyCon Estonia 2023

If your friends are bullshitting using SNLI



How to Measure Someone's Bullshit

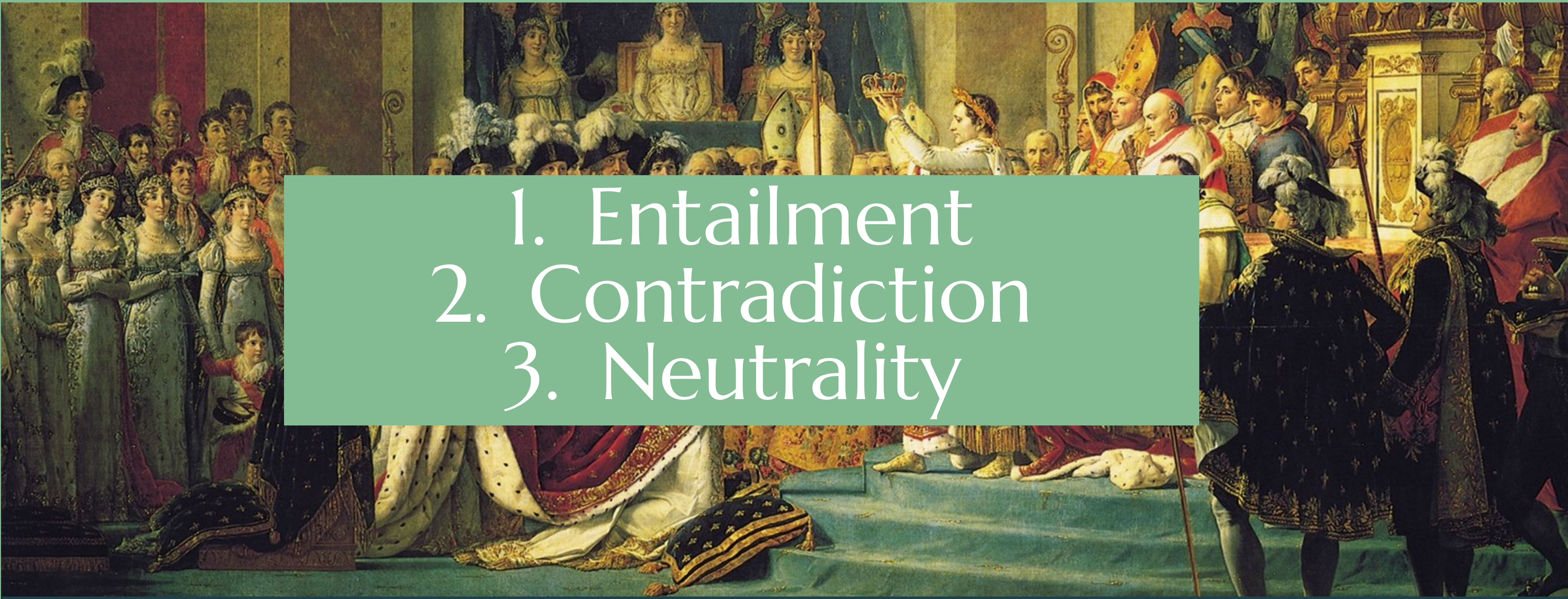
Advice from someone on the spectrum

Jessica Wildfire · Follow

9 min read · Jun 10, 2019

PyCon Estonia 2023

If your friends are bullshitting using SNLI



1. Entailment
2. Contradiction
3. Neutrality

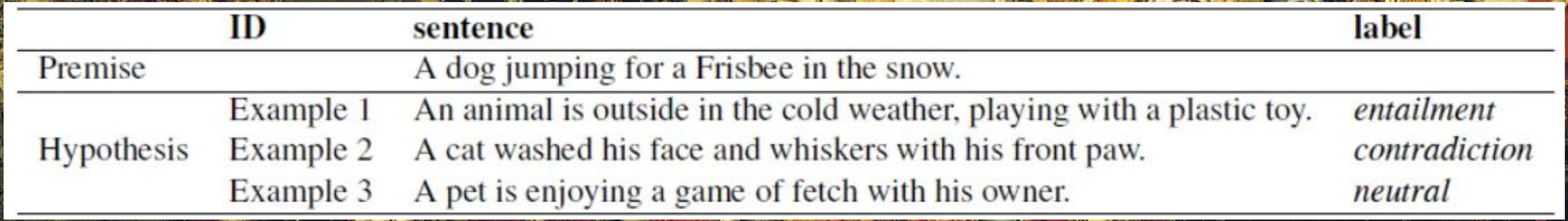
PyCon Estonia 2023

If your friends are bullshitting using SNLI

1. Jim rides a bike to school every morning.
2. Jim can ride a bike.

PyCon Estonia 2023

If your friends are bullshitting using SNLI

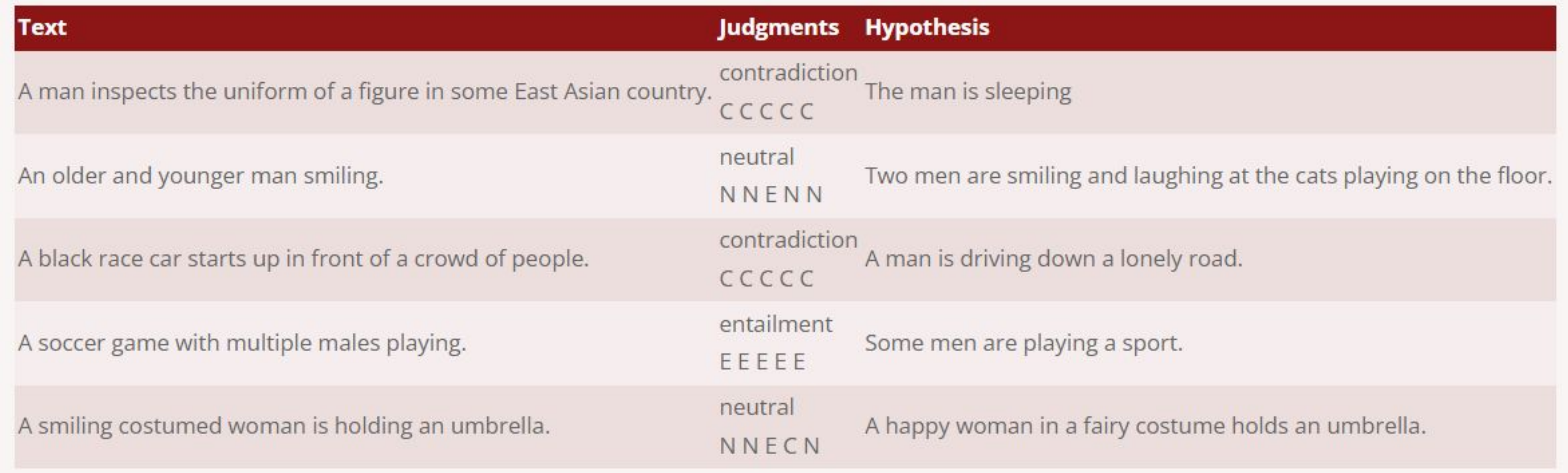


	ID	sentence	label
Premise		A dog jumping for a Frisbee in the snow.	
Hypothesis	Example 1	An animal is outside in the cold weather, playing with a plastic toy.	*entailment*
	Example 2	A cat washed his face and whiskers with his front paw.	*contradiction*
	Example 3	A pet is enjoying a game of fetch with his owner.	*neutral*

PyCon Estonia 2023

Ask a Fine-tuned LLM

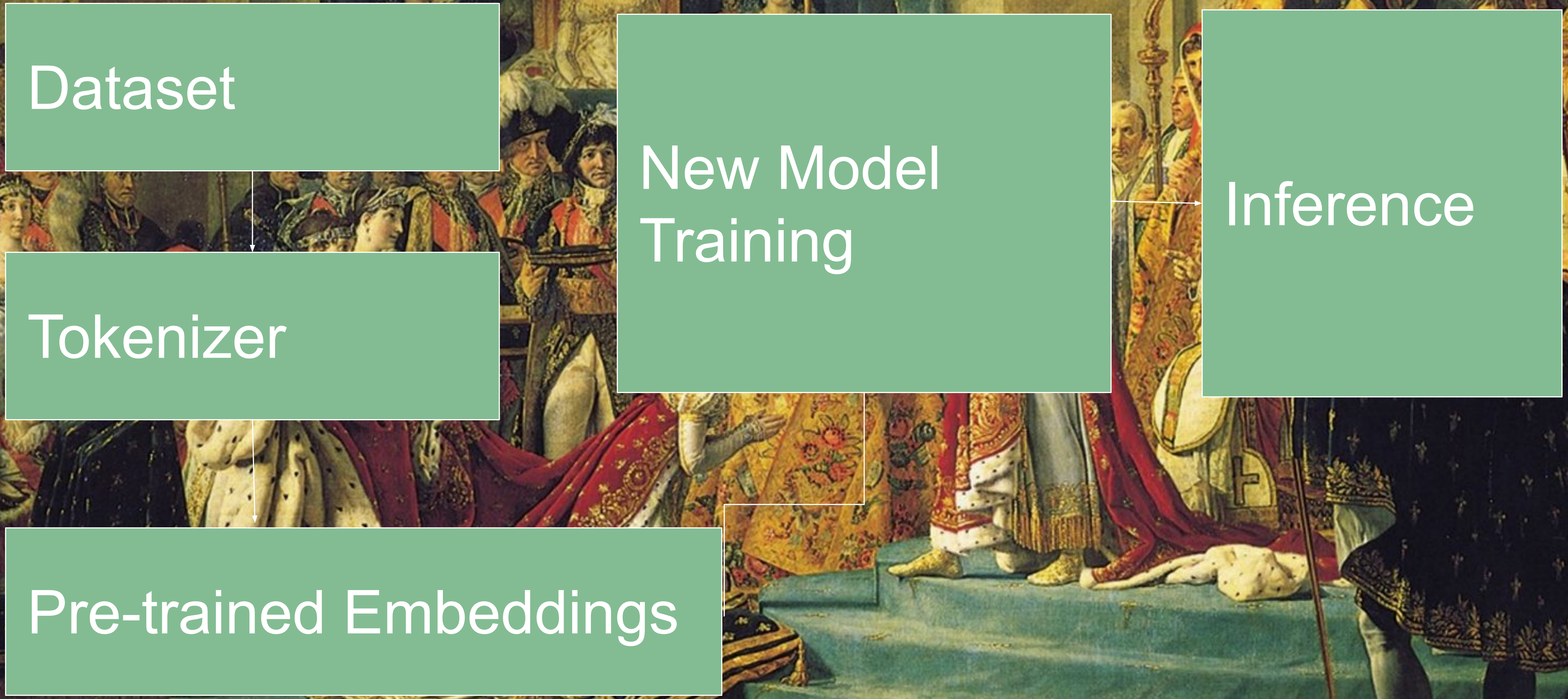
The Stanford Natural Language Inference (SNLI) corpus (version 1.0) is a collection of 570k human-written English sentence pairs manually labeled for balanced classification with the labels entailment, contradiction, and neutral. We aim for it to serve both as a benchmark for evaluating representational systems for text, especially including those induced by representation-learning methods, as well as a resource for developing NLP models of any kind.



Text	Judgments	Hypothesis
A man inspects the uniform of a figure in some East Asian country.	contradiction C C C C C	The man is sleeping
An older and younger man smiling.	neutral N N E N N	Two men are smiling and laughing at the cats playing on the floor.
A black race car starts up in front of a crowd of people.	contradiction C C C C C	A man is driving down a lonely road.
A soccer game with multiple males playing.	entailment E E E E E	Some men are playing a sport.
A smiling costumed woman is holding an umbrella.	neutral N N E C N	A happy woman in a fairy costume holds an umbrella.

PyCon Estonia 2023

If your friends are bullshitting using SNLI



Dataset

Tokenizer

Pre-trained Embeddings

New Model Training

Inference

Ask a Fine-tuned LLM

The BERT model was proposed in BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding by Jacob Devlin, Ming-Wei Chang, Kenton Lee and Kristina Toutanova. It’s a bidirectional transformer pretrained using a combination of masked language modeling objective and next sentence prediction on a large corpus comprising the Toronto Book Corpus and Wikipedia.

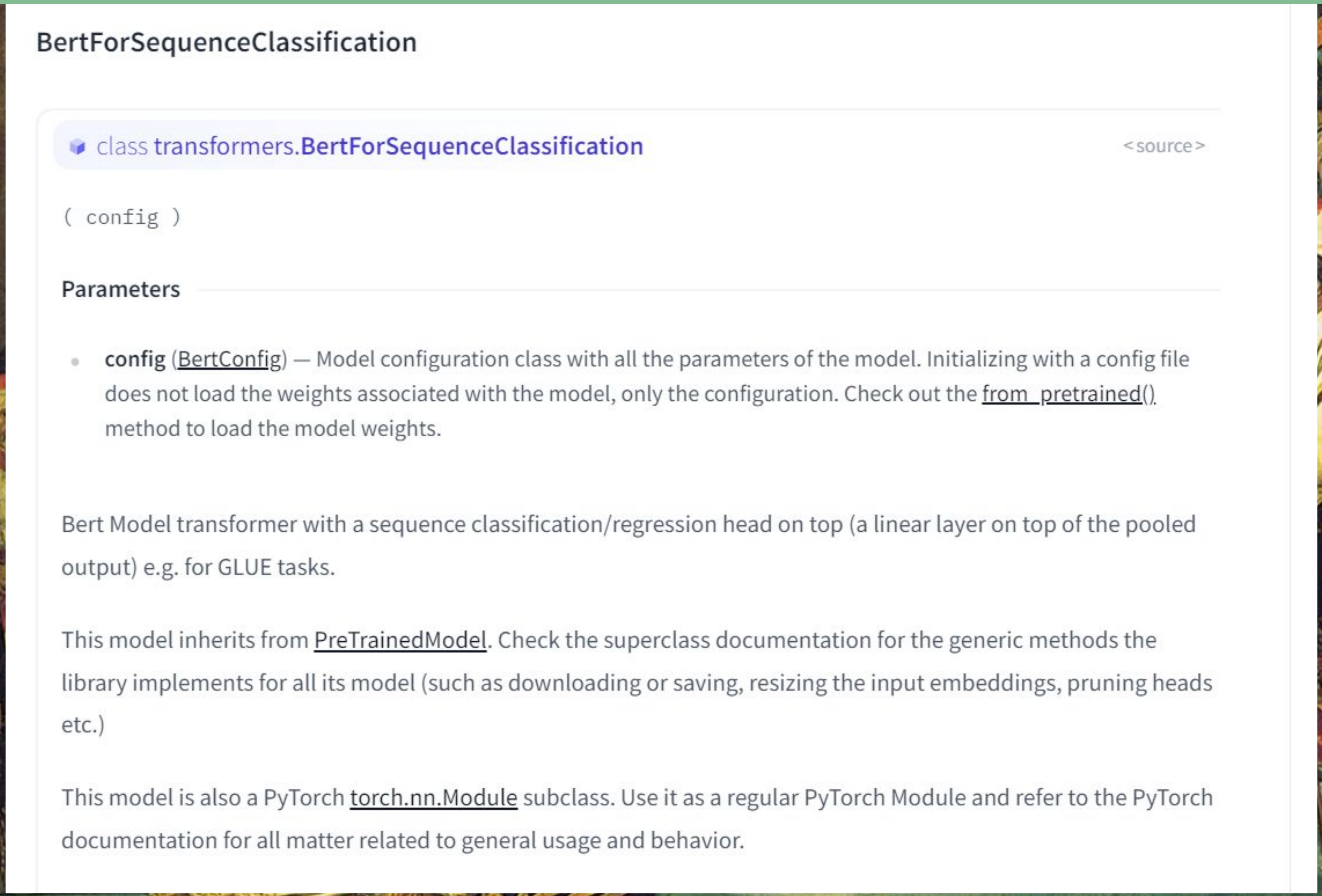
The abstract from the paper is the following:

We introduce a new language representation model called BERT, which stands for Bidirectional Encoder Representations from Transformers. Unlike recent language representation models, BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications.

BERT is conceptually simple and empirically powerful. It obtains new state-of-the-art results on eleven natural language processing tasks, including pushing the GLUE score to 80.5% (7.7% point absolute improvement), MultiNLI accuracy to 86.7% (4.6% absolute improvement), SQuAD v1.1 question answering Test F1 to 93.2 (1.5 point absolute improvement) and SQuAD v2.0 Test F1 to 83.1 (5.1 point absolute improvement).

PyCon Estonia 2023

Ask a Fine-tuned LLM



BertForSequenceClassification

class transformers.BertForSequenceClassification

`( config )`

Parameters

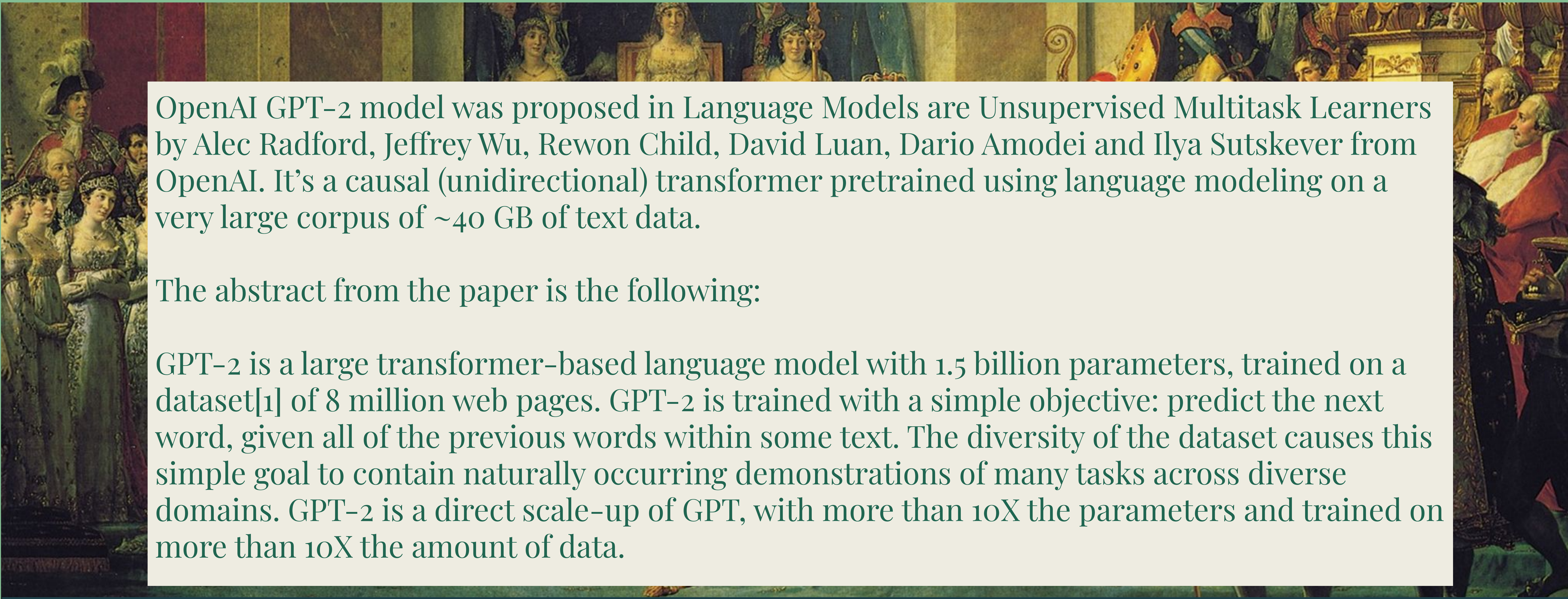
- **config** (BertConfig) — Model configuration class with all the parameters of the model. Initializing with a config file does not load the weights associated with the model, only the configuration. Check out the from_pretrained() method to load the model weights.

Bert Model transformer with a sequence classification/regression head on top (a linear layer on top of the pooled output) e.g. for GLUE tasks.

This model inherits from PreTrainedModel. Check the superclass documentation for the generic methods the library implements for all its model (such as downloading or saving, resizing the input embeddings, pruning heads etc.)

This model is also a PyTorch torch.nn.Module subclass. Use it as a regular PyTorch Module and refer to the PyTorch documentation for all matter related to general usage and behavior.

Ask a Fine-tuned LLM



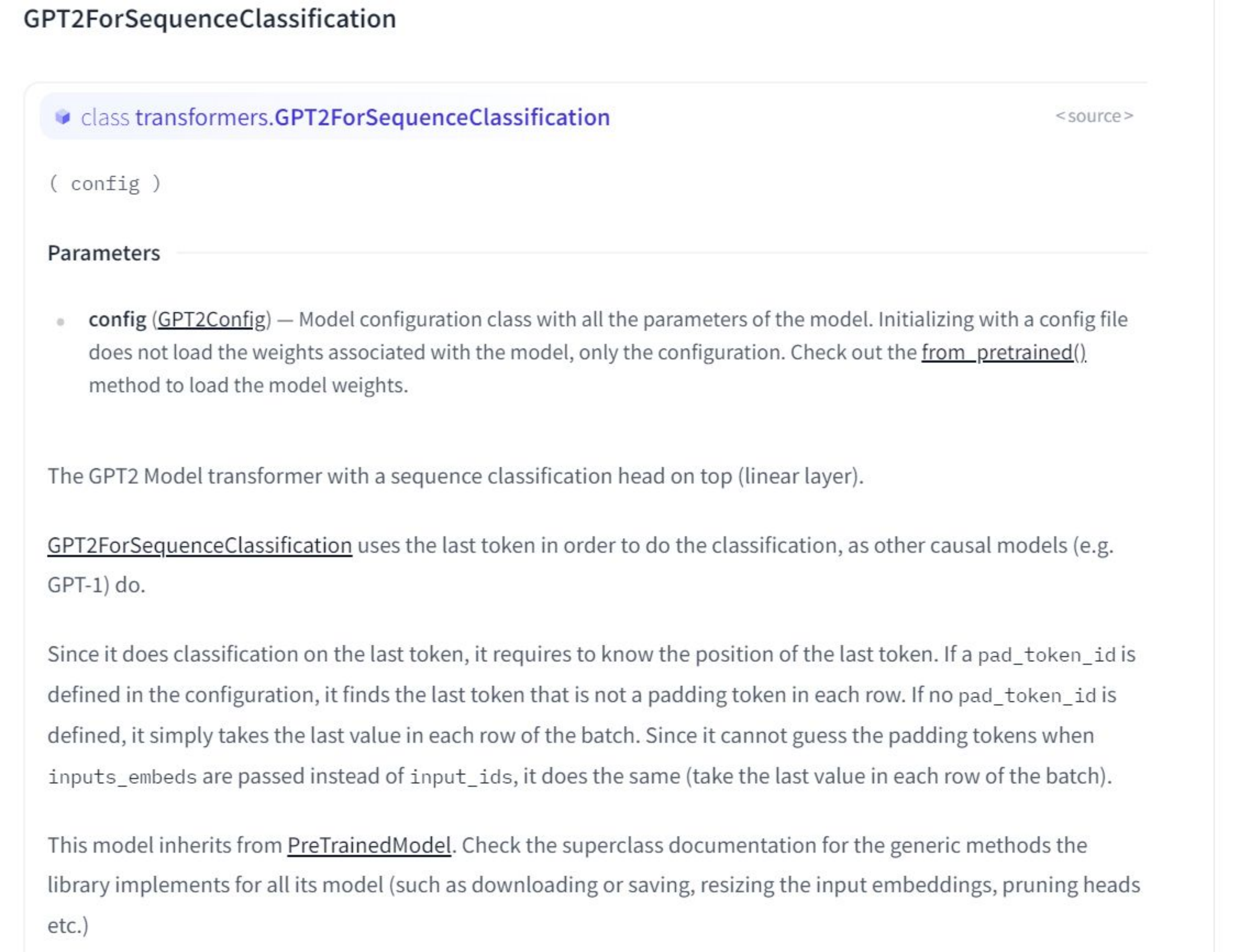
OpenAI GPT-2 model was proposed in Language Models are Unsupervised Multitask Learners by Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei and Ilya Sutskever from OpenAI. It’s a causal (unidirectional) transformer pretrained using language modeling on a very large corpus of ~40 GB of text data.

The abstract from the paper is the following:

GPT-2 is a large transformer-based language model with 1.5 billion parameters, trained on a dataset[1] of 8 million web pages. GPT-2 is trained with a simple objective: predict the next word, given all of the previous words within some text. The diversity of the dataset causes this simple goal to contain naturally occurring demonstrations of many tasks across diverse domains. GPT-2 is a direct scale-up of GPT, with more than 10X the parameters and trained on more than 10X the amount of data.

PyCon Estonia 2023

Ask a Fine-tuned LLM



GPT2ForSequenceClassification

class transformers.**GPT2ForSequenceClassification** <source>

(config)

Parameters

- **config** (GPT2Config) — Model configuration class with all the parameters of the model. Initializing with a config file does not load the weights associated with the model, only the configuration. Check out the from_pretrained() method to load the model weights.

The GPT2 Model transformer with a sequence classification head on top (linear layer).

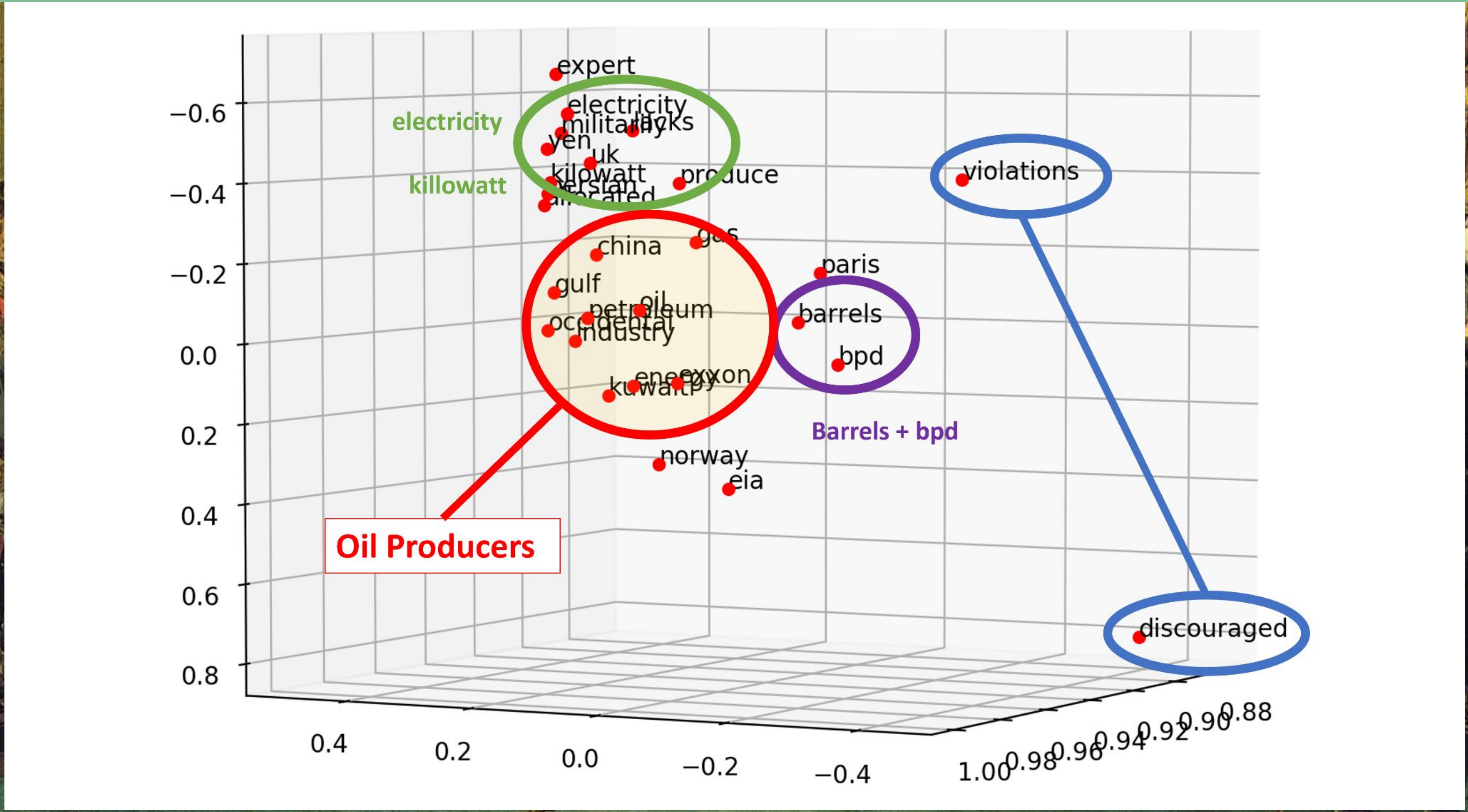
GPT2ForSequenceClassification uses the last token in order to do the classification, as other causal models (e.g. GPT-1) do.

Since it does classification on the last token, it requires to know the position of the last token. If a `pad_token_id` is defined in the configuration, it finds the last token that is not a padding token in each row. If no `pad_token_id` is defined, it simply takes the last value in each row of the batch. Since it cannot guess the padding tokens when `inputs_embeds` are passed instead of `input_ids`, it does the same (take the last value in each row of the batch).

This model inherits from PreTrainedModel. Check the superclass documentation for the generic methods the library implements for all its model (such as downloading or saving, resizing the input embeddings, pruning heads etc.)

PyCon Estonia 2023

Ask a Fine-tuned LLM



electricity

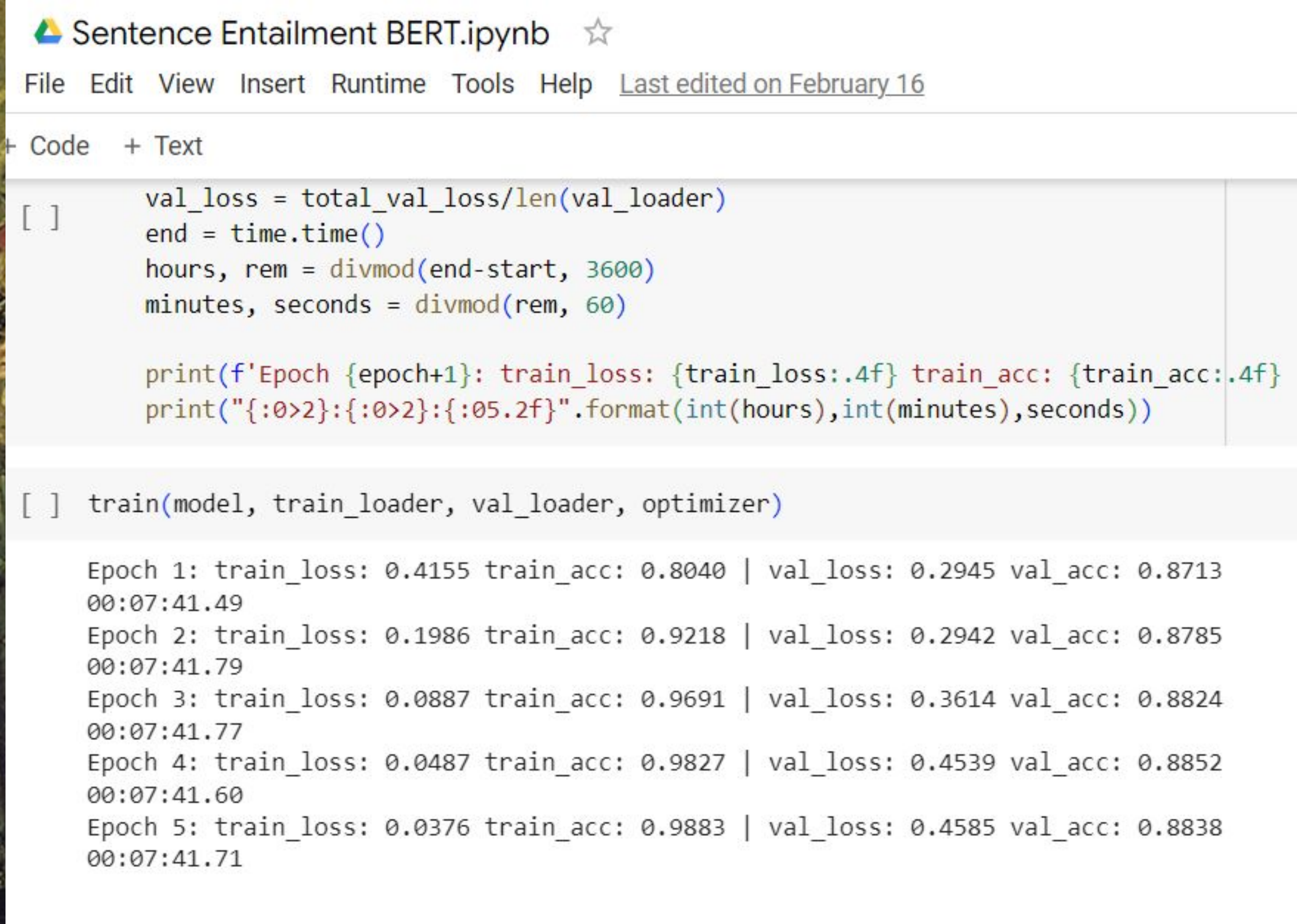
killowatt

Oil Producers

Barrels + bpd

PyCon Estonia 2023

Ask a Fine-tuned LLM

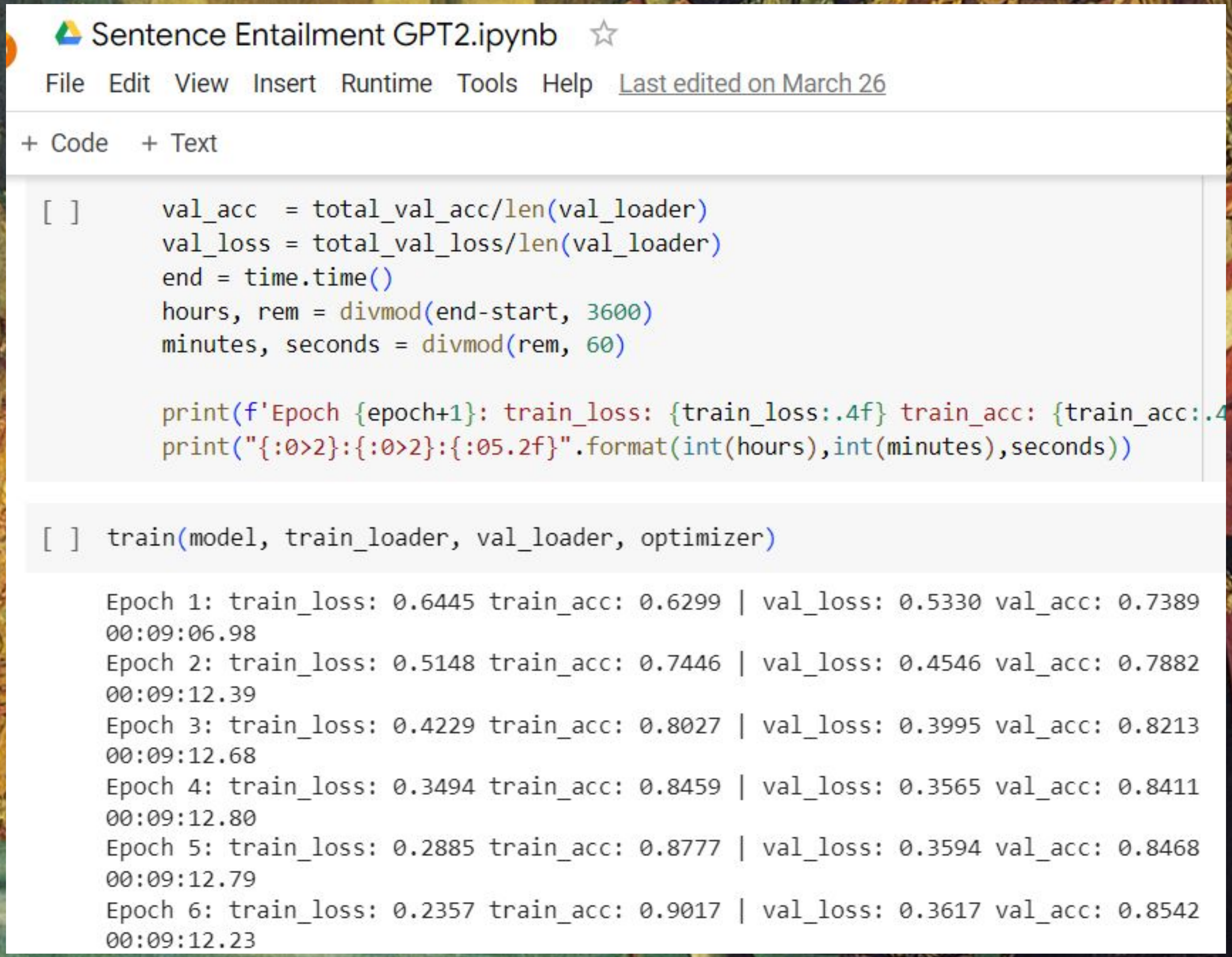

Sentence Entailment BERT.ipynb
File Edit View Insert Runtime Tools Help Last edited on February 16
+ Code + Text

```
[ ]     val_loss = total_val_loss/len(val_loader)
        end = time.time()
        hours, rem = divmod(end-start, 3600)
        minutes, seconds = divmod(rem, 60)

        print(f'Epoch {epoch+1}: train_loss: {train_loss:.4f} train_acc: {train_acc:.4f}
        print("{:0>2}:{:0>2}:{:05.2f}".format(int(hours),int(minutes),seconds))
```

```
[ ] train(model, train_loader, val_loader, optimizer)

    Epoch 1: train_loss: 0.4155 train_acc: 0.8040 | val_loss: 0.2945 val_acc: 0.8713
    00:07:41.49
    Epoch 2: train_loss: 0.1986 train_acc: 0.9218 | val_loss: 0.2942 val_acc: 0.8785
    00:07:41.79
    Epoch 3: train_loss: 0.0887 train_acc: 0.9691 | val_loss: 0.3614 val_acc: 0.8824
    00:07:41.77
    Epoch 4: train_loss: 0.0487 train_acc: 0.9827 | val_loss: 0.4539 val_acc: 0.8852
    00:07:41.60
    Epoch 5: train_loss: 0.0376 train_acc: 0.9883 | val_loss: 0.4585 val_acc: 0.8838
    00:07:41.71
```


Sentence Entailment GPT2.ipynb
File Edit View Insert Runtime Tools Help Last edited on March 26
+ Code + Text

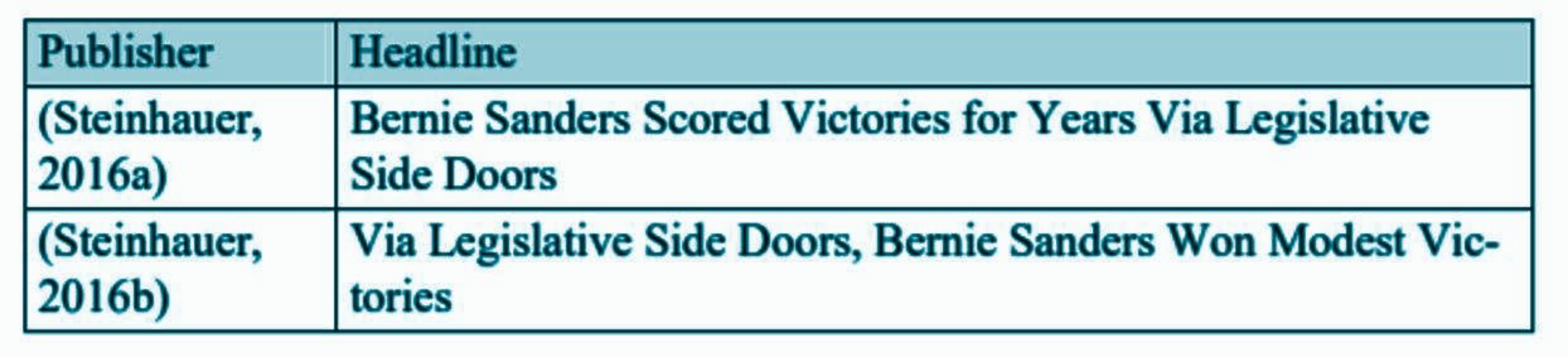
```
[ ]     val_acc  = total_val_acc/len(val_loader)
        val_loss = total_val_loss/len(val_loader)
        end = time.time()
        hours, rem = divmod(end-start, 3600)
        minutes, seconds = divmod(rem, 60)

        print(f'Epoch {epoch+1}: train_loss: {train_loss:.4f} train_acc: {train_acc:.4
        print("{:0>2}:{:0>2}:{:05.2f}".format(int(hours),int(minutes),seconds))
```

```
[ ] train(model, train_loader, val_loader, optimizer)

    Epoch 1: train_loss: 0.6445 train_acc: 0.6299 | val_loss: 0.5330 val_acc: 0.7389
    00:09:06.98
    Epoch 2: train_loss: 0.5148 train_acc: 0.7446 | val_loss: 0.4546 val_acc: 0.7882
    00:09:12.39
    Epoch 3: train_loss: 0.4229 train_acc: 0.8027 | val_loss: 0.3995 val_acc: 0.8213
    00:09:12.68
    Epoch 4: train_loss: 0.3494 train_acc: 0.8459 | val_loss: 0.3565 val_acc: 0.8411
    00:09:12.80
    Epoch 5: train_loss: 0.2885 train_acc: 0.8777 | val_loss: 0.3594 val_acc: 0.8468
    00:09:12.79
    Epoch 6: train_loss: 0.2357 train_acc: 0.9017 | val_loss: 0.3617 val_acc: 0.8542
    00:09:12.23
```

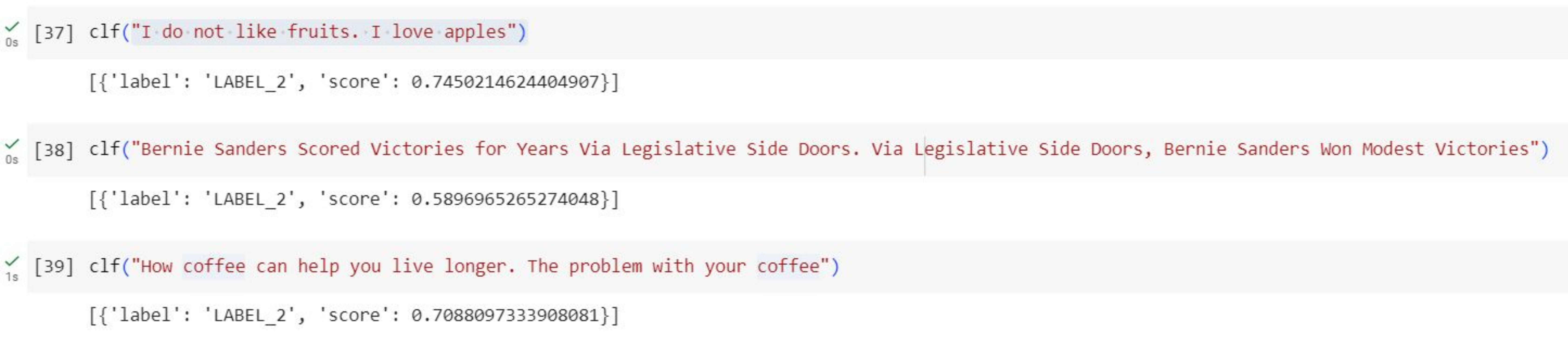
Ask a Fine-tuned LLM



Publisher	Headline
(Steinhauer, 2016a)	Bernie Sanders Scored Victories for Years Via Legislative Side Doors
(Steinhauer, 2016b)	Via Legislative Side Doors, Bernie Sanders Won Modest Victories

PyCon Estonia 2023

Ask a Fine-tuned LLM



```
[37] clf("I do not like fruits. I love apples")

     [{'label': 'LABEL_2', 'score': 0.7450214624404907}]

[38] clf("Bernie Sanders Scored Victories for Years Via Legislative Side Doors. Via Legislative Side Doors, Bernie Sanders Won Modest Victories")

     [{'label': 'LABEL_2', 'score': 0.5896965265274048}]

[39] clf("How coffee can help you live longer. The problem with your coffee")

     [{'label': 'LABEL_2', 'score': 0.7088097333908081}]
```

EuroPython 2022, PyCon Hong Kong 2021, PyCon Sweden 2021

💡 Objective: Analyze sentiments in news headlines to examine bias, comparing it to public perception from surveys

🧩 Challenge: Initial approaches failed to capture the complexity of sentiment and bias in headlines

🔄 Revised Approach: Leveraged advanced NLP techniques to better align sentiment with real-world reader biases

🏆 Outcome: Failure led to deeper understanding of contextual sentiment analysis and bias measurement—essential for interpreting media influence

Lessons from Failed Research: Analyzing News Bias from Headlines

PyOhio 2020

Analyzing bias in children’s educational materials

Research has shown that children's educational materials often reinforce gender biases, such as portraying female characters predominantly as mothers and housewives while male characters are seen as breadwinners. Using NLP techniques, we can analyze representation bias by tracking the frequency of gendered terms, identify stereotypes through adjective associations, and detect victim-blaming language in texts. For instance, in some countries, the proportion of female characters in mathematics textbooks was found to be as low as 30%. To combat these biases, it's crucial to revise curricula to promote gender equality, update textbooks to be more inclusive, and use NLP tools to continually monitor and correct these biases in educational content.

PyOhio 2020

Analyzing bias in children's educational materials

Research Focus:

Examining how gender bias in children's educational materials shapes perceptions and roles.

Example: Female characters predominantly portrayed as housewives, while male characters are shown as breadwinners (Kostas, 2019).

Key Findings in Bias

Representation Bias: Female characters underrepresented in fields like mathematics.

Stereotype Bias: Gendered roles (e.g., doctor = male, nurse = female) perpetuated in texts.

Culture of Blame: Identifying victim-blaming language using NLP tools.

PyOhio 2020

Analyzing bias in children's educational materials

Techniques for Bias Detection

Representation Analysis using token frequency in texts.

Stereotype Evaluation through adjective and role association using spaCy and NLTK.

Blame Language Detection via Path Model of Blame, identifying causality and labeling bias.

The Change: Advocating for Gender Equality

Revise curriculum to reflect gender equality and revise textbooks to remove gender stereotypes.

Implement tools to analyze educational content for bias and advocate for inclusivity.

STORIES

VII

PyCon Lithuania 2023

Chatting with ChatGPT about everything and nothing at all

VIII

AI in Software Development Summit 2024

Experimenting with SVMs

IX

FOSSASIA 2016 Shoes for the Visually Impaired

PyCon Lithuania 2023

Chatting with ChatGPT about everything and nothing at all

Introduction to ChatGPT and AGI

Overview of ChatGPT's rise and the question of whether it's a true AGI (Artificial General Intelligence).

How Large Language Models (LLMs) Work

Explanation of the training process for LLMs, including data usage and model architecture.

The Wit Test: Riddles for ChatGPT

A fun experiment where ChatGPT is tested with various riddles to explore its cognitive capabilities.

Contextual Understanding Limitations

Discussion on how ChatGPT often struggles to understand complex contexts and nuanced language.

PyCon Lithuania 2023

Chatting with ChatGPT about everything and nothing at all

Verbal Math and Conceptual Challenges

Analysis of ChatGPT’s difficulty in verbal math problems and basic conceptual understanding, like Venn diagrams.

Sensory Experience Gaps

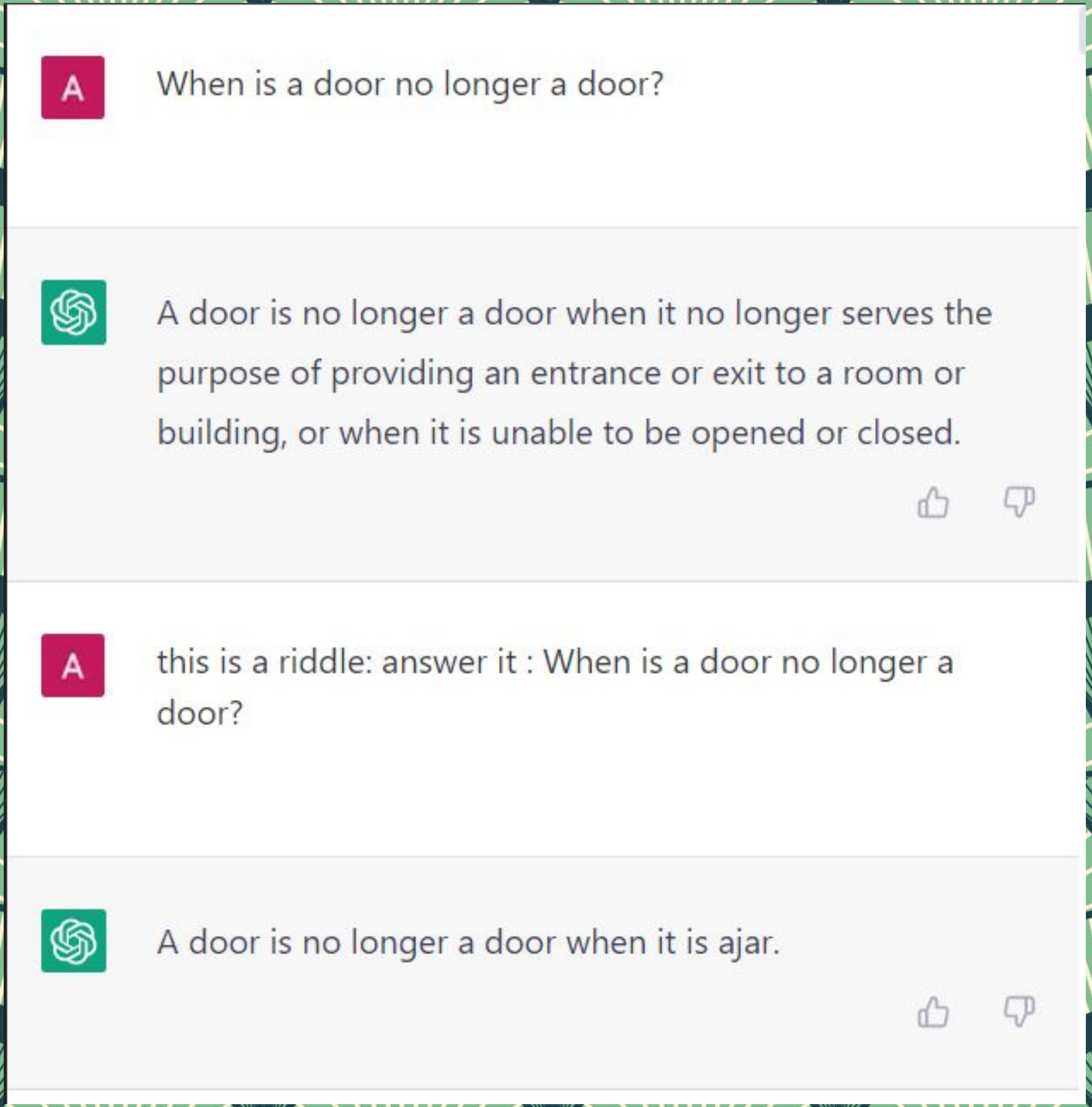
Highlighting the model’s inability to process real-world sensory experiences, such as "hearing an egg crack."

Metaphorical Language and Relationships

Deep dive into how ChatGPT interprets metaphorical language and convoluted relationships, and why it falters.

Comparing ChatGPT with Human Performance

Comparison of ChatGPT’s responses to audience answers to assess where AGI stands in relation to human cognition.

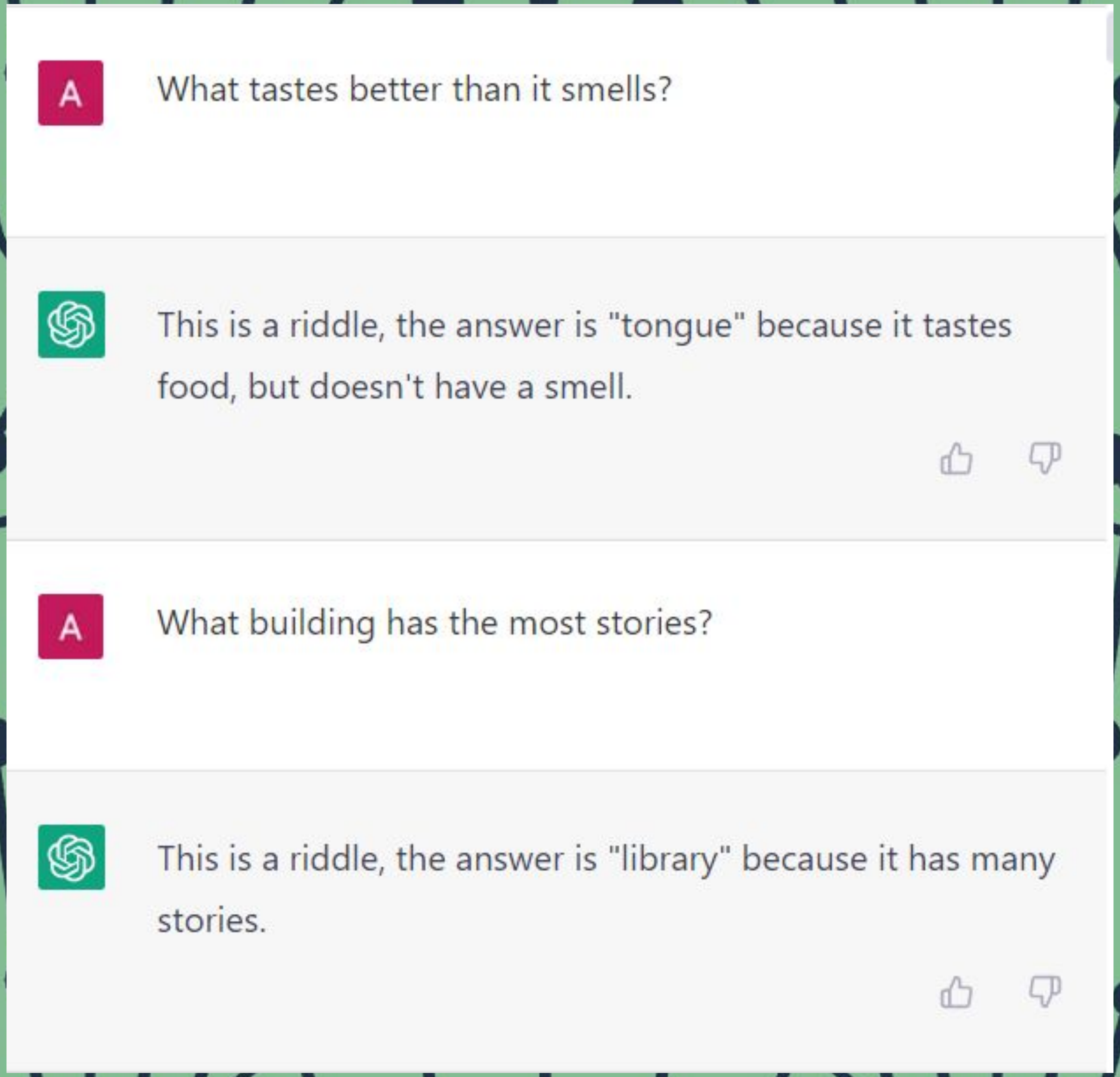


When is a door no longer a door?

A door is no longer a door when it no longer serves the purpose of providing an entrance or exit to a room or building, or when it is unable to be opened or closed.

this is a riddle: answer it : When is a door no longer a door?

A door is no longer a door when it is ajar.

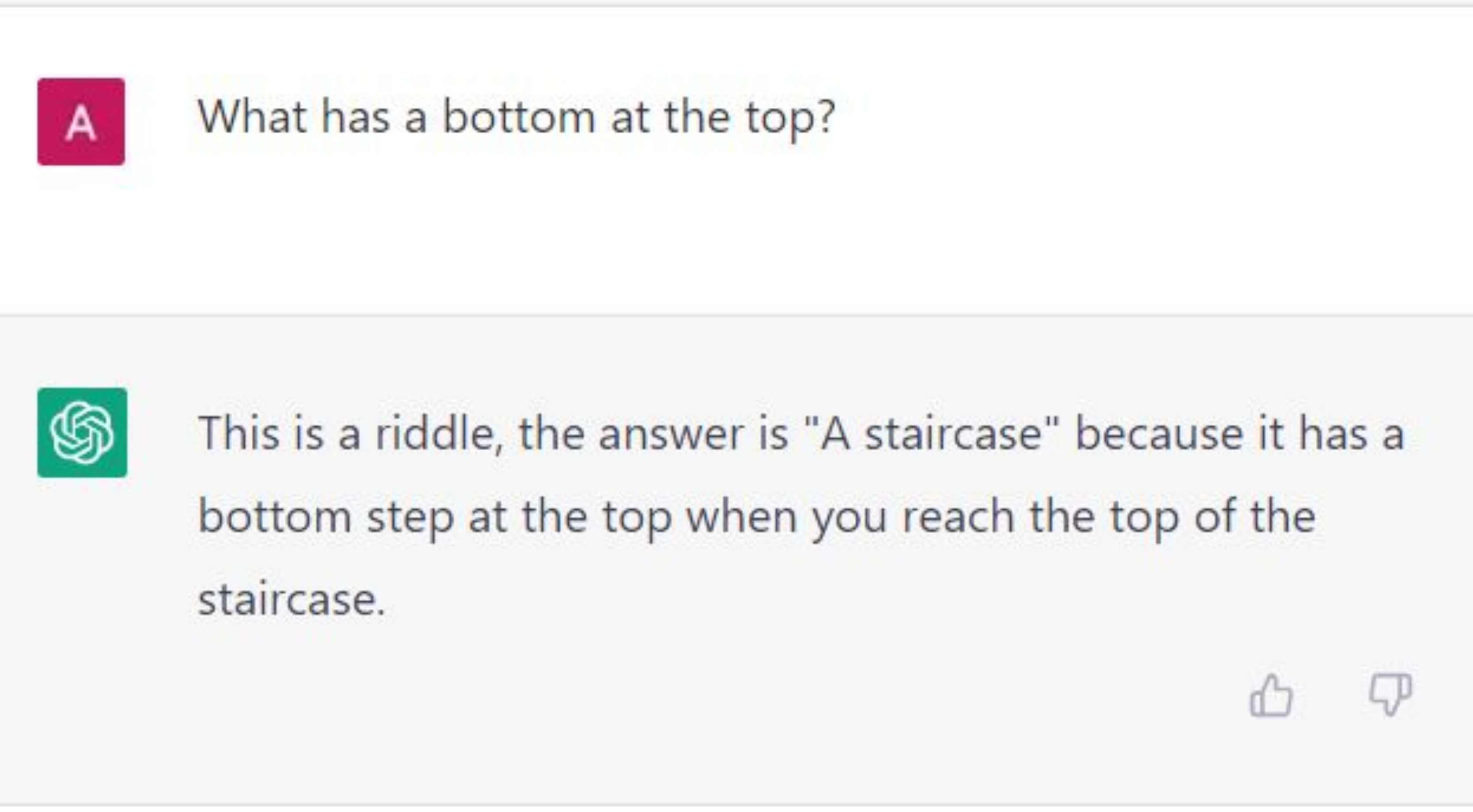


What tastes better than it smells?

This is a riddle, the answer is "tongue" because it tastes food, but doesn't have a smell.

What building has the most stories?

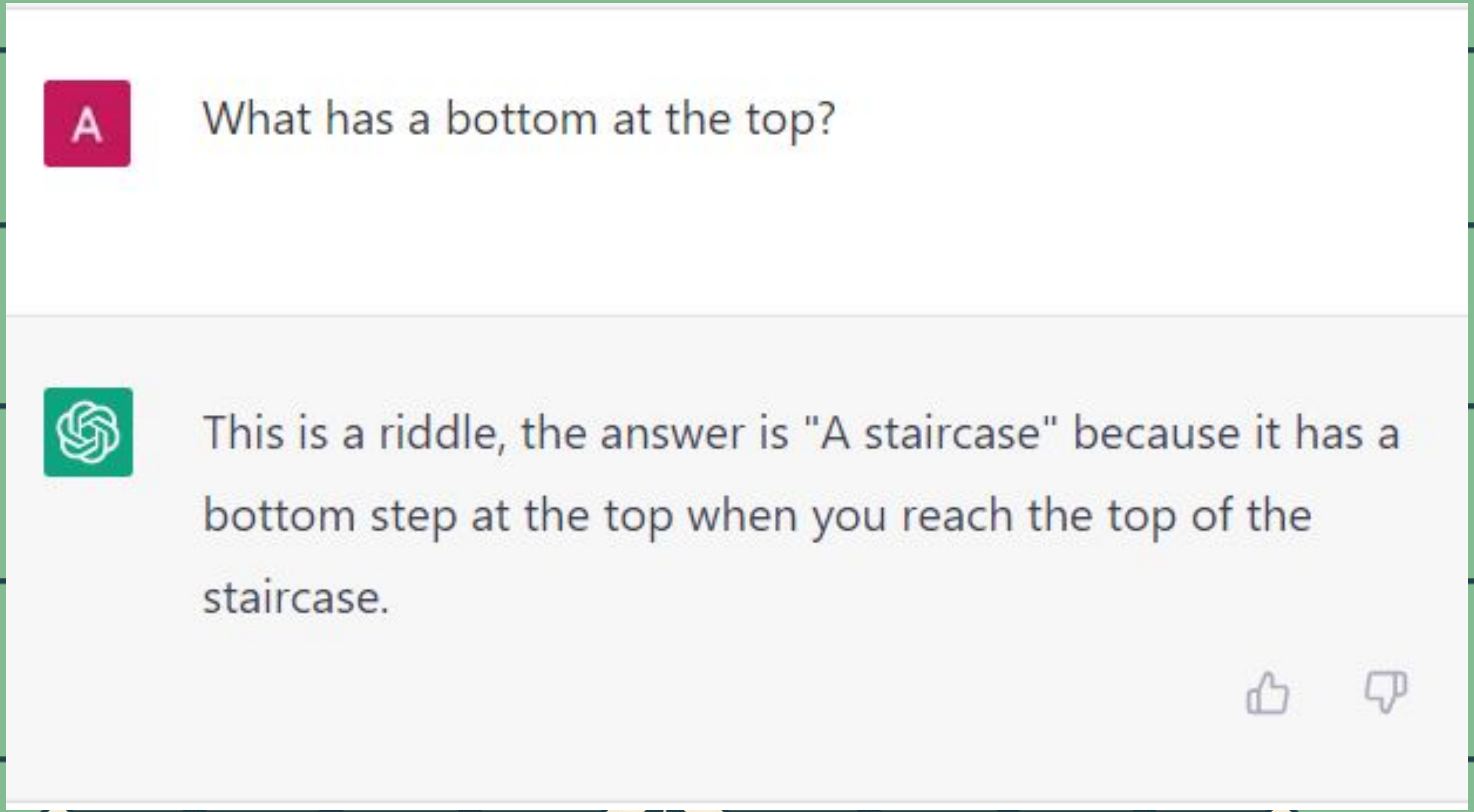
This is a riddle, the answer is "library" because it has many stories.



A

What has a bottom at the top?

This is a riddle, the answer is "A staircase" because it has a bottom step at the top when you reach the top of the staircase.



What has a bottom at the top?

This is a riddle, the answer is "A staircase" because it has a bottom step at the top when you reach the top of the staircase.

Legs

AI in Software Development Summit 2024

Experimenting with SVMs

What is Support Vector Clustering (SVC)?
Clustering technique using SVM principles for unsupervised learning

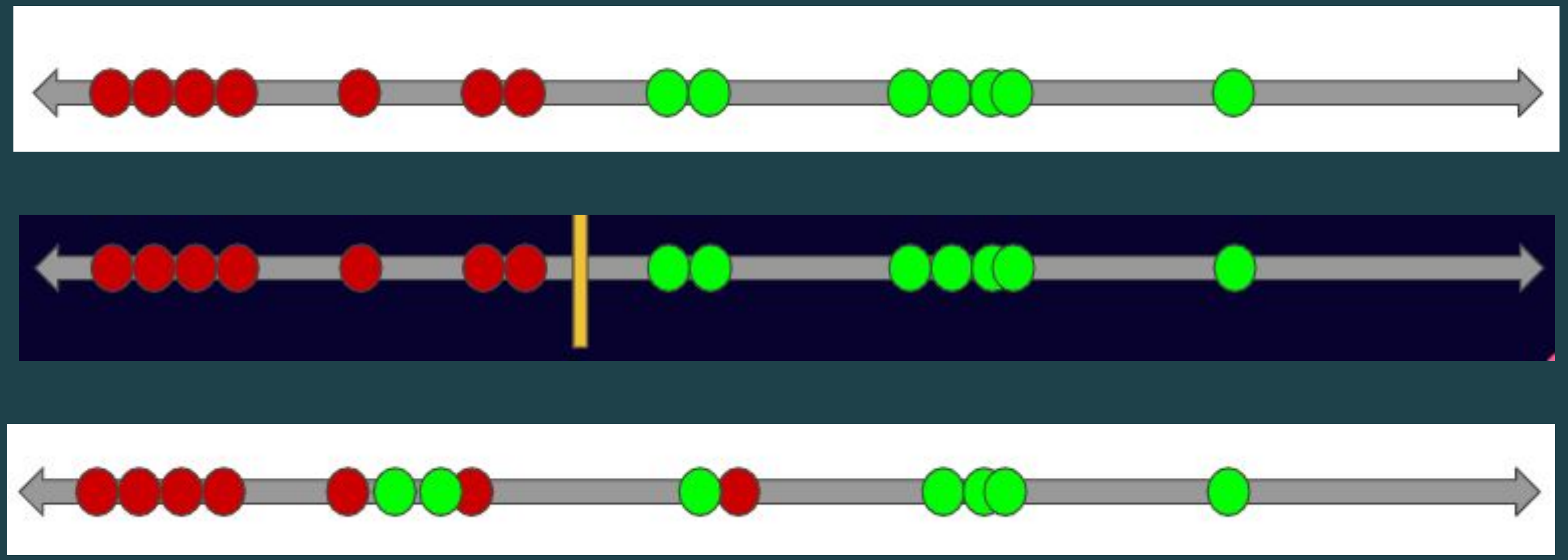
Maps data to high-dimensional Hilbert space via kernel trick

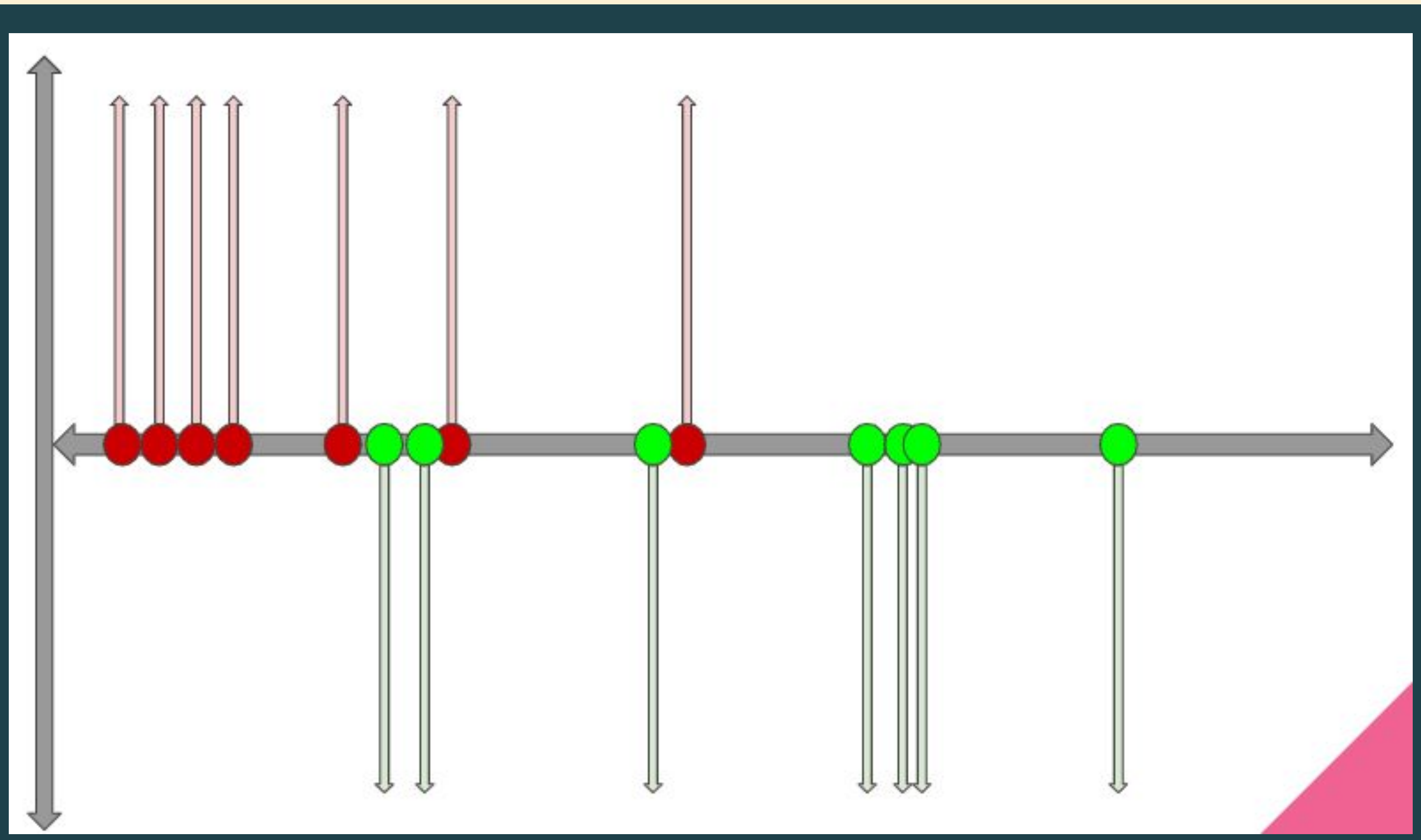
Finds minimum enclosing sphere, then maps boundaries back to original space

Uses Gaussian kernel (nonlinear transformation)
Two key hyperparameters:
q: kernel width (controls cluster granularity)
p (or C): soft margin (controls tolerance to outliers)
Does not assume shape or number of clusters; adapts naturally

Handles non-convex, overlapping, noisy data
Better accuracy than traditional clustering on complex datasets (e.g., Iris)
Elegant blend of theory (SVM) + practical clustering power

Experimenting with SVMs





FOSSASIA 2016

Shoes for the Visually Impaired

Purpose: Designed to improve daily mobility and confidence for the visually impaired.

Core Feature: Shoes equipped with ground-level obstruction sensing.

Control Unit: Arduino + Bluetooth module transmit sensor data to an Android device.

App Interface: Android app (built with MIT App Inventor + PhoneGap) provides voice feedback via a single-button interface.

Advanced Vision: MATLAB-based prototype explores cloud-powered image processing for detecting colors, motion, and signals.

Scalability: Lays the groundwork for future modules (e.g., detecting hanging obstacles) and broader assistive tech innovation.

STORIES

x

Geekle.us

Create your own data

CREATE YOUR OWN DATA - GEEKLE.US

Generative models are useful for creating synthetic data, even if they struggle with factual accuracy and reliability.

Use Case 1 – Tautology Generator: Train a model to generate tautologies, focusing on internal consistency rather than grounded truth.

Use Case 2 – Travel Image Tagging: Generate image captions from personal photos, using LLMs to create contextual and creative tags for journaling or curation.

Synthetic data creation through LLMs: Use LLMs to bootstrap datasets for tasks like tautology generation or image captioning without relying on factual correctness. Techniques like distillation and fine-tuning help adapt LLMs to specialized tasks by reusing generated data and tailoring models for specific use cases. Takeaway: Synthetic data generation can unlock meaningful, creative workflows in areas where factual accuracy is not a priority.

Result: Generate 10 examples of tautology, do not give reasons why

1. The sun is the sun
2. Water is wet
3. A tree is a tree
4. Fire is hot
5. A car is a car
6. A cat is a cat
7. A dog is a dog
8. A house is a house
9. A bird is a bird
10. A fish is a fish

Result: Generate 10 examples of metaphors, do not give reasons why, use the word CD in the sentence

1. Her voice was a CD that played on repeat in my mind.
2. His anger was a CD that scratched and scuffed the surface of our relationship.
3. My job was a CD that skipped and glitched, making it hard to enjoy the music.
4. The city was a CD that had too many scratches and skips, making it hard to listen to.
5. Our love was a CD that had a perfect harmony, making it a beautiful sound.
6. His lies were a CD that had a hidden track, revealing the truth.
7. My fear was a CD that had a skip, making it hard to listen to.
8. The sunset was a CD that had a fade-out, ending the day on a high note.
9. My mem

Result: Generate 10 examples of metaphors, do not give reasons why, use the word SUV in the sentence

THANK YOU!

THANK YOU!

Credits : SlidesCarnival

RESOURCE PAGE

This presentation template uses the following free fonts:

Titles: Marcellus
Body Copy: Ovo

Don't forget to delete this page before presenting.

Happy Designing!