University of Wyoming
College of Engineering and Physical Sciences
Electrical Engineering and Computer Science

LLM Usage and Demo

SENIOR DESIGN FALL 2024

TESSA RODGERS & YAQOOB MAJEED

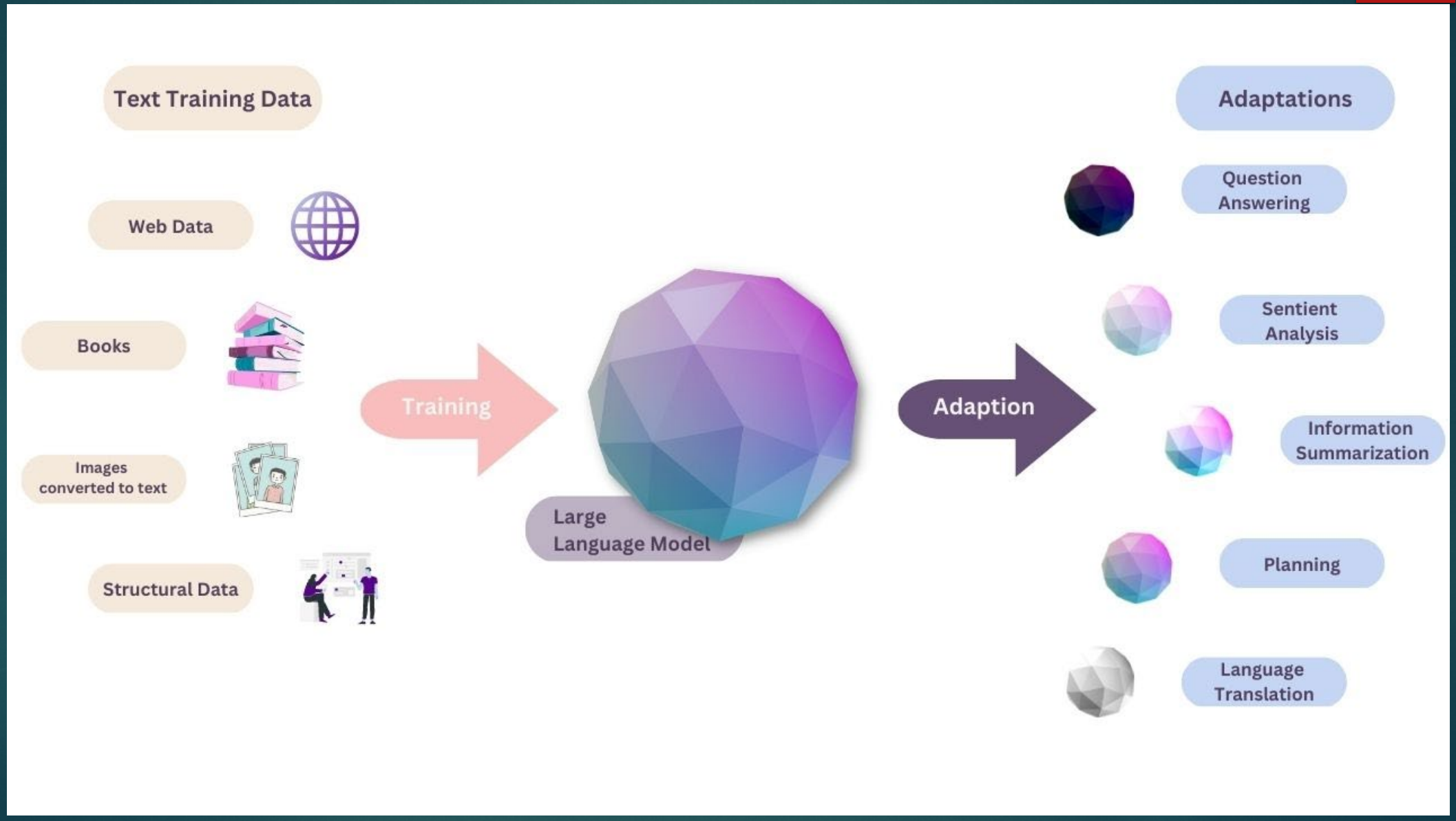
Overview / Module Goals

- Understand what an LLM is
- How to use open source LLMs such as ChatGPT and Gemini
- Understand the capabilities and limitations of LLMs in practice
- How to use ZeroGPT to identify LLM generated text
- LLMs in writing, math, and coding

LLM: Definition

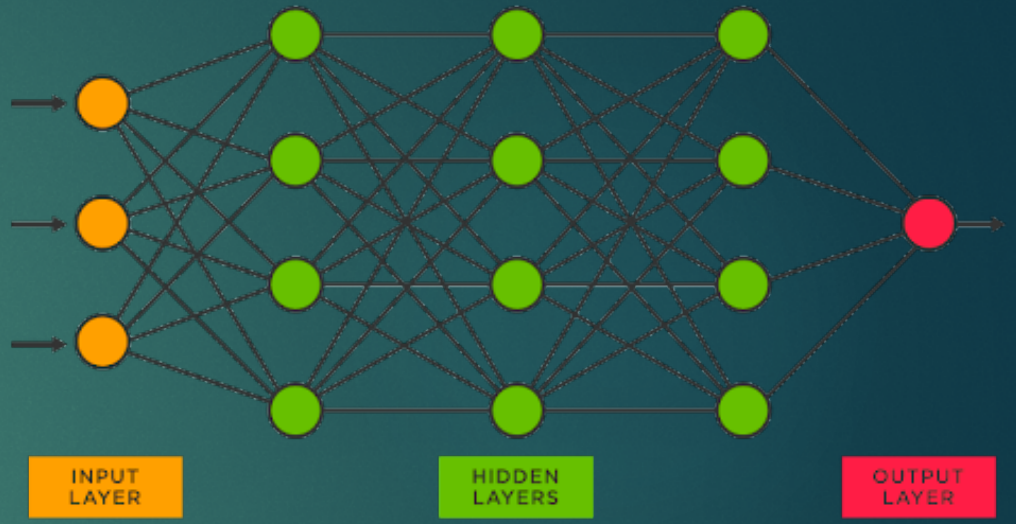
- Large Language Models (LLMs) are a class of foundation artificial intelligence (AI) models
- Primarily designed to understand and **generate** human-like **text**
  - Generative AI models create new content as their output, can be text, images, music, code, etc.
  - Natural Language Understanding (NLU)
  - Natural Language Processing (NLP)

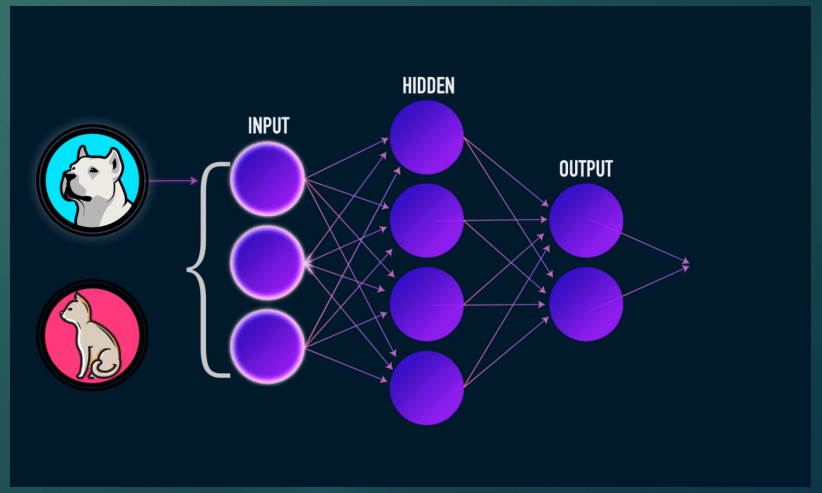
Model Details



Model Details

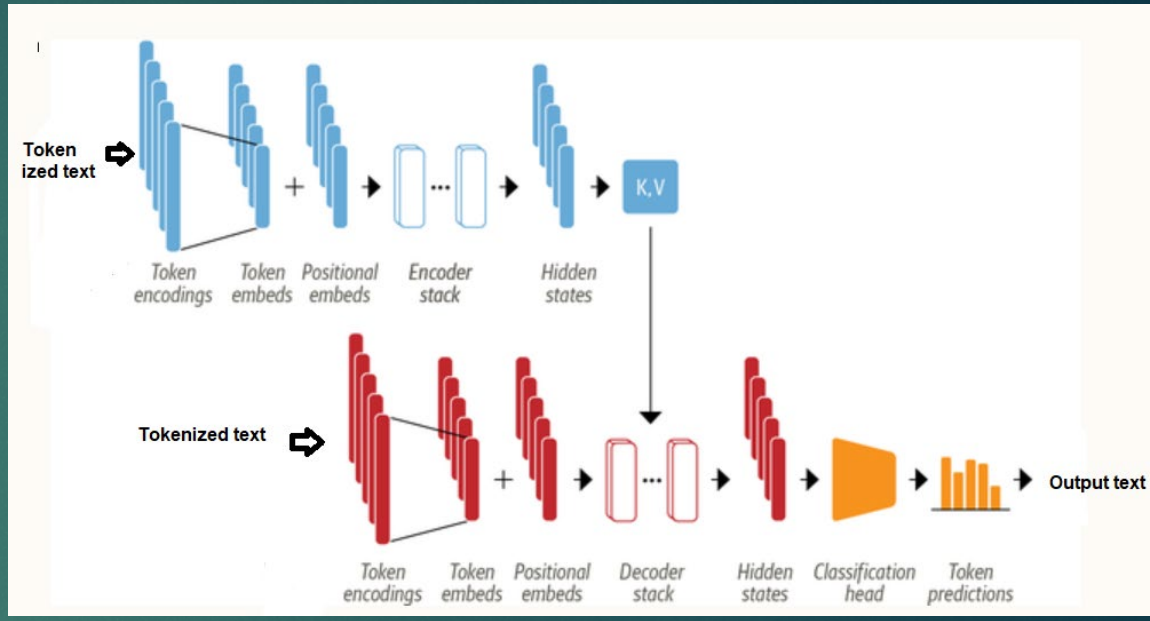
- Largely based on deep learning techniques
  - Neural network architectures such as transformers, which are designed to work with sequential data
- Comprised of multiple layer neural networks
- Networks have parameters that can be tuned for performance
- Transformer networks include an attention mechanism which captures relationships within the data
  - This increases their capability to recognize context

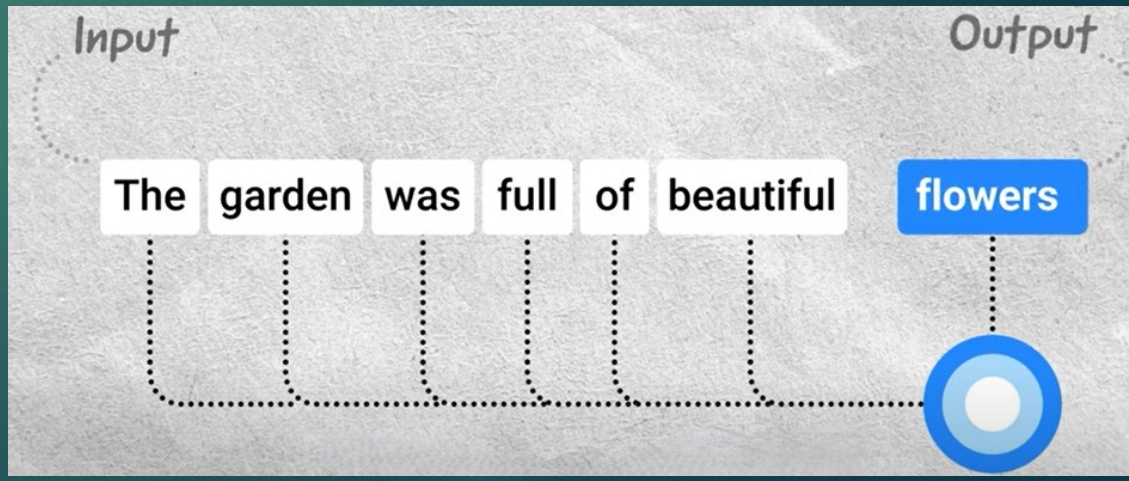




Training

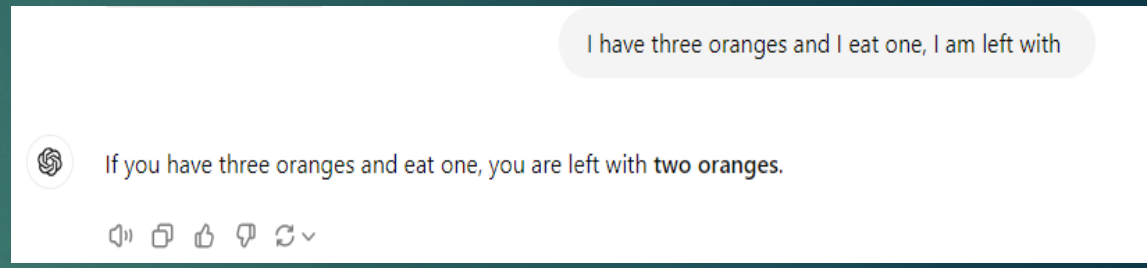
- LLMs are foundation models
  - Trained on (very) large amounts of data to provide foundational capabilities to apply the model to many different cases and task
  - LLMs are typically trained on the equivalent of billions of pages of text
- Model learns to predict the next word in a sentence
  - Uses context based on previous words
- Tokens: sequence of characters within a sentence (such as a word)
- Learns by assigning a probability score to the tokenized words
- Tokens are transformed into embeddings: numerical representations of text

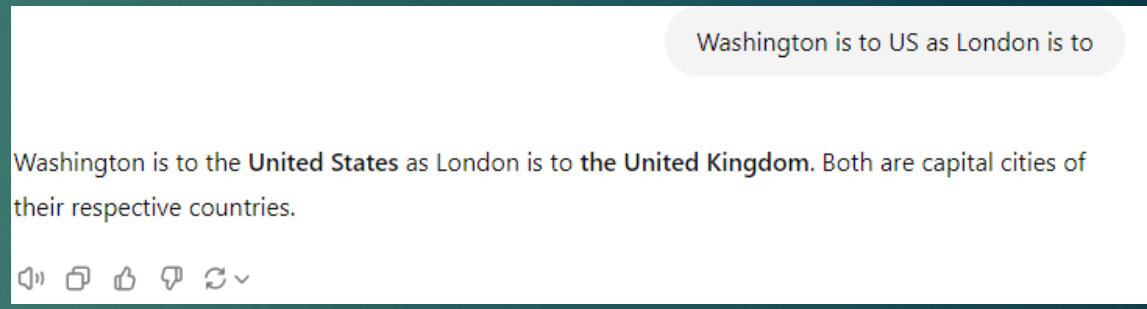


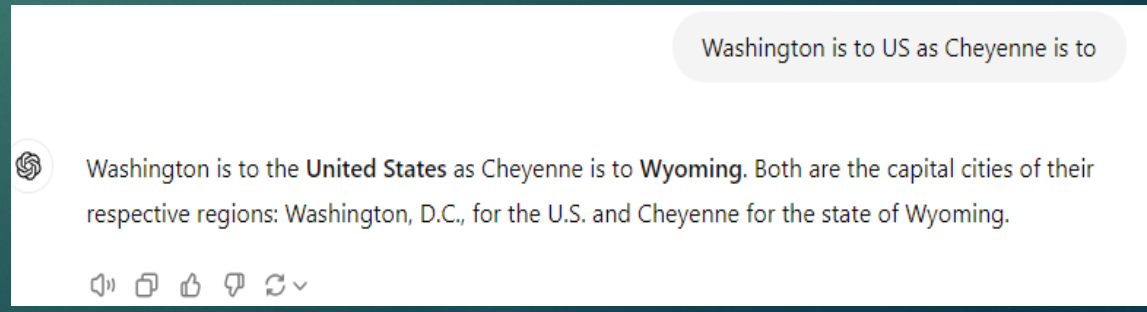


Training

- LLMs are foundation models
  - Trained on (very) large amounts of data to provide foundational capabilities to apply the model to many different cases and task
  - LLMs are typically trained on the equivalent of billions of pages of text
- Model learns to predict the next word in a sentence
  - Uses context based on previous words
- Tokens: sequence of characters within a sentence (such as a word)
- Learns by assigning a probability score to the tokenized words
- Tokens are transformed into embeddings: numerical representations of text







Increasing Performance

- Prompt engineering and prompt tuning to increase the likelihood of a robust output
- Reinforcement learning with human feedback (RLHF)
  - Used to remove biases, inappropriate speech and incorrect answers
- Since models must be trained on large datasets, it is very time intensive
  - Models aren't updated with most recent information
- Thus, some models have the capability to search the internet to inform their output

ChatGPT - Introduction

- **Chat Generative Pre-Trained Transformer (ChatGPT)** is an AI chatbot released by OpenAI in 2022. (latest version is GPT-4o as of May 2024)
- The core function of ChatGPT is to mimic a human conversationalist.
- AI chatbots have existed for many years, but the domain of their conversations are very limited, and their application is usually limited to specific tasks (e.g. online customer service).
- However, due to the cutting-edge model architecture, reinforcement learning with human feedback, and an enormous and meticulously curated textual data set, ChatGPT is the most versatile and robust AI chatbot to date.
- Many other companies (Google, Microsoft, Amazon, etc.) have also developed competitor chatbots using the same ML techniques.

ChatGPT - Introduction

- ChatGPT is a *large language model*, a type of AI trained on vast amounts of curated text from the internet.
- The underlying architecture of GPT is the *Transformer*, a family of neural networks which have been the premier natural language processing model since its inception in 2017.
- ChatGPT learns syntactic and semantic language patterns by analyzing billions of sentences and words.
- It is an *autoregressive* model, meaning that it generates the next most likely word in its output by looking at its previously generated words.
- The model's responses are generated based on probability of word occurrence, and the computer doesn't "actually understand" what it's saying.
- This means that there is no guarantee that what ChatGPT outputs is accurate or sensical.

ChatGPT - Capabilities

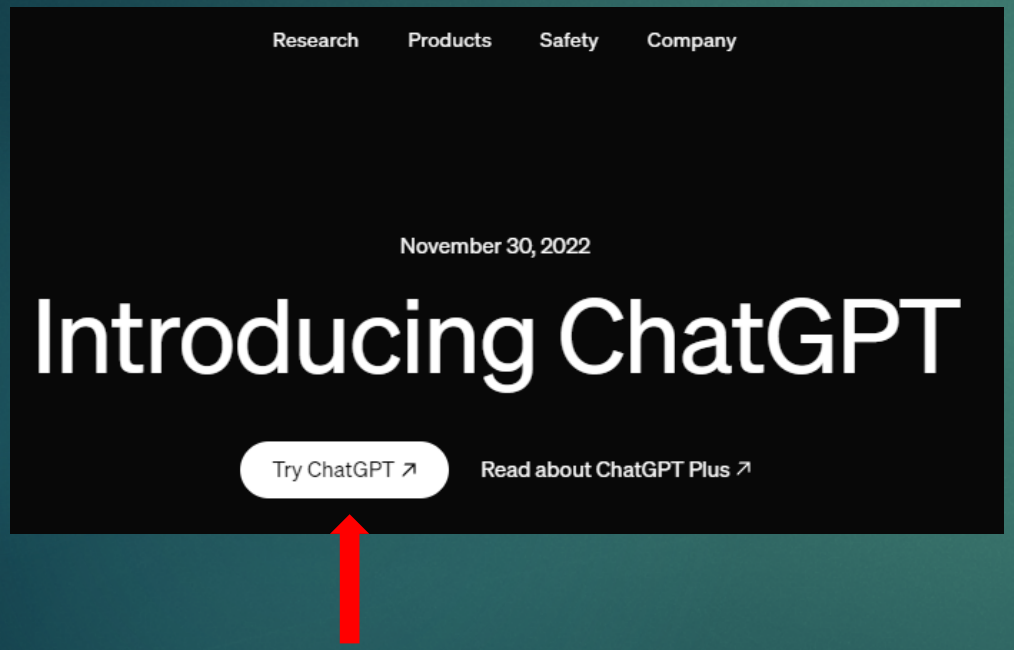
- Due to its emergent versatility, ChatGPT can perform many complex tasks, including but not limited to:
  - Writing and debugging computer programs
  - Composing essays and stories
  - Translating or summarizing text
  - Answering examination questions
  - Simulate scenarios
  - Perhaps most interestingly, receive feedback and remembers conversations, allowing extended and dynamic conversations.
- Much development has been done to increase the accuracy of ChatGPT's responses, reduce bias in answering subjective/controversial topics, and overall reduce harmful or deceitful responses.
- While large improvements have been made, ChatGPT still receives criticism for its shortcomings.

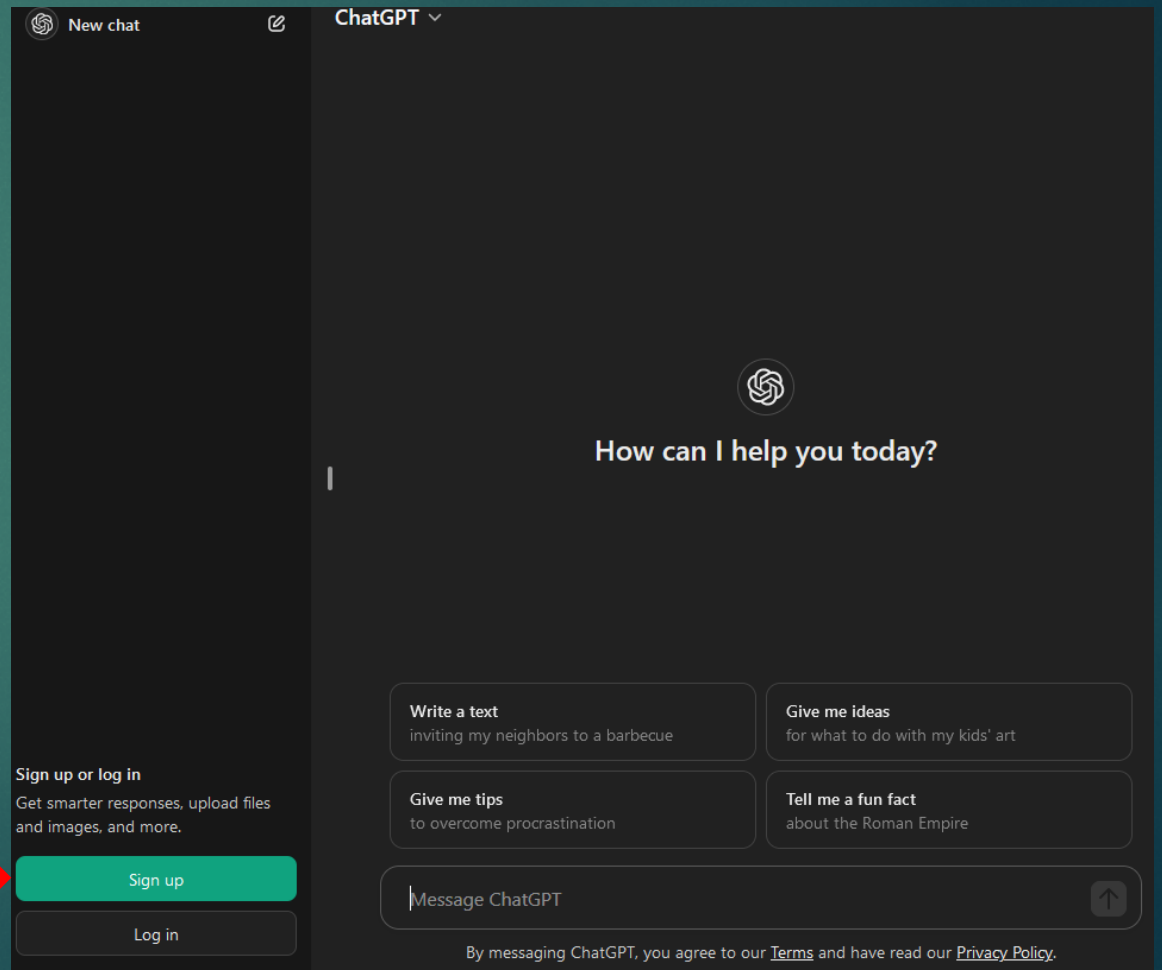
ChatGPT - Limitations

- A limitation in all AI systems are the quality of the training data. Despite the large training data set, it is not "knowledgeable" about many events, especially recent ones.
- ChatGPT tends not to be able to provide in depth information, sticking to summaries and short answers.
- Since no curated data set can be unbiased, each iteration of ChatGPT is criticized for its perceived political bias when discussing politicians, world events, and public figures.
- As more training and human feedback are always being given to the model, these issues are ever-changing and not consistent.

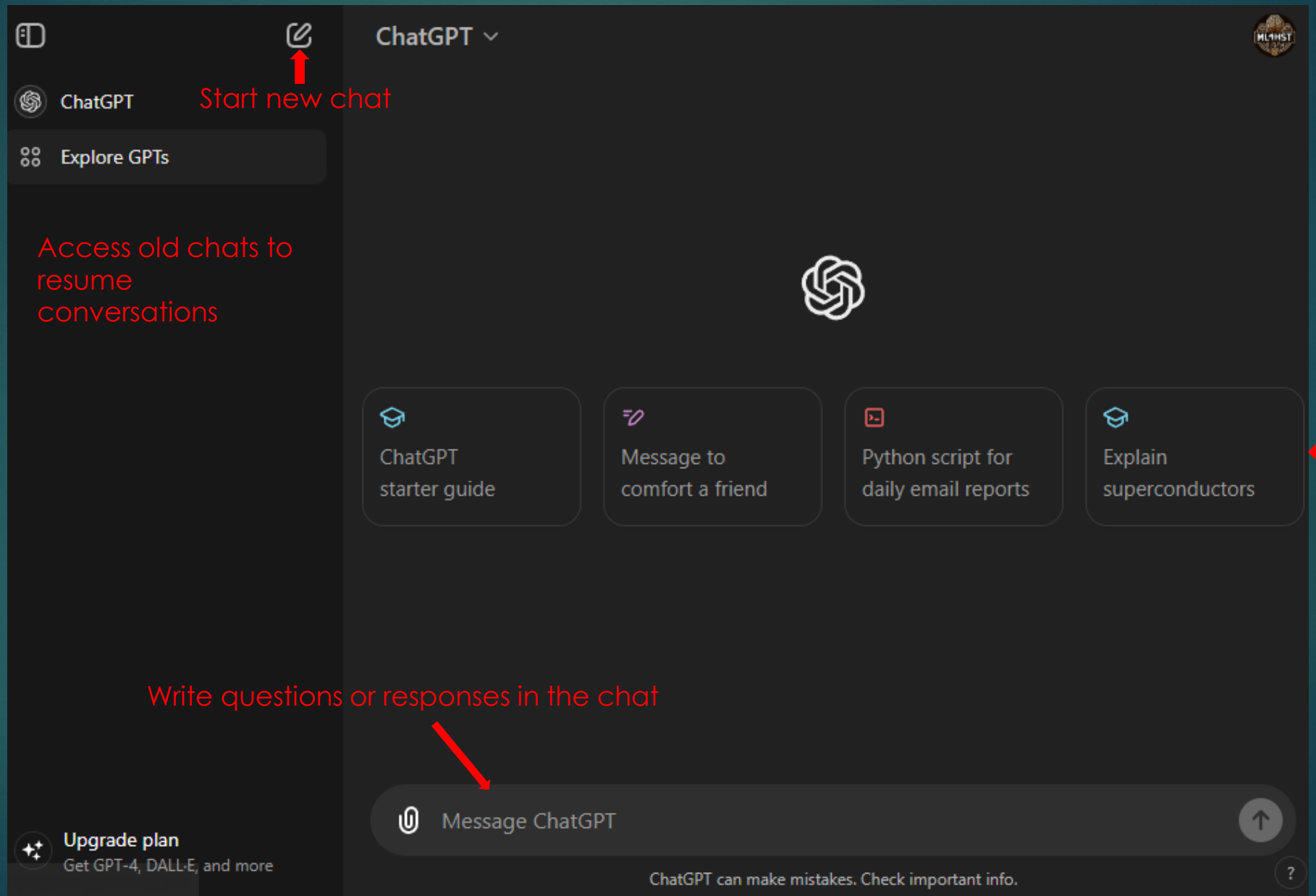
ChatGPT – How to Use

- Go to https://openai.com/index/chatgpt/ and sign up for an account (can use Gmail account).
- After confirming, you can access the service from the same page by logging in.





ChatGPT – How to Use

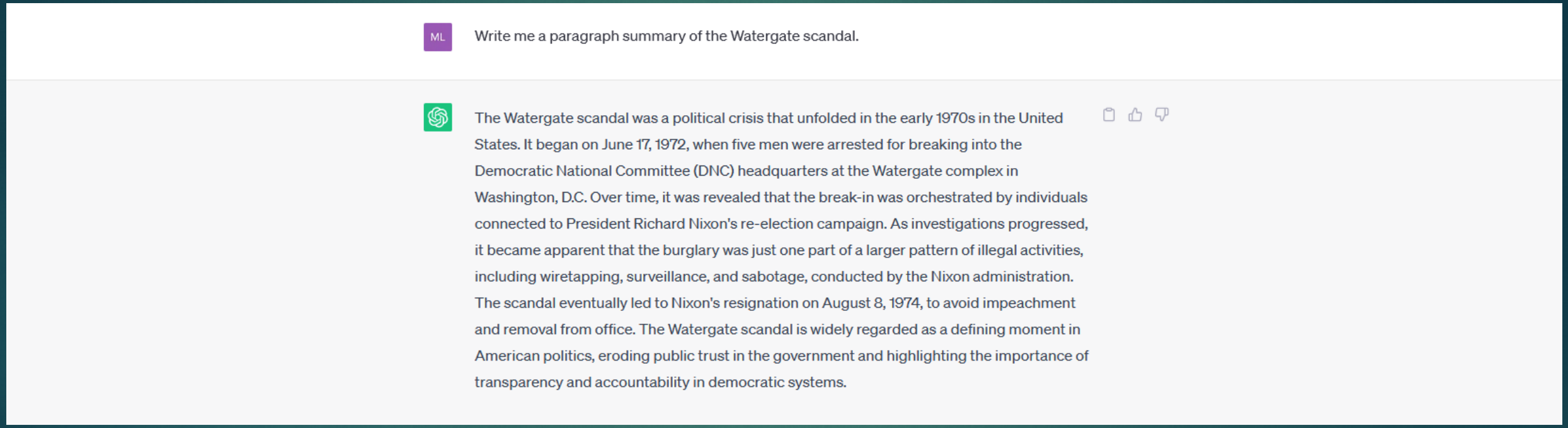


ChatGPT – Education

- Students can utilize ChatGPT for the following purposes:
  - Quickly generating text for writing assignments based on prompt
    - Text summarization
    - Essay writing
  - Getting step-by-step solutions to math problems
  - Use instead of or along with a search engine as a information gathering tool.
- ChatGPT and similar AI tools are a paradigm shifting technology, and their impact on many sectors, including education, will be equally unpredictable and profound.
- One thing we can be sure of; AI tools like ChatGPT are only going to become more widespread.
- It is important for you to know the capabilities and limitations so you can use it as **a tool and not a crutch.**

ChatGPT – Writing

- ChatGPT can be used for generating writing based on a simple prompt



- For most well-documented subject matter, ChatGPT can be quite coherent and factually correct.
- Try a couple prompts of your own.

ChatGPT – Writing

- There are a few (imperfect) tools, also based on AI models, which purport to detect whether text was written by a language model, including OpenAI's own AI text classifier and the third party ZeroGPT.
- The next slide shows ZeroGPT's assessment on three writing samples.
- Note that these detection tools are lagging behind the language models.
- My own written summary was still flagged as likely to contain AI generated content, so these measures are not perfect yet.
- Try to copy and paste a ChatGPT response you generated into ZeroGPT.

ChatGPT – Writing

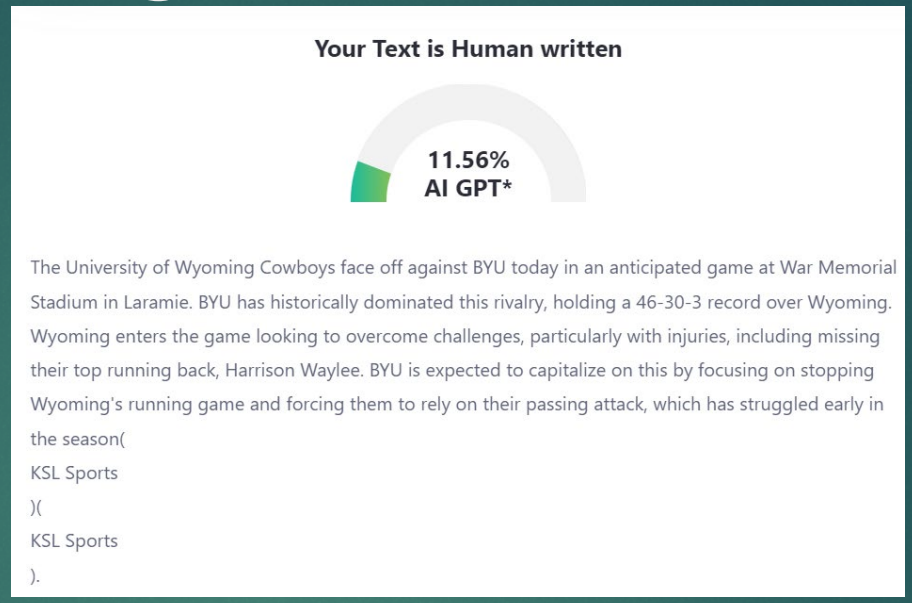
Summary generated by ChatGPT

give me insights of todays football game between UW and BYU

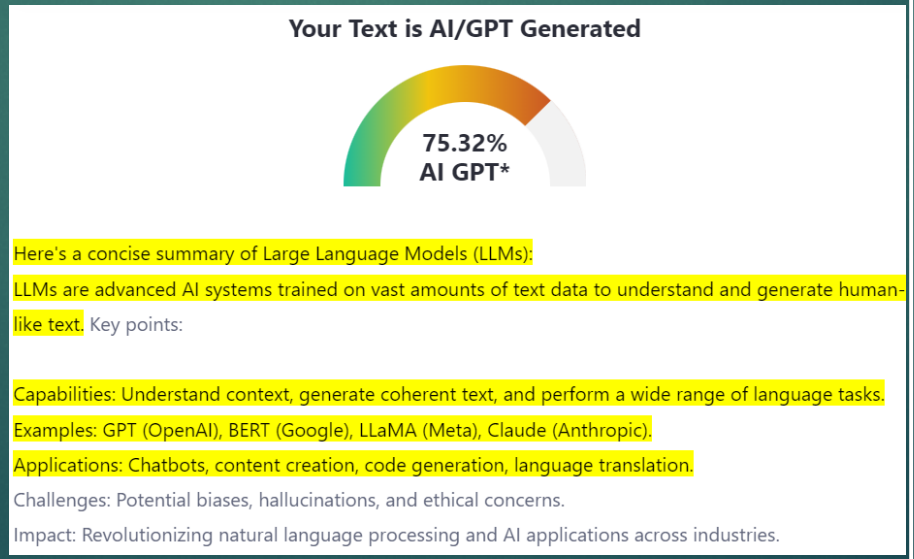
Searched 3 sites

The University of Wyoming Cowboys face off against BYU today in an anticipated game at War Memorial Stadium in Laramie. BYU has historically dominated this rivalry, holding a 46-30-3 record over Wyoming. Wyoming enters the game looking to overcome challenges, particularly with injuries, including missing their top running back, Harrison Waylee. BYU is expected to capitalize on this by focusing on stopping Wyoming's running game and forcing them to rely on their passing attack, which has struggled early in the season(KSL Sports)(KSL Sports).

Kickoff is at 7 PM MT, and the game will be broadcast on CBS Sports Network. Wyoming's defensive strength, particularly on the line, could make this a competitive matchup(KSL Sports).



Summary written by ChatGPT is classified as Human written text



Summary written by me. I tried to replicate the professional and neutral tone of ChatGPT.

ChatGPT – Writing

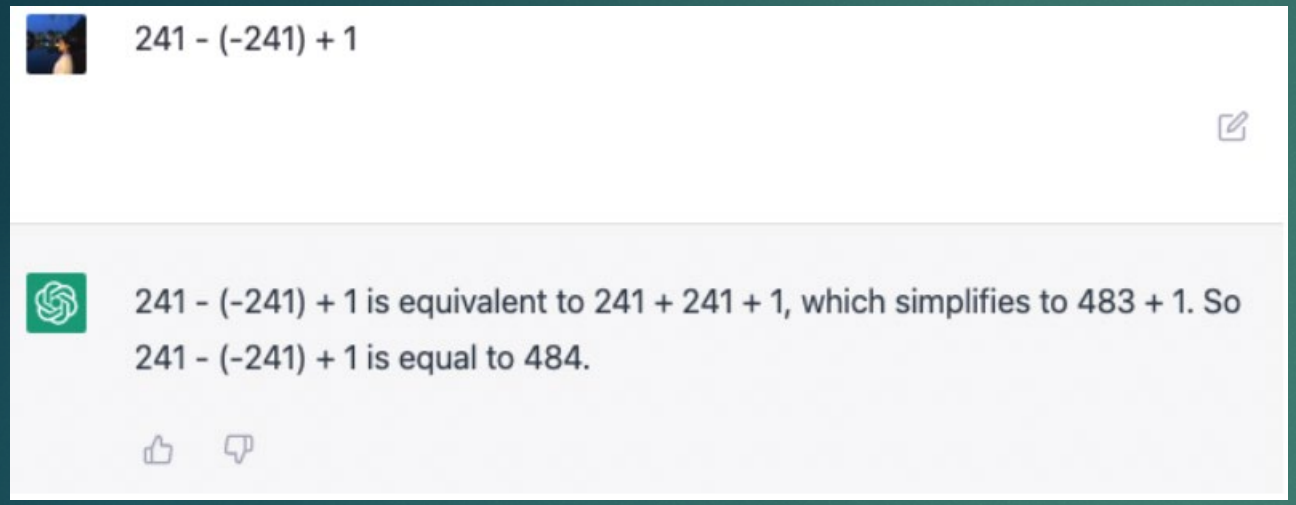
- **Tips:**
  - Explore ChatGPT by messing around with it. Ask it questions on a variety of topics, factual and subjective, and get a sense of its tone, capabilities, and limitations.
  - You could think of ChatGPT as a system which gives a response which is the "statistical consensus" of its training data set. Thus, ChatGPT tends to speak in a formal tone, often restates part of the prompt given to it, and is rather ambivalent on most opinionated topics. This makes generated text differentiable from that written by a student.
  - While you likely will not find much success writing entire assignments with AI tools, these tools are very valuable for gathering information. Ethical usage is key when it comes to your education.

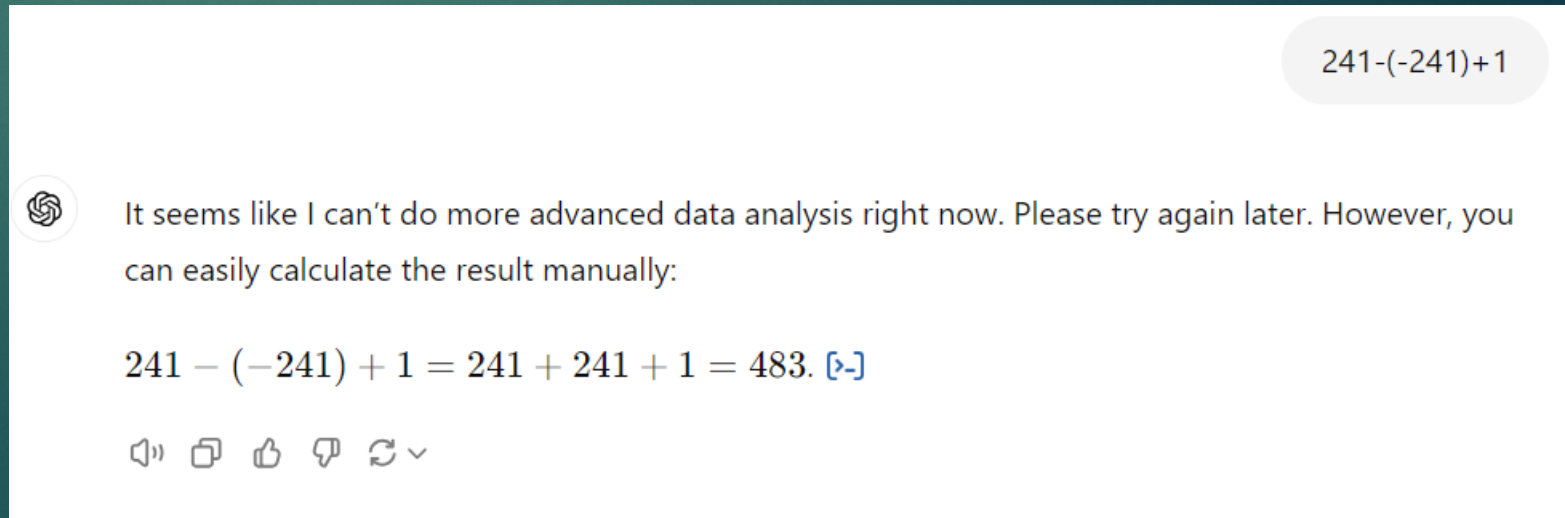
ChatGPT – Math

- As its primary use is as a natural language processor, it only "understands" quantitative reasoning in so far as it understands logic from semantics.
- ChatGPT has mixed results when it comes to solving math problems.
- While it can sometimes answer a question with impressive insight, it can also fail with equal confidence.

ChatGPT – Math

- Response of different versions of ChatGPT to solve a system of math problems:





ChatGPT – Math

- Here I ask ChatGPT to find derivative of a function:

find the derivative of this function f(x) = 3x*2+2x-1

The derivative of the function $f(x) = 3x^2 + 2x - 1$ is $f'(x) = 6x + 2$

Analysis

python

```
import sympy as sp

# Define the variable and function
x = sp.symbols('x')
f = 3*x**2 + 2*x - 1

# Compute the derivative
derivative_f = sp.diff(f, x)
derivative_f
```

Always show details | Copy code

Result

```
6*x + 2
```

find the derivative of this function with complete process f(x) = 3x*2+2x-1

To find the derivative of the function $f(x) = 3x^2 + 2x - 1$, we'll use basic rules of differentiation. Here's the step-by-step process:

Given Function:

$$f(x) = 3x^2 + 2x - 1$$

Step 1: Apply the Power Rule

The power rule states that if $f(x) = ax^n$, then $f'(x) = n \cdot ax^{n-1}$.

1. **Derivative of $3x^2$:**
   - Apply the power rule where $n = 2$.

$$\frac{d}{dx}(3x^2) = 2 \cdot 3x^{2-1} = 6x$$

2. **Derivative of $2x$:**
   - This is a linear term, and the derivative of ax is simply a.

$$\frac{d}{dx}(2x) = 2$$

3. **Derivative of -1:**
   - The derivative of a constant is always zero.

$$\frac{d}{dx}(-1) = 0$$

Step 2: Combine the Results

Now, we sum the individual derivatives:

$$f'(x) = 6x + 2 + 0 = 6x + 2$$

Final Answer:

The derivative of the function $f(x) = 3x^2 + 2x - 1$ is:

$$f'(x) = 6x + 2$$

ChatGPT – Math

- Here I ask ChatGPT to find derivative of a function:

write me python code to solve derivative of f(x) = 3x*2+2x-1

Here is a Python code using the `sympy` library to compute the derivative of $f(x) = 3x^2 + 2x - 1$:

python Copy code

```python
import sympy as sp

# Define the variable and the function
x = sp.symbols('x')
f = 3*x**2 + 2*x - 1

# Compute the derivative
derivative_f = sp.diff(f, x)

# Display the derivative
print(f"The derivative of the function is: {derivative_f}")
```

ChatGPT – Math

- However, when asked to calculate *2 x (cos(5π/8) – 1)*:

what is 2*(cos(5*pi/8)-1)?

Let's break down the calculation $2 \times (\cos(5\pi/8) - 1)$.

1. **Calculate** $\cos(5\pi/8)$:
   $\cos(5\pi/8) \approx -0.3827$.
2. **Subtract 1 from** $\cos(5\pi/8)$:
   $\cos(5\pi/8) - 1 \approx -0.3827 - 1 = -1.3827$.
3. **Multiply by 2:**
   $2 \times -1.3827 = -2.7654$.

Thus, $2 \times (\cos(5\pi/8) - 1) \approx -2.7654$.

1 $\cos\left(\frac{5\pi}{8}\right)$ = -0.382683432365

2 $2\left(\cos\left(\frac{5\pi}{8}\right)-1\right)$ = -2.76536686473

ML What is 2 * (cos(5 * pi / 8) - 1)?

To evaluate the expression 2 * (cos(5 * π / 8) - 1), we can use the properties of trigonometric functions and basic arithmetic operations.

First, let's compute the value of cos(5 * π / 8). Since π radians is equal to 180 degrees, we can convert 5 * π / 8 to degrees:

5 * π / 8 ≈ 180 * (5 / 8) ≈ 112.5 degrees

Next, we calculate the cosine of 112.5 degrees. Note that cosine is a periodic function with a period of 360 degrees, so cos(112.5 degrees) is equivalent to cos(112.5 degrees + 360 degrees):

cos(112.5 degrees) = cos(112.5 degrees + 360 degrees) ≈ cos(472.5 degrees)

Since cos(x) = cos(360 degrees - x), we can rewrite cos(472.5 degrees) as cos(360 degrees - 472.5 degrees):

cos(472.5 degrees) = cos(360 degrees - 472.5 degrees) = cos(-112.5 degrees)

The cosine function has symmetry around the y-axis, which means cos(-x) = cos(x). Therefore, cos(-112.5 degrees) is equal to cos(112.5 degrees).

Now, we can evaluate the cosine value using a calculator or trigonometric tables to find:

cos(112.5 degrees) ≈ 0.92388 1

Substituting this value back into the original expression:

2 * (cos(5 * π / 8) - 1) = 2 * (0.92388 - 1) = 2 * (-0.07612) = -0.15224

So, 2 * (cos(5 * π / 8) - 1) is approximately equal to -0.15224. 2

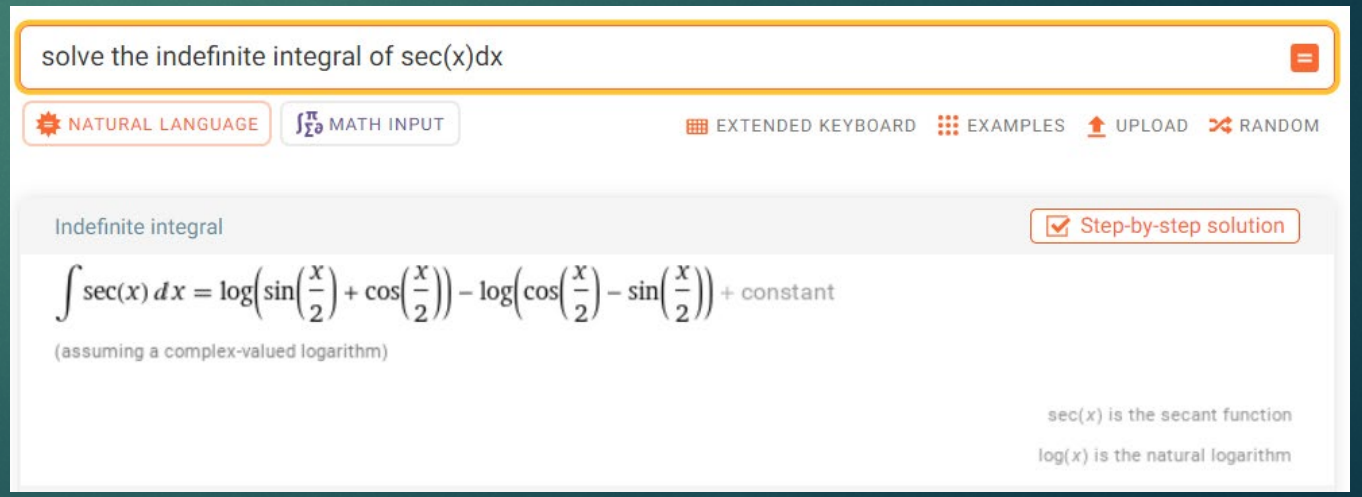
ChatGPT – Math

- Now asking ChatGPT to solve the indefinite integral $\int \sec(x)\,dx$
- However, this is not equal to the answer given by WolframAlpha.
- While ChatGPT has some level of mathematical sophistication, it struggles with true logical reasoning, and cannot calculate expressions correctly.

Final Answer:

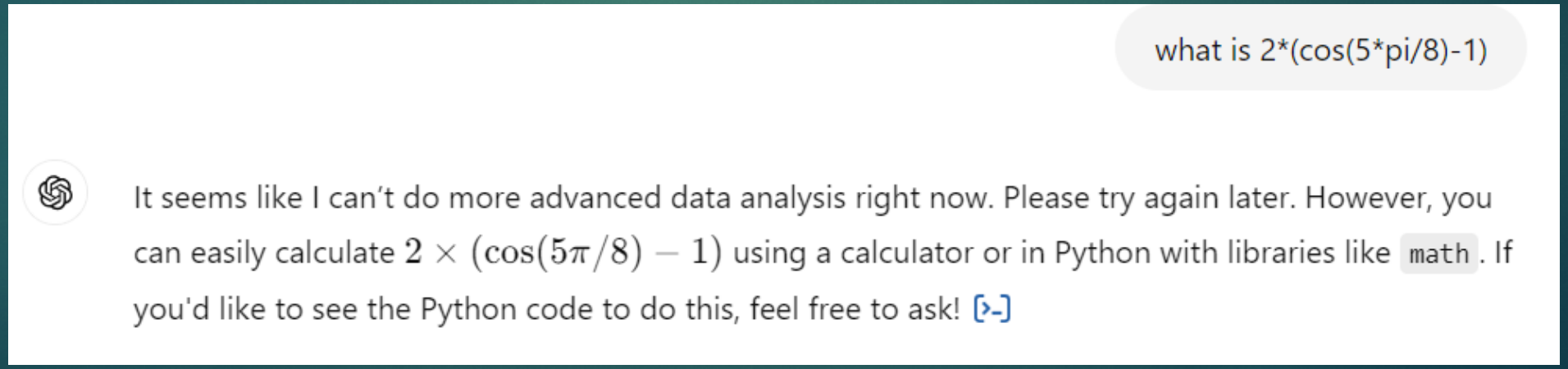
$$\int \sec(x)\,dx = \ln|\sec(x) + \tan(x)| + C$$

This is the indefinite integral of $\sec(x)\,dx$.



ChatGPT – Math

- **Tips:**
  - ChatGPT will/will not tell the user that it doesn't know how to solve a problem, some time it will just make things up that **resemble mathematical logic.**
  - Many existing tools, such as WolframAlpha and Symbolab, already offer a wide variety of free and paid assistance in solving a variety of math problems,
  - However, **ChatGPT is quite good at summarizing mathematical concepts**, and can be prompted to rephrase or reduce the assumed expertise of the audience.



ChatGPT – Math

- Prompting ChatGPT for an explanation on a mathematical concept tailored to different audiences

Explain the difference between rational and irrational numbers

Rational and irrational numbers are two distinct types of real numbers, each with specific properties. Here's a breakdown of their differences:

1. Rational Numbers:

- **Definition**: Rational numbers are numbers that can be expressed as the ratio (or fraction) of two integers $\frac{a}{b}$, where a and b are integers, and $b \neq 0$.
- **Examples**: $\frac{1}{2}$, $\frac{-3}{4}$, 5 (since $5 = \frac{5}{1}$), 0 (since $0 = \frac{0}{1}$).
- **Decimal Representation**: Rational numbers either terminate (e.g., 0.5) or repeat in a pattern (e.g., 0.333...).

2. Irrational Numbers:

- **Definition**: Irrational numbers cannot be expressed as a fraction of two integers. Their decimal representations are non-terminating and non-repeating.
- **Examples**: π, $\sqrt{2}$, e (Euler's number).
- **Decimal Representation**: Irrational numbers have infinite decimal places without any repeating pattern. For instance, $\pi \approx 3.141592653\ldots$ goes on forever without repeating.

Key Differences:

1. **Form**: Rational numbers can be written as a fraction of two integers, whereas irrational numbers cannot.
2. **Decimal Representation**: Rational numbers have either a terminating or repeating decimal expansion, while irrational numbers have a non-terminating, non-repeating decimal expansion.
3. **Examples**: $\frac{3}{4}, 0.75$ are rational, while $\pi, \sqrt{3}$ are irrational.

Common Misconception:

- Some may think that numbers with long decimals are irrational, but a repeating pattern (e.g., 0.666...) still makes the number rational. What makes a number irrational is the absence of any pattern in its decimal expansion.

Explain the difference between rational and irrational numbers to an audience of Junior High School Math Students with Limited knowledge of the subject

Let's talk about **rational** and **irrational** numbers in a simple way.

Rational Numbers:

- Think of **rational** numbers as "friendly" numbers because they can be written as a **fraction**. A fraction is when you have one number on top of another, like $\frac{3}{4}$.
- The top number (numerator) and the bottom number (denominator) are **whole numbers**, and the bottom number can't be zero.
- **Examples**: Numbers like $\frac{1}{2}$, $\frac{3}{5}$, and even whole numbers like 7 (since $7 = \frac{7}{1}$) are all rational.
- Rational numbers can also be written as **decimals**. These decimals might **stop** (like 0.5) or **repeat** in a pattern (like 0.3333...).

Irrational Numbers:

- Now, **irrational** numbers are the "mysterious" numbers because they **cannot** be written as a fraction. They are special because their **decimals never end** and they **never repeat**.
- For example, have you heard of π (pi)? It's about 3.14, but the numbers after the decimal go on forever and don't repeat! That makes it irrational.
- Another example is $\sqrt{2}$ (the square root of 2). When you put it in a calculator, it gives you a long decimal that keeps going, and it doesn't follow a pattern. That's why it's irrational.

Simple Difference:

- **Rational numbers** can be written as fractions (like $\frac{3}{4}$ or 0.75), and their decimals either stop or repeat.
- **Irrational numbers** can't be written as fractions, and their decimals go on forever without repeating.

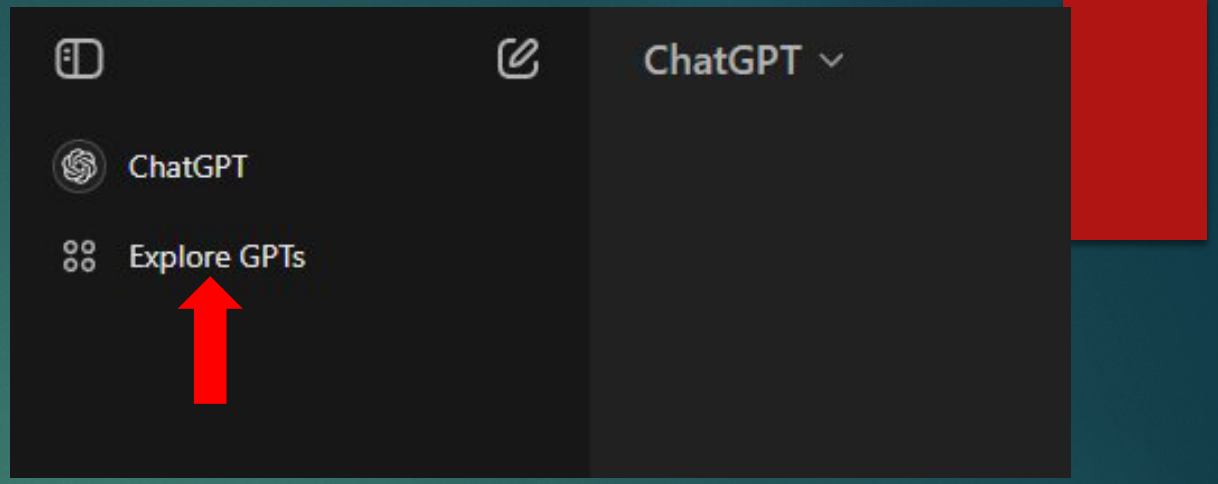
Example:

- **Rational**: $\frac{1}{2} = 0.5$ (the decimal stops).
- **Irrational**: $\pi \approx 3.14159\ldots$ (the decimal keeps going and never repeats).

ChatGPT – Coding

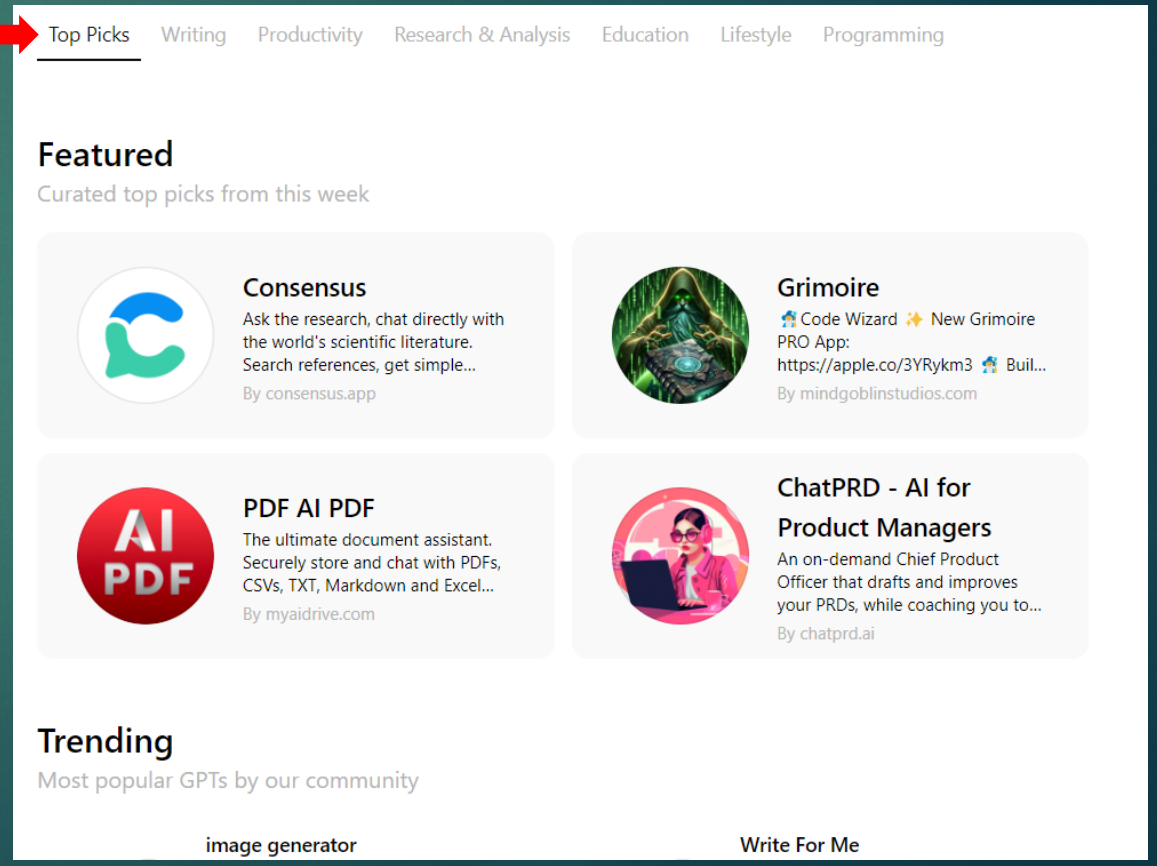
- ChatGPT can generate code in most programming languages
  - Built in code interpreter
  - Trained the same as with any other domain
- Pick a module and experiment writing prompts to generate code
- Be specific about what you want and how
- Suggest changes and see how it responds

Explore GPTs



- There are many custom GPTs available
- All use ChatGPT, but are trained on more specific content areas
- Many have specific prompts
- Pick a couple to try out

Categories

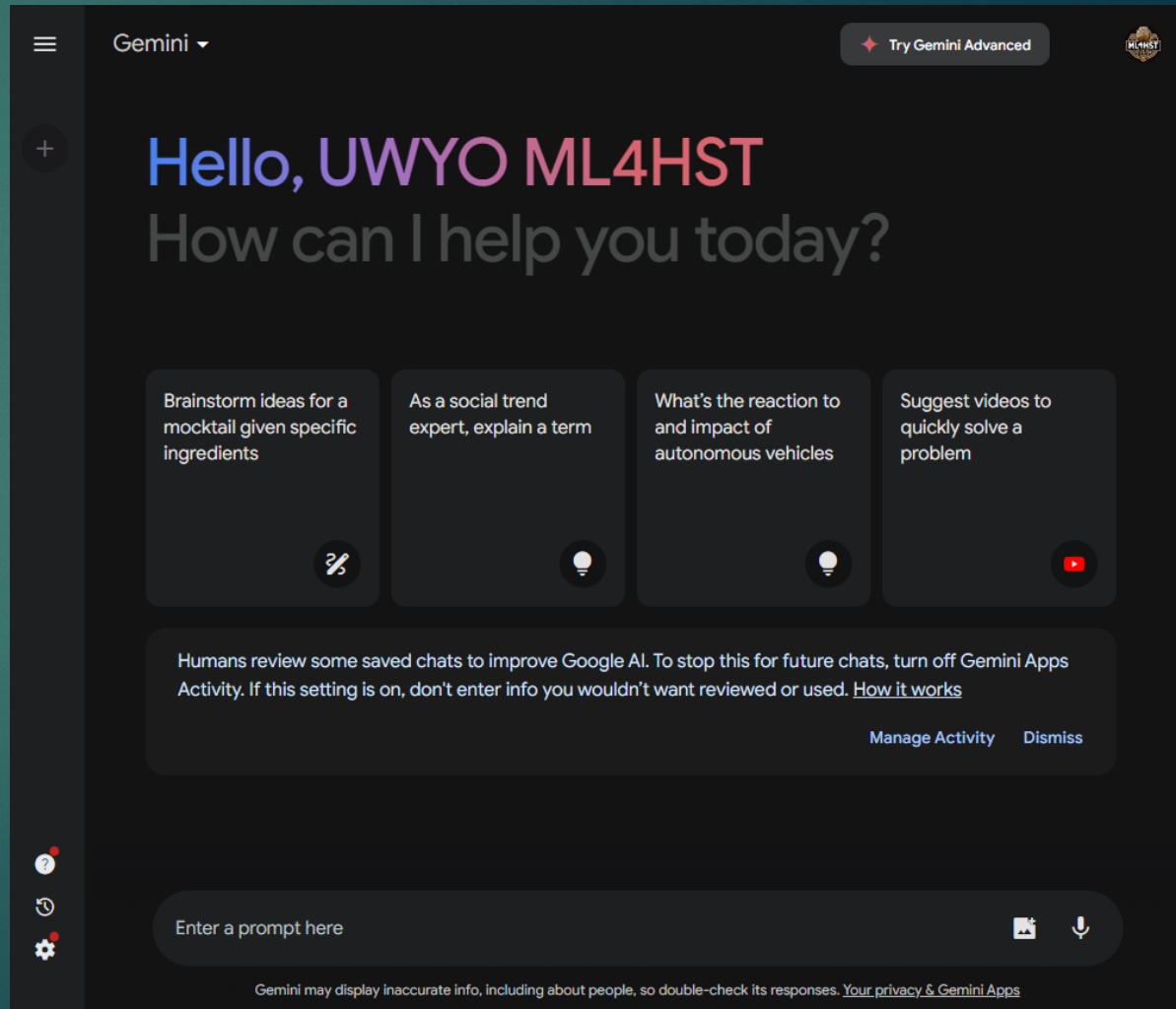


Google Gemini

- Similar to ChatGPT, primarily built for research
  - Built in tools for fact verification using Google search results
- Trained on specific use cases for industrial applications
- Can also generate images

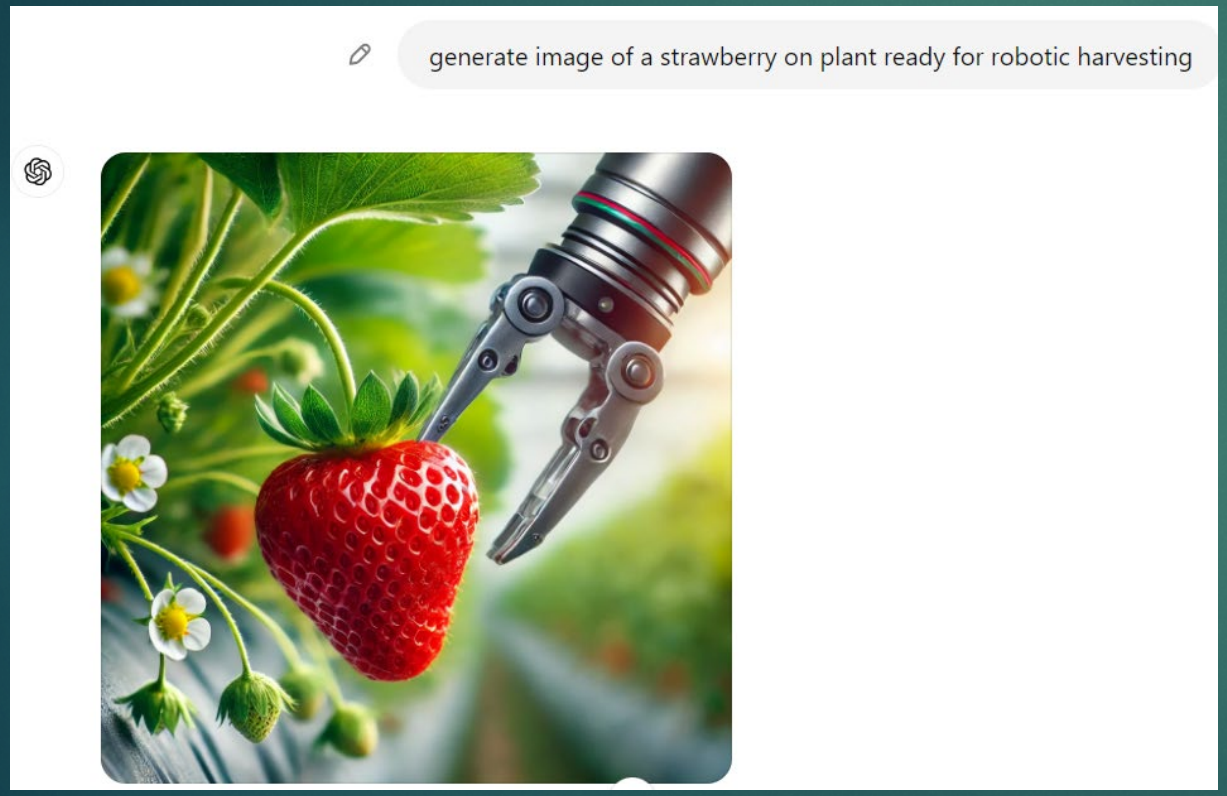
Using Google Gemini

- Go to: https://gemini.google.com/app
- Login using your Gmail account, as you did for ChatGPT
- Write prompts as you have in ChatGPT
  - Try comparing responses given the same prompt
- Try the image generation
  - Ex: "generate an image of a computer"

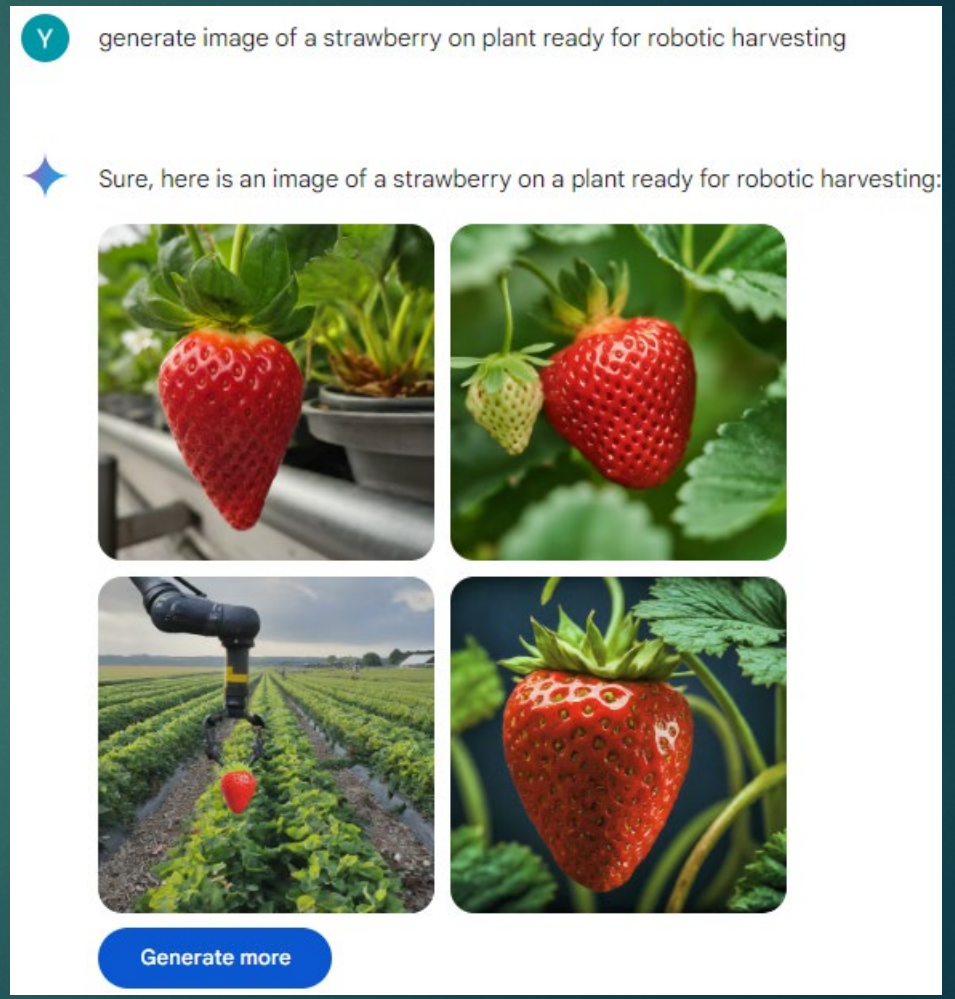


Using Google Gemini

- Image Generation GPT vs Gemini
- Latest versions of ChatGPT leverages DALL.E 3 to generate images



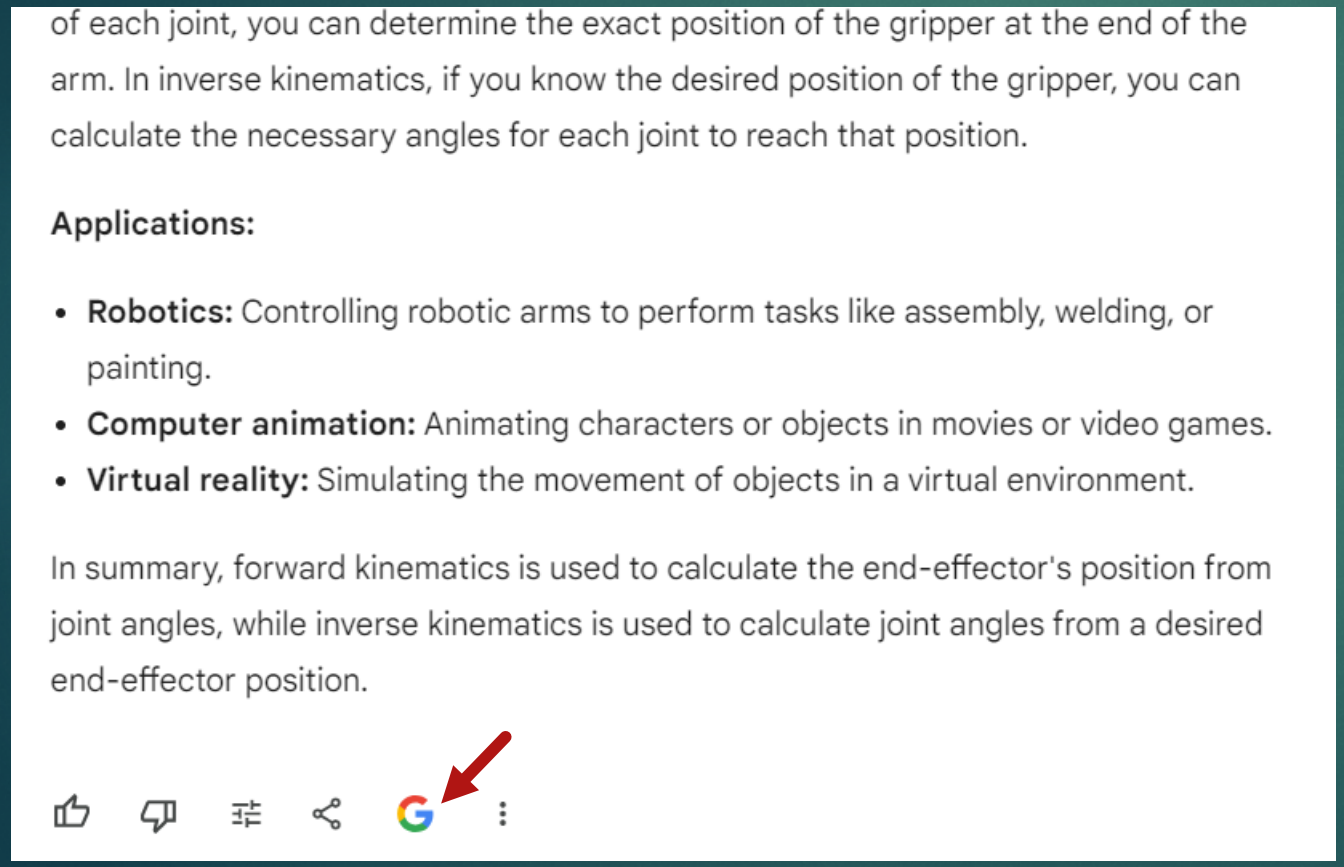
GPT

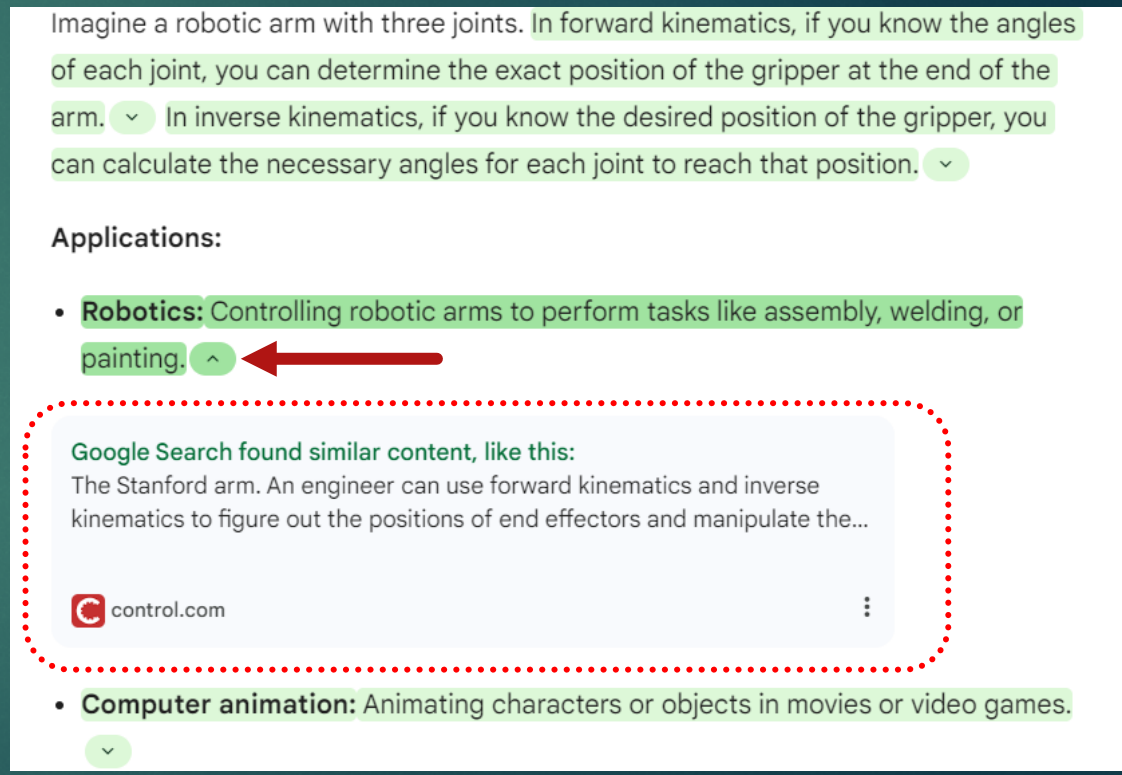


Gemini

Fact Verification

- Use Gemini's double-check response feature by clicking on the Google logo at the end of your generated response
- Then, expand highlighted text

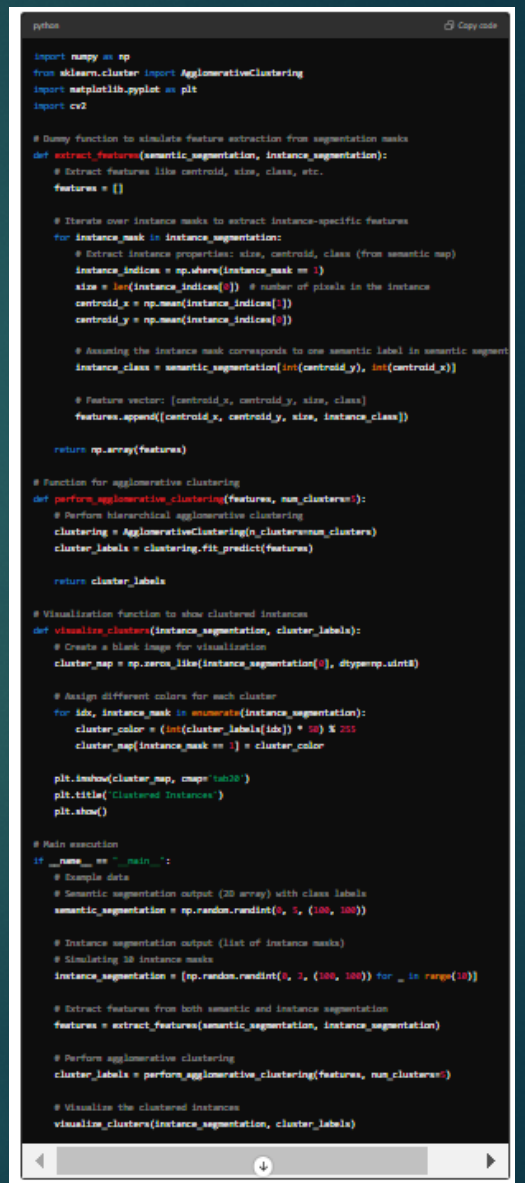




Code Generation

- ChatGPT vs Gemini
  - Prompt sentence “write me a code for agglomerative hierarchical clustering using input from semantic segmentation and instance segmentation outputs”

GPT

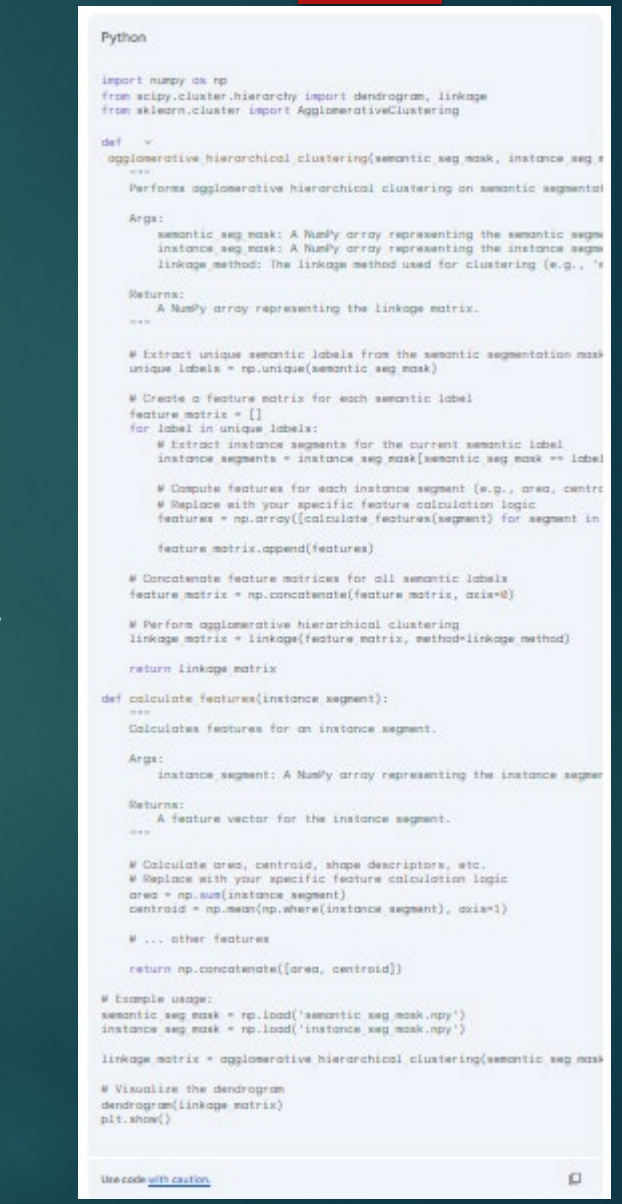


```python
import numpy as np
from sklearn.cluster import AgglomerativeClustering
import matplotlib.pyplot as plt
import cv2

# Dummy function to simulate feature extraction from segmentation masks
def extract_features(semantic_segmentation, instance_segmentation):
    # Extract features like centroid, size, class, etc.
    features = []

    # Iterate over instance masks to extract instance-specific features
    for instance_mask in instance_segmentation:
        # Extract instance properties: size, centroid, class (from semantic map)
        instance_indices = np.where(instance_mask == 1)
        size = len(instance_indices[0])  # number of pixels in the instance
        centroid_x = np.mean(instance_indices[1])
        centroid_y = np.mean(instance_indices[0])

        # Assuming the instance mask corresponds to one semantic label in semantic segment
        instance_class = semantic_segmentation[int(centroid_y), int(centroid_x)]

        # Feature vector: [centroid_x, centroid_y, size, class]
        features.append([centroid_x, centroid_y, size, instance_class])

    return np.array(features)

# Function for agglomerative clustering
def perform_agglomerative_clustering(features, num_clusters=5):
    # Perform hierarchical agglomerative clustering
    clustering = AgglomerativeClustering(n_clusters=num_clusters)
    cluster_labels = clustering.fit_predict(features)

    return cluster_labels

# Visualization function to show clustered instances
def visualize_clusters(instance_segmentation, cluster_labels):
    # Create a blank image for visualization
    cluster_map = np.zeros_like(instance_segmentation[0], dtype=np.uint8)

    # Assign different colors for each cluster
    for idx, instance_mask in enumerate(instance_segmentation):
        cluster_color = (int(cluster_labels[idx]) * 50) % 255
        cluster_map[instance_mask == 1] = cluster_color

    plt.imshow(cluster_map, cmap='tab20')
    plt.title('Clustered Instances')
    plt.show()

# Main execution
if __name__ == "__main__":
    # Example data
    # Semantic segmentation output (2D array) with class labels
    semantic_segmentation = np.random.randint(0, 5, (100, 100))

    # Instance segmentation output (list of instance masks)
    # Simulating 10 instance masks
    instance_segmentation = [np.random.randint(0, 2, (100, 100)) for _ in range(10)]

    # Extract features from both semantic and instance segmentation
    features = extract_features(semantic_segmentation, instance_segmentation)

    # Perform agglomerative clustering
    cluster_labels = perform_agglomerative_clustering(features, num_clusters=5)

    # Visualize the clustered instances
    visualize_clusters(instance_segmentation, cluster_labels)
```

Code Generation

- ChatGPT vs Gemini
  - Prompt sentence “write me a code for agglomerative hierarchical clustering using input from semantic segmentation and instance segmentation outputs”

Gemini



```
Python

import numpy as np
from scipy.cluster.hierarchy import dendrogram, linkage
from sklearn.cluster import AgglomerativeClustering

def
 agglomerative_hierarchical_clustering(semantic_seg_mask, instance_seg_
    """
    Performs agglomerative hierarchical clustering on semantic segmentat

    Args:
        semantic_seg_mask: A NumPy array representing the semantic segme
        instance_seg_mask: A NumPy array representing the instance segme
        linkage_method: The linkage method used for clustering (e.g., '

    Returns:
        A NumPy array representing the linkage matrix.
    """

    # Extract unique semantic labels from the semantic segmentation mask
    unique_labels = np.unique(semantic_seg_mask)

    # Create a feature matrix for each semantic label
    feature_matrix = []
    for label in unique_labels:
        # Extract instance segments for the current semantic label
        instance_segments = instance_seg_mask[semantic_seg_mask == label

        # Compute features for each instance segment (e.g., area, centr
        # Replace with your specific feature calculation logic
        features = np.array([calculate_features(segment) for segment in

        feature_matrix.append(features)

    # Concatenate feature matrices for all semantic labels
    feature_matrix = np.concatenate(feature_matrix, axis=0)

    # Perform agglomerative hierarchical clustering
    linkage_matrix = linkage(feature_matrix, method=linkage_method)

    return linkage_matrix

def calculate_features(instance_segment):
    """
    Calculates features for an instance segment.

    Args:
        instance_segment: A NumPy array representing the instance segmen

    Returns:
        A feature vector for the instance segment.
    """

    # Calculate area, centroid, shape descriptors, etc.
    # Replace with your specific feature calculation logic
    area = np.sum(instance_segment)
    centroid = np.mean(np.where(instance_segment), axis=1)

    # ... other features

    return np.concatenate([area, centroid])

# Example usage:
semantic_seg_mask = np.load('semantic_seg_mask.npy')
instance_seg_mask = np.load('instance_seg_mask.npy')

linkage_matrix = agglomerative_hierarchical_clustering(semantic_seg_mas

# Visualize the dendrogram
dendrogram(linkage_matrix)
plt.show()

Use code with caution.
```

Similar Tools

- **Claudi** from Anthropic
- **Meta AI** by Meta
- **Microsoft Copilot** by Microsoft
- And many more
- Specialized LLMs for code generation
  - OpenAI Codex
  - Github Copilot
  - AlphaCode
  - InCoder
- Specialized for Image Generation
  - DALL.E 3
  - Stable Diffusion
  - Imagen
  - Deep Dream