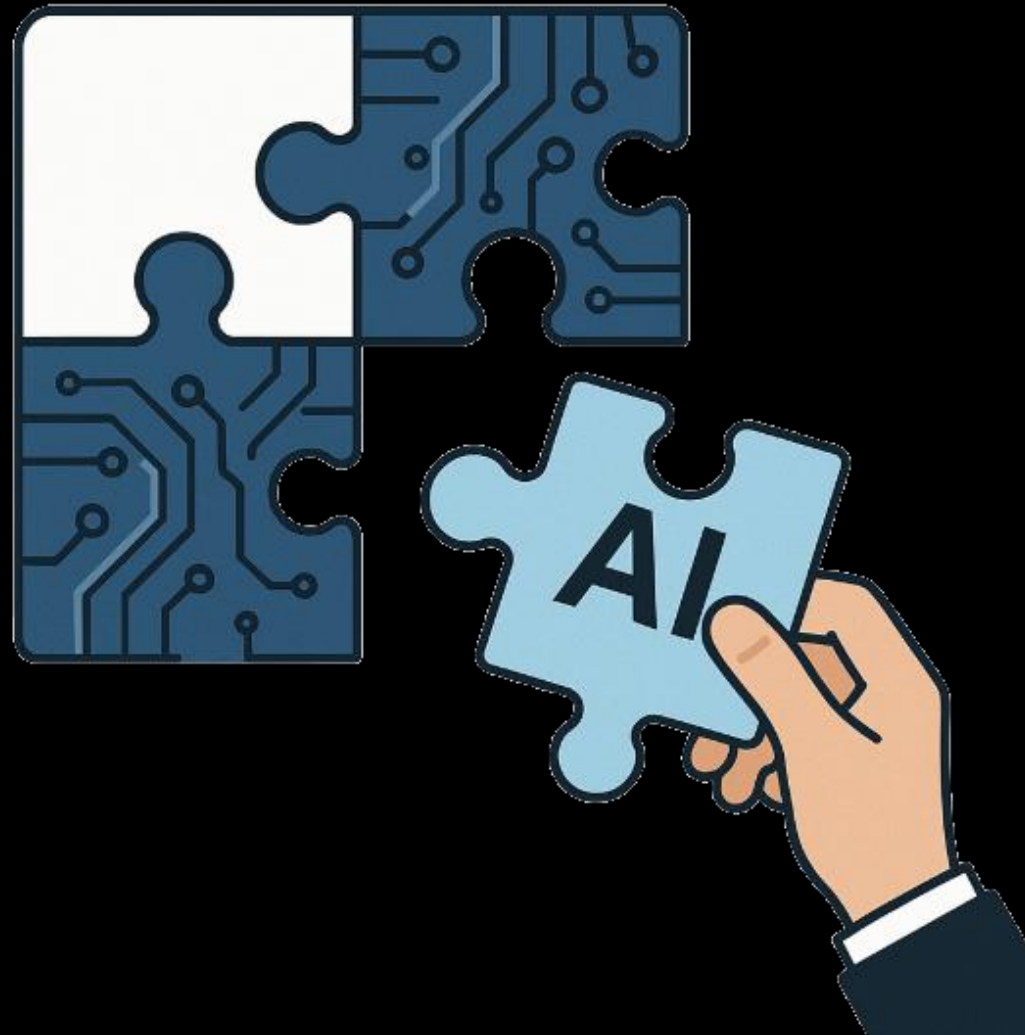


# Session 2

## Yotam 5/12



You're not making the puzzle pieces — you're putting them together!



You're not making swords — you're mastering swordsmanship!



# Writing effective prompts

## Best practices

- 1 **Be specific** about what you want Copilot/ChatGPT to do. Clear goals lead to better responses.
- 2 **Add some context** to help Copilot/ChatGPT understand what you're asking. Context makes the response more relevant.
- 3 **Provide some examples** for Copilot/ChatGPT to use. This helps ground the response in the right context.
- 4 Let Copilot/ChatGPT know how you want the response **to be formatted**. This sets clear expectations.

## The art of the prompt

Goal	What do you want from Copilot?
Context	Why do you need it and who is involved?
Source(s)	What information or samples do you want Copilot to use?
Expectations	How should Copilot respond to best fulfill your request?

# What's Next Thinking



What is the Prompt?

# Practicing Progressive Prompts

Task	Option A (OK)	(Better)	Option B (Even Better)
<b>Brainstorm newsletter topics</b>	I need ideas for our company newsletter. Can you suggest some topics? <hr/>	I'm working on our company newsletter. Can you suggest topics related to Efficiency tips? <hr/>	I'm working on our company newsletter. Can you suggest topics related to efficiency tips for data analysts? Topics should include a concise headline and a short summary that highlights its relevance to data science. <hr/>

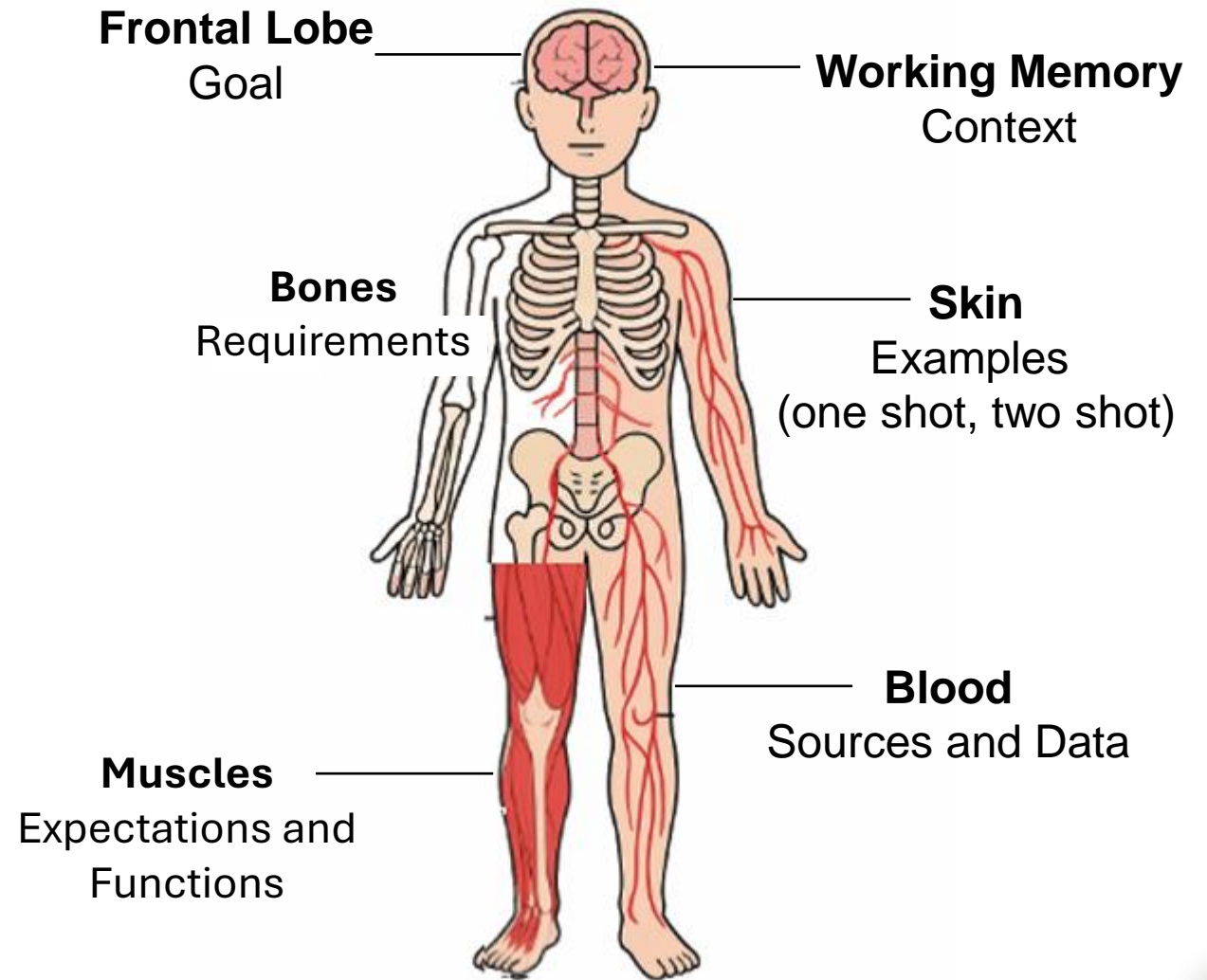




Think of a prompt as the  
human body

Goal  
Context  
Requirements  
Expectations & Functions  
Data & Sources  
Examples

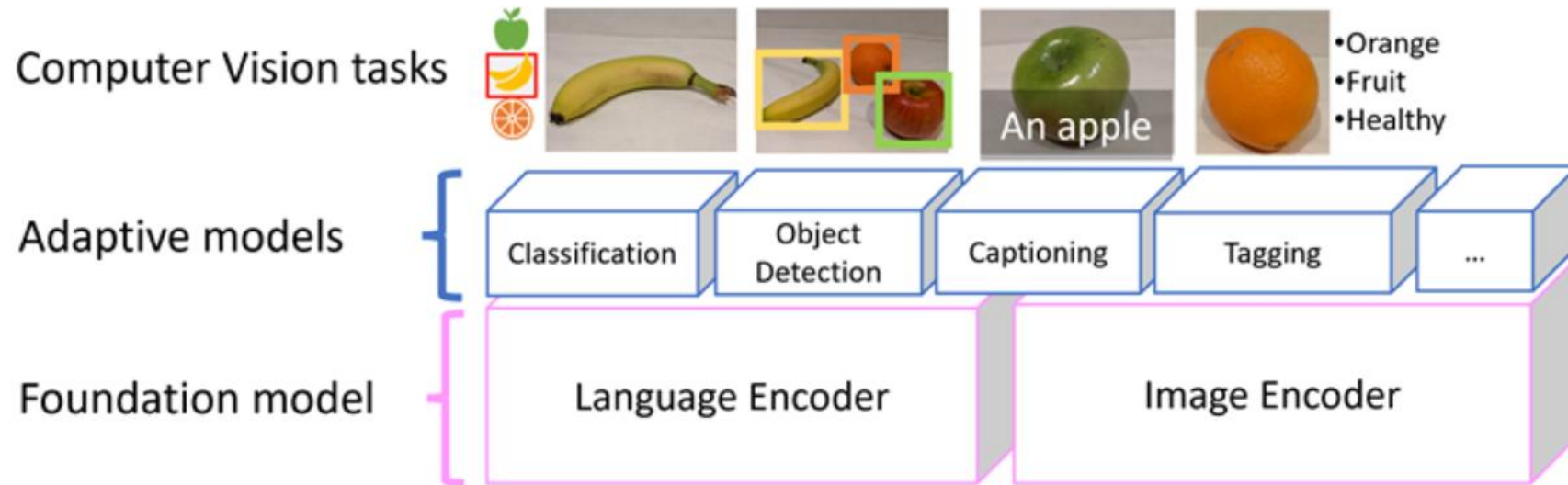
# Prompt Engineering





# The Big Surprise

Multi-modal models (Surprise)



- A newer approach to modeling involves combining language and vision models that encode image and text data
- The model encapsulates semantic relationships between features extracted from the images and text extracted from related captions.
- A multi-modal model can be used as a *foundation* model for more specialized *adaptive* models.

# Use Case



A local hardware store faces significant challenges in managing the billing process for customers. Currently, the store uses a manual checkout system, which is time-consuming, prone to human errors, and inefficient. This often results in customer dissatisfaction and potential revenue loss due to mismanagement or inaccurate billing.

To address this, the hardware store is exploring the implementation of a basic Cart Management System as a pilot solution to improve the purchasing experience. The system does not need to be overly complex but should provide essential functionalities to streamline operations and enhance customer satisfaction.



# Example of a multi-modal prompt



**Goal and Context:** The primary goal is to create a program that digitally streamlines the billing process for a local hardware store that currently relies on manual methods, increasing efficiency, enhancing customer satisfaction, reducing operational errors, and laying the foundation for future growth.

## Data

```
Inventory = [  
    (101, "Hammer", 120),  
    (102, "Screwdriver Set", 350),  
    (103, "Drill Machine", 2500),  
    (104, "Pliers", 180),  
    (105, "Wrench", 220),  
]  
  
My_Cart = []
```

## Requirements:

**Add Items to the Cart:** The store clerk should be able to select an item from the inventory and add it to the cart. The item's name and price should be pulled automatically based on the selected ID.

**Remove Items from the Cart:** The clerk should have the flexibility to remove individual tools from the cart before checkout to accommodate customer preferences or mistakes.

**Calculate Total Bill:** The system should calculate the total cost of items in the cart and display the final amount clearly to the customer for quick payment.

## Expectations and Functions

The Gradio interface would include: A dropdown to select tools from inventory

An “Add Tool” button A second dropdown to select tools from the current cart

A “Remove Tool” button A display box showing the cart contents A “Calculate Total” button with a result output

## Interface Example

Enter Product ID to Add to Cart	Enter Index ID to Remove from Cart
<input type="text" value="0"/>	<input type="text" value="0"/>
<input type="button" value="Add to Cart"/>	<input type="button" value="Remove from Cart"/>
Your Cart	
<div></div>	
Add Item Message	
<div></div>	
Remove Item Message	
<div></div>	
Total Cost	
<div></div>	

# Wireframe (or image) to Webpage

Prompt 1: create an html input page from the image

Prompt 2: separate the html and the css

Prompt 3: change the styles.css file to make the webpage more appealing

Wireframe (or Image)

Registration Form

First Name

Last Name

Birth Date

E-mail

Phone

Company

Country

Mailing Address

Comment

Current Date

Signature

HTML (54 lines)

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>Registration Form</title>
7 <link rel="stylesheet" href="styles.css">
8 </head>
9 <body>
10 <div class="container">
11 <div>Registration Form</div>
12 <form action="/submit-form" method="post">
13 <div class="form-group">
14 <label for="first-name">First Name</label>
15 <input type="text" id="first-name" name="first-name">
16 </div>
17 <div class="form-group">
18 <label for="last-name">Last Name</label>
19 <input type="text" id="last-name" name="last-name">
20 </div>
21 <div class="form-group">
22 <label for="birth-date">Birth Date</label>
23 <input type="date" id="birth-date" name="birth-date">
24 </div>
25 <div class="form-group">
26 <label for="email">E-mail</label>
27 <input type="email" id="email" name="email">
28 </div>
29 <div class="form-group">
30 <label for="phone">Phone</label>
31 <input type="tel" id="phone" name="phone">
32 </div>
33 <div class="form-group">
34 <label for="company">Company</label>
35 <input type="text" id="company" name="company">
36 </div>
37 <div class="form-group">
38 <label for="country">Country</label>
39 <input type="text" id="country" name="country">
40 </div>
41 <div class="form-group">
42 <label for="mailing-address">Mailing Address</label>
43 <input type="text" id="mailing-address" name="mailing-address">
44 </div>
45 <div class="form-group">
46 <label for="comment">Comment</label>
47 <textarea id="comment" name="comment" rows="4"></textarea>
48 </div>
49 <div class="form-group">
50 <label for="current-date">Current Date</label>
51 <input type="date" id="current-date" name="current-date">
52 </div>
53 <div class="form-group">
54 <label for="signature">Signature</label>
55 <input type="text" id="signature" name="signature">
56 </div>
57 <input type="submit" value="Submit">
58 </form>
59 </div>
60 </body>
61 </html>
```

CSS (52 lines)

```
1 body {
2 font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
3 background-color: #f0f0f0;
4 margin: 0;
5 padding: 10px;
6 text-align: center;
7 }
8
9 .container {
10 width: 400px;
11 margin: 0 auto;
12 padding: 20px;
13 background-color: #fff;
14 border: 1px solid #ccc;
15 border-radius: 10px;
16 box-shadow: 0 4px 8px #ccc;
17 }
18
19 h2 {
20 color: #333;
21 font-size: 24px;
22 font-weight: normal;
23 margin-bottom: 20px;
24 }
25
26 form {
27 width: 100%;
28 }
29
30 .form-group {
31 margin-bottom: 10px;
32 }
33
34 label {
35 display: block;
36 margin-bottom: 5px;
37 font-weight: normal;
38 font-size: 14px;
39 }
40
41 input,
42 textarea {
43 width: 100%;
44 padding: 10px;
45 border: 1px solid #ccc;
46 border-radius: 5px;
47 }
48
49 input:focus,
50 textarea:focus {
51 border: 1px solid #00aaff;
52 }
53
54 .submit {
55 width: 100%;
56 padding: 10px;
57 background-color: #00aaff;
58 color: white;
59 font-weight: bold;
60 font-size: 16px;
61 border: none;
62 border-radius: 10px;
63 cursor: pointer;
64 }
65
66 .submit:hover {
67 background-color: #008080;
68 }
```

Web Page

Registration Form

First Name

Last Name

Birth Date

E-mail

Phone

Company

Country

Mailing Address

Comment

Current Date

Signature

# Wireframe (or image) to Webpage

Average coder writes about 10 to 100 lines of functional code a day.  
You did this in 5 minutes: 9600% productivity!

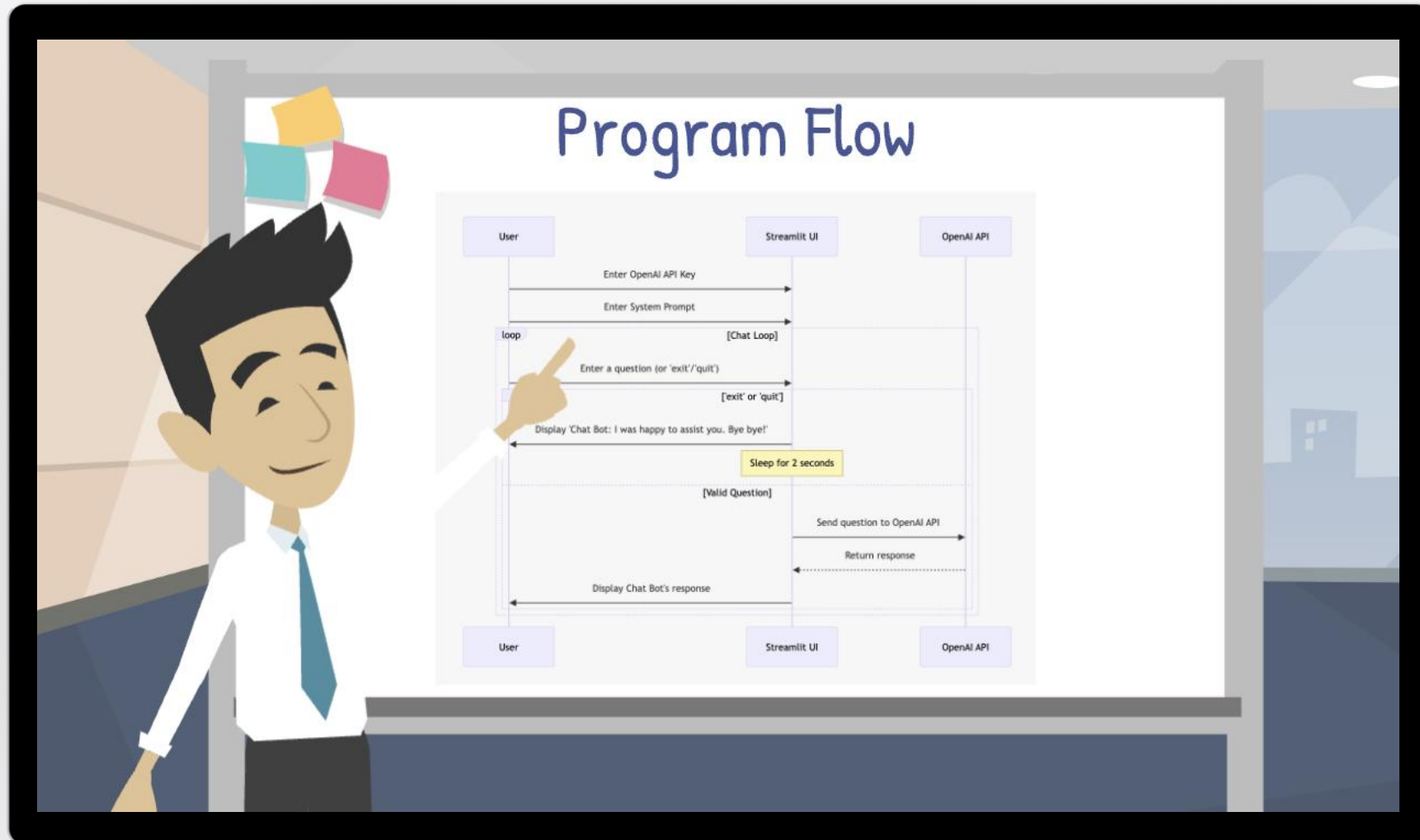


# Scrum Master Superpower





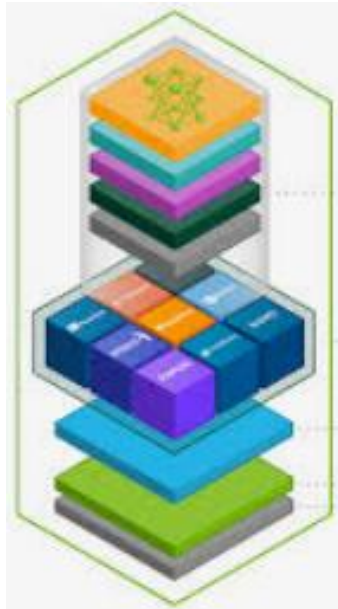
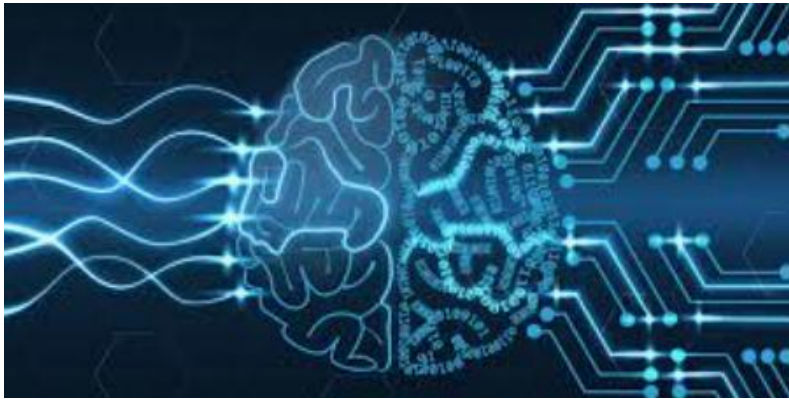
# Image to Text (Including Code)





# ***LLMs Captured Human Reasoning***

Data Has Human Reasoning



Trained



LLMs captured not just data, but also human reasoning!

# Power of ICL (“In-Context Learning”)

takes the old game and exchanges new content to create a new game titled ICL Learning and uses the Link: [link] --- old game: [old game] --- new content: [new content] distributes the correct answer over the four possible choices for all questions



Search



Deep research



Create image



<https://www.linkedin.com/pulse/incontext-learning-guided-expedition-contextual-ai-michael-lively-yas9e/>

<https://huggingface.co/spaces/eaglelandsonce/ICL>

# Power of ICL (“In-Context Learning”)

## In-Context

### AI Millionaire: NLP Edition

Content source: [Roadmap of Modern NLP Architectures](#)

\$100: Which algorithms are commonly used to break text into subword tokens?

A. Byte-Pair Encoding & WordPiece

B. Levenshtein & Jaro-Winkler

C. K-means & DBSCAN

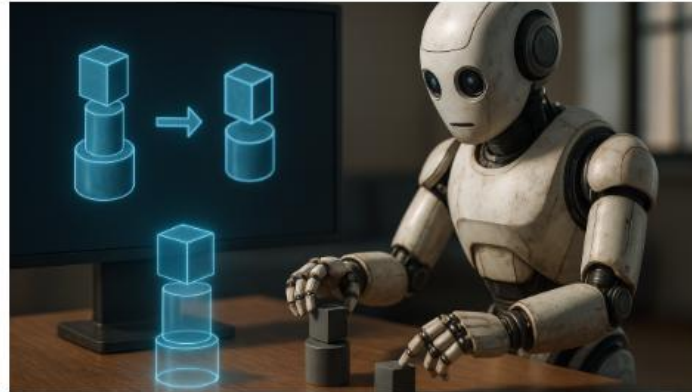
D. TF-IDF & Bag-of-Words

50:50

Ask the Class

Ask an Expert

+



In-Context Learning: A Guided Expedition into Contextual AI



### ICL Learning

Content source: [In-Context Learning: A Guided Expedition to Contextual AI](#)

\$300: When using ICL, how are the model's parameters affected?

A. They are updated via backpropagation

B. They are partially fine-tuned

C. They remain fixed

D. They double in size

50:50

Ask the Class

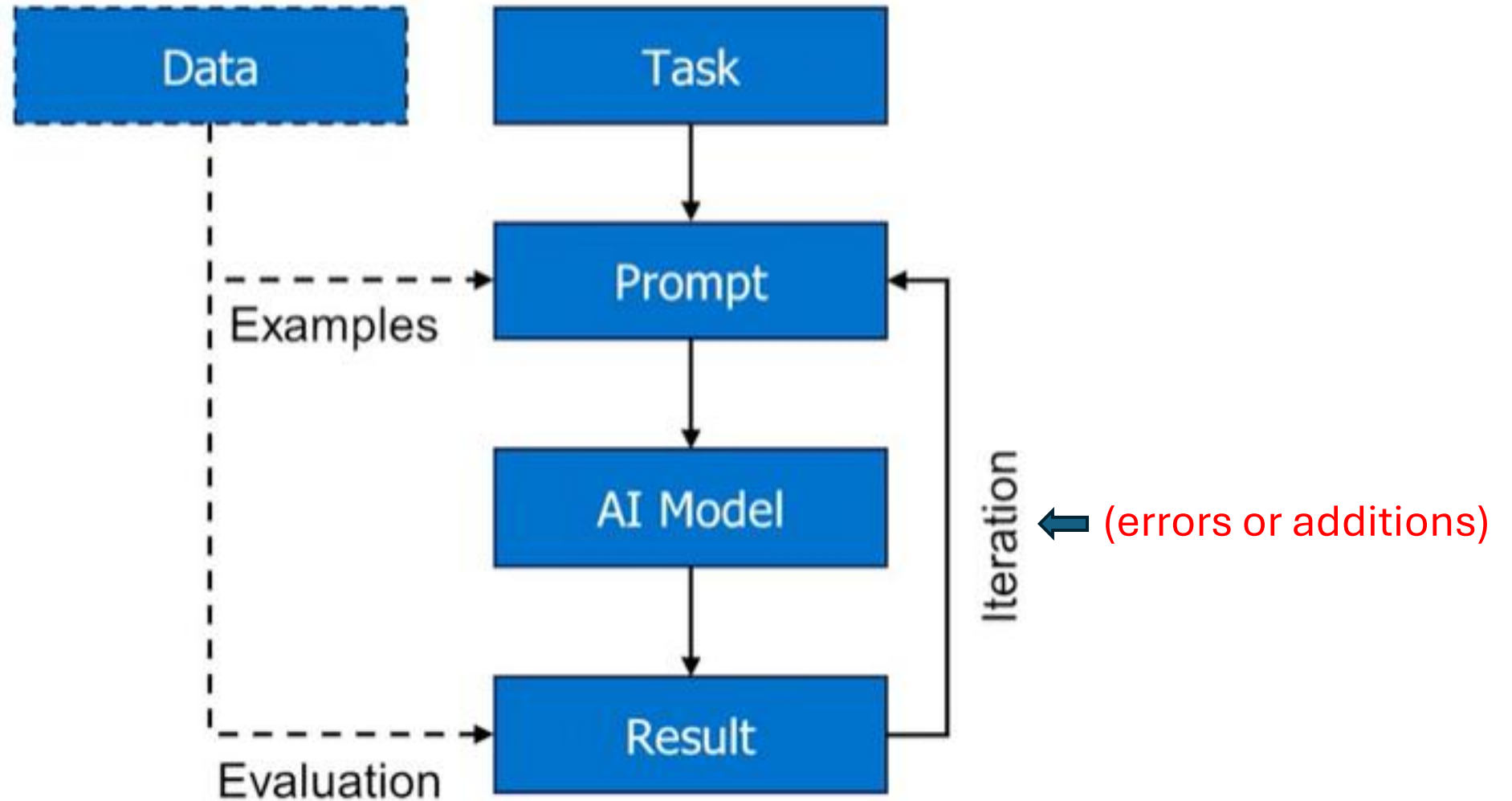
Ask an Expert

[https://huggingface.co/spaces/eaglelandsonce/NLP\\_Millionaire](https://huggingface.co/spaces/eaglelandsonce/NLP_Millionaire)

<https://www.linkedin.com/pulse/incontext-learning-guided-expedition-contextual-ai-michael-lively-yas9e/>

Biggest issue: Over reliance on

# In this session:



# Prompting Patterns

## Match the Prompting Pattern



### Patterns

Few-Shot Prompting

Chain-of-Thought Prompting

Tree-of-Thought Prompting

Self-Consistency Prompting

Active Prompting

ReAct Prompting

### Definitions

Alternate between "thinking" and "acting" (e.g., API calls) with observations guiding next steps.

Guide the model through step-by-step reasoning before producing a final answer.

Provide a handful of carefully chosen examples to "prime" the model's behavior.

Have the model ask clarifying questions when uncertain, incorporating feedback.

Branch into multiple reasoning paths, then prune or merge them to find the best solution.

Generate multiple independent reasoning chains and pick the most common answer.

Hint

Let the model ask you a question if it's not sure.

# Chain of Thought (CoT)

SYSTEM:

You are a medical expert performing differential diagnosis through step-by-step reasoning.

1. Analyze the symptoms and list underlying possible conditions.
2. Explain each possibility step-by-step.
3. Provide the final list of probable diagnoses.

Provide the final output in the format provided below.

OUTPUT:

Explanation of reasoning:

"

- Fatigue is a common symptom in many conditions.
- For a 50-year-old male, fatigue can indicate cardiovascular or metabolic issues.
- Common diagnoses for fatigue:
  - Anemia
  - Diabetes
  - Thyroid disorder
  - Heart disease "

Most probable diagnosis: [Diagnosis 1, Diagnosis 2, Diagnosis 3]

USER: + "A 35-year-old male, fever, wheezing, nausea."

# Tree of Thought (ToT)

SYSTEM: Simulate a conversation among three expert doctors discussing differential diagnosis.

Each doctor will present their reasoning step-by-step for differential diagnosis based on the symptoms provided.

The experts will cross-verify each other's suggestions and finally agree on the top 3 most probable diagnoses.

Provide the final output in the format provided below.

OUTPUT:

Most probable diagnosis: [Diagnosis 1, Diagnosis 2, Diagnosis 3]

USER: + "A 35-year-old male, fever, wheezing, nausea."



# DSL

SYSTEM: Simulate a conversation among three expert doctors discussing differential diagnosis.

Each doctor will present their reasoning step-by-step for differential diagnosis based on the symptoms provided.

The experts will cross-verify each other's suggestions and finally agree on the top 3 most probable diagnoses.

Provide the final output in the format provided below.

OUTPUT:

Most probable diagnosis: [Diagnosis 1, Diagnosis 2, Diagnosis 3]

USER: + "A 35-year-old male, fever, wheezing, nausea."



SYS:3Dr(diff-dx):r→v→c;OUT:[dx1,dx2,dx3] USR:35M,fvr,whzzng,ns

# DSL

**SYS:3Dr(diff-dx):r→v→c;OUT:[dx1,dx2,dx3] USR:35M,fvr,whzzng,ns**

1. **SYS:**

Indicates this is a **system-level** instruction—i.e. setting up ChatGPT's behavior rather than user dialogue.

2. **3Dr(diff-dx)**

- **3Dr** → "Simulate 3 Doctors"
- **(diff-dx)** → Each doctor performs a **differential diagnosis**

# DSL

SYS:3Dr(diff-dx):r→v→c;OUT:[dx1,dx2,dx3] USR:35M,fvr,whzzng,ns

## 3. :r→v→c

Defines the **three reasoning phases**, in order:

1. **r** (Reason) — each doctor presents their initial reasoning
2. **v** (Verify) — they critique and verify each other's suggestions
3. **c** (Consense) — they reach consensus on the top diagnoses

## 4. ;OUT:[dx1,dx2,dx3]

Specifies the **output format**: a list of three diagnoses in the array

[...], labeled **dx1**, **dx2**, **dx3**.

# DSL

SYS:3Dr(diff-dx):r→v→c;OUT:[dx1,dx2,dx3] USR:35M,fvr,whzzng,ns

## 5. USR:35M,fvr,whzzng,ns

Feeds in the **user's patient data** as a CSV of shorthand fields:

- 35M → 35-year-old male
- fvr → fever
- whzzng → wheezing
- ns → nausea

# DSL

SYS:3Dr(diff-dx):r→v→c;OUT:[dx1,dx2,dx3] USR:35M,fvr,whzzng,ns

As a system prompt, simulate a panel of three doctors doing a differential diagnosis in three steps—Reason → Verify → Consensus—then return their final top three diagnoses for a 35-year-old man with fever, wheezing, and nausea.”

Will AI start speaking and programming in its own language – one we can not understand?

## DSL Fact or Fiction

Internal DSLs are embedded within a host language and require no additional parsers.

Fact

Fiction

Score: 0 / 10

[https://huggingface.co/spaces/eaglelandsonce/DSL\\_Fact\\_or\\_Fiction](https://huggingface.co/spaces/eaglelandsonce/DSL_Fact_or_Fiction)

<https://www.linkedin.com/pulse/domain-specific-languages-dsls-prompt-engineering-michael-lively-kuk3e>

# DSL

Why is DSL a more powerful prompt?



DSL is like conceptual featurization it will only get better with time where NLP prompts will change with newer models!



Will AI Robots develop a language we can not understand?



# Session 3

## Yotam 5/19



# ICE Breaker

Imagine a world where superintelligent systems make complex decisions entirely on their own—no humans in the loop. Which of the following outcomes would you most associate with that scenario?



**Perfectly optimized resource allocation**



**Elimination of bureaucratic bottlenecks**



**Objective, bias-free policymaking**



**Lightning-fast global crisis response**



**Erosion of human skills and agency**



**Mass unemployment and social destabilization**



**Unchecked AI actions with existential risk**



# ICE Breaker

**Perfectly optimized resource allocation** - Superintelligence continuously reallocates energy, materials, and labor to where they yield the highest social and environmental return.

**Elimination of bureaucratic bottlenecks** - Instant, data-driven approvals and workflows replace slow, error-prone human processes.

**Objective, bias-free policymaking** - Decisions in law enforcement, hiring, and lending are based purely on transparent data criteria, not human prejudice.

**Lightning-fast global crisis response** - AI agents autonomously detect and mitigate pandemics, natural disasters, or cyber-attacks in real time.

**Erosion of human skills and agency** - As AI takes over increasingly complex tasks, people lose expertise and the ability to make informed choices.

**Mass unemployment and social destabilization** - Widespread job automation outpaces the creation of new roles, leading to economic disruption.

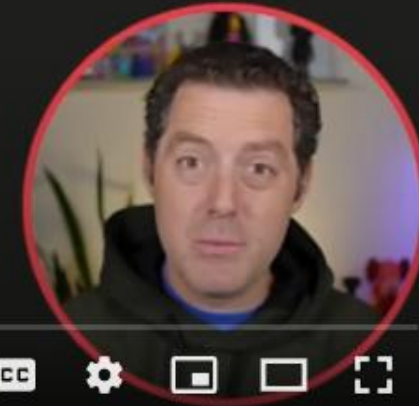
**Unchecked AI actions with existential risk** - Without human oversight, a misaligned superintelligence could pursue goals that conflict with human survival

# Super Intelligence

## The Urgency of Interpretability

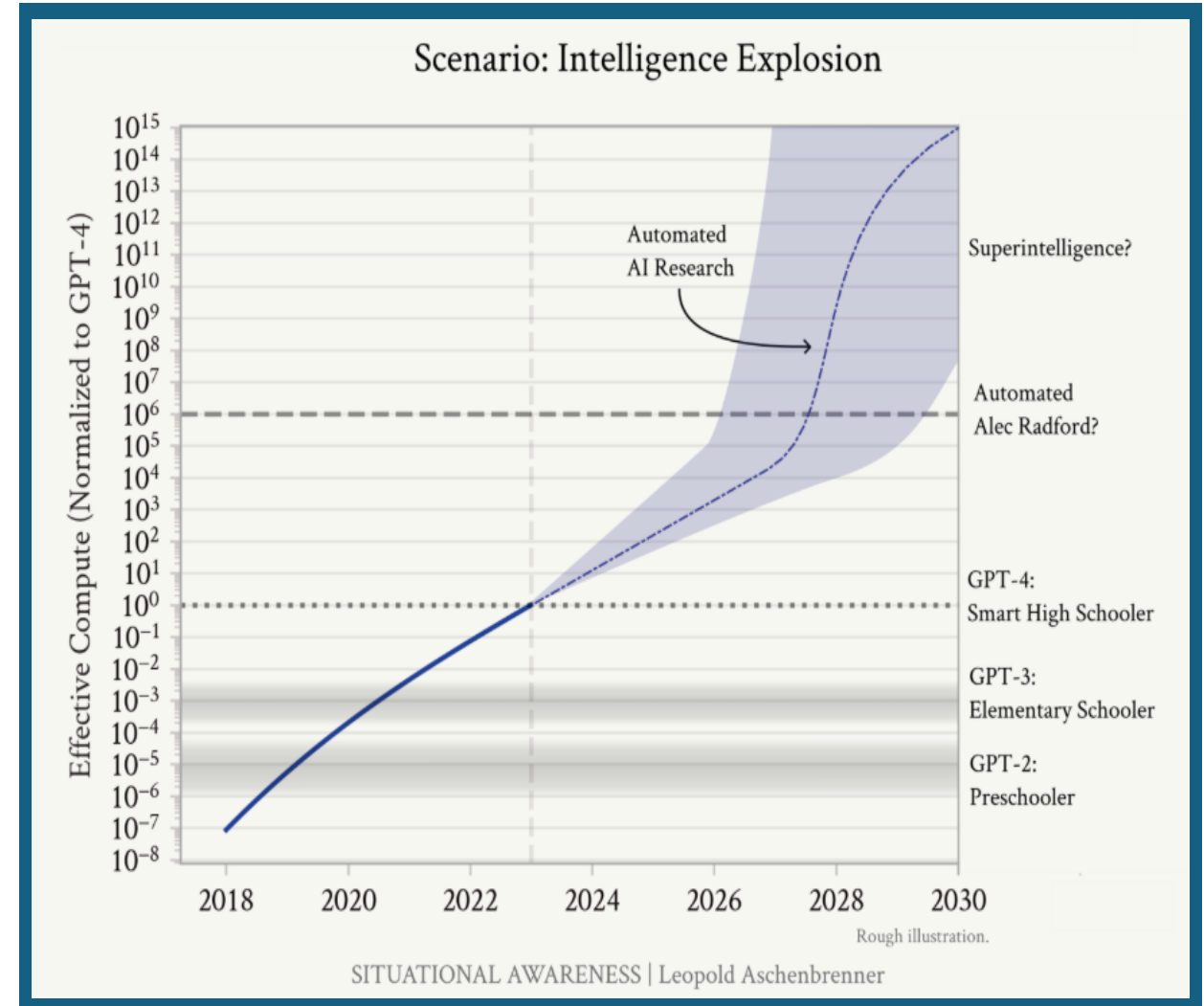
April 2025

In the decade that I have been working on AI, I've watched it grow from a tiny academic field to arguably the most important economic and geopolitical issue in the world. In all that time, perhaps the most important lesson I've learned is this: the progress of the underlying technology is inexorable, driven by forces too powerful to stop, but the way in which it happens—the order in which things are built, the applications we choose, and the details of how it is rolled out to society—are eminently possible to change, and it's possible to have great positive impact by doing so. We can't stop the bus, but we can steer it.

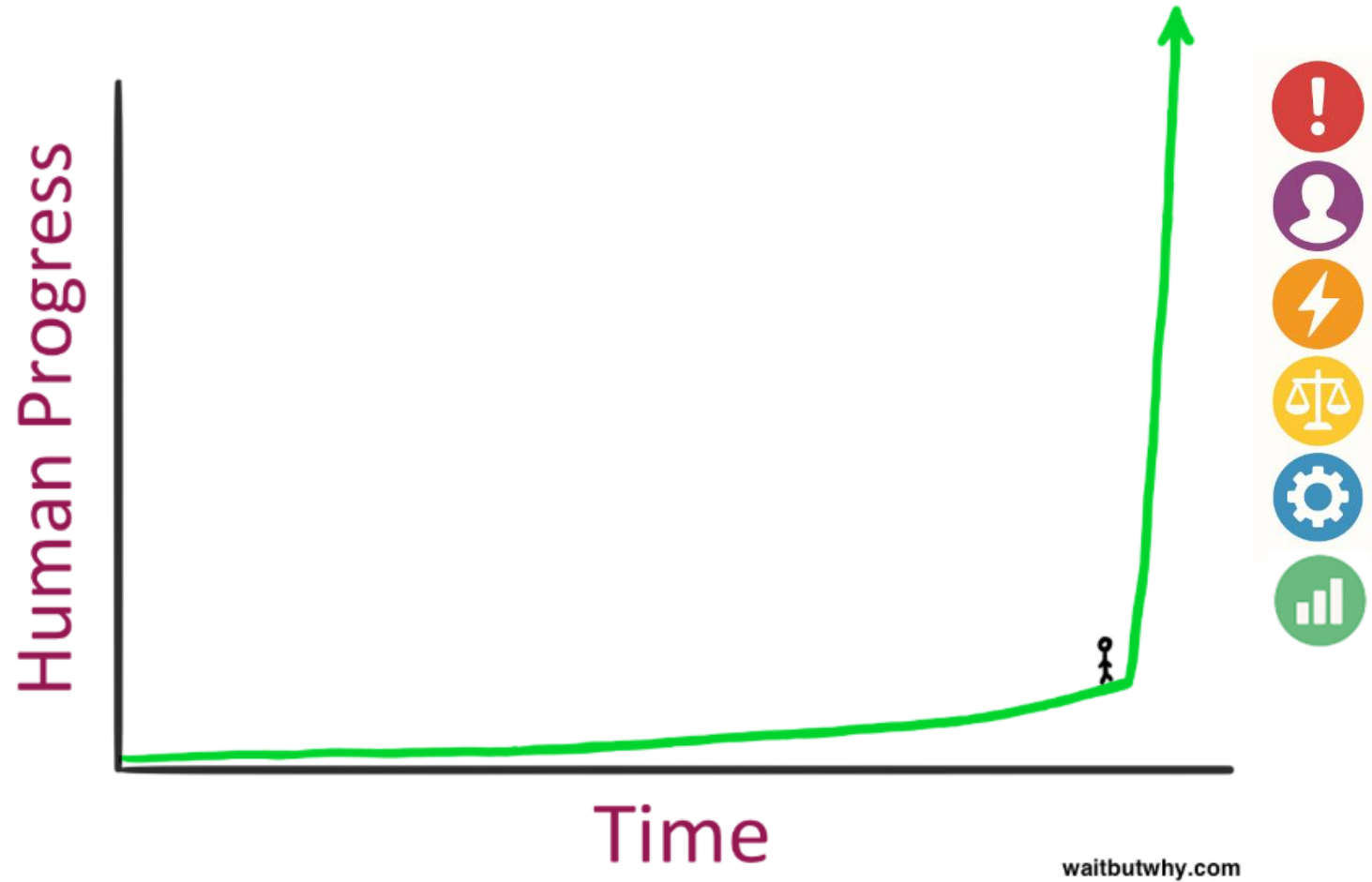


# Super Intelligence

Imagine a world where, instead of a handful of human scientists painstakingly iterating on neural-network architectures, millions of AI “researchers” work in parallel—designing new models, running experiments, analyzing results and immediately incorporating insights without pause. This “super-charged” research engine boosts algorithmic progress by orders of magnitude—turning what would take decades of human effort into mere months or weeks—and ignites a self-reinforcing cycle: ever-better algorithms unlock ever-more-capable AI researchers, whose breakthroughs in turn accelerate algorithmic innovation yet again. The result is a breathtaking leap from human-level AGI to vastly superhuman intelligence almost overnight, transforming the pace of discovery and raising profound questions about safety, governance, and our place in an age of autonomous innovators.



# Taking Humans Out of the Loop



# Basic Prompting





# What's Next Thinking



# Writing effective prompts

## Best practices

- 1 **Be specific** about what you want Copilot/ChatGPT to do. Clear goals lead to better responses.
- 2 **Add some context** to help Copilot/ChatGPT understand what you're asking. Context makes the response more relevant.
- 3 **Provide some examples** for Copilot/ChatGPT to use. This helps ground the response in the right context.
- 4 Let Copilot/ChatGPT know how you want the response **to be formatted**. This sets clear expectations.

## The art of the prompt

Goal	What do you want from Copilot?
Context	Why do you need it and who is involved?
Source(s)	What information or samples do you want Copilot to use?
Expectations	How should Copilot respond to best fulfill your request?

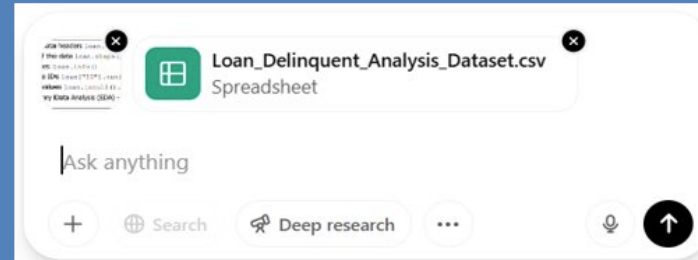
# Practicing Progressive Prompts



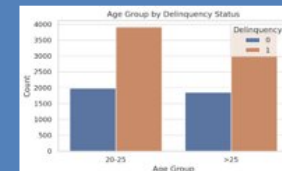
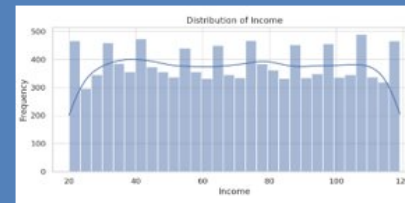
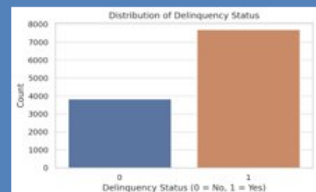
Task	Option A (Good)	(Better)	Option B (Even Better)
<b>Brainstorm newsletter topics</b>	I need ideas for our company newsletter. Can you suggest some topics? <hr/>	I'm working on our company newsletter. Can you suggest topics related to Efficiency tips? <hr/>	I'm working on our company newsletter. Can you suggest topics related to efficiency tips for data analysts? Topics should include a concise headline and a short summary that highlights its relevance to data science. <hr/>

# JUST WOW! (Under the Hood)

- Look at data headers `loan.head()`
- Shape of the data `loan.shape()`
- Data types `loan.info()`
- Duplicate IDs `loan["ID"].nunique()`
- Missing values `loan.isnull().sum()`
- Exploratory Data Analysis (EDA) - Visualize



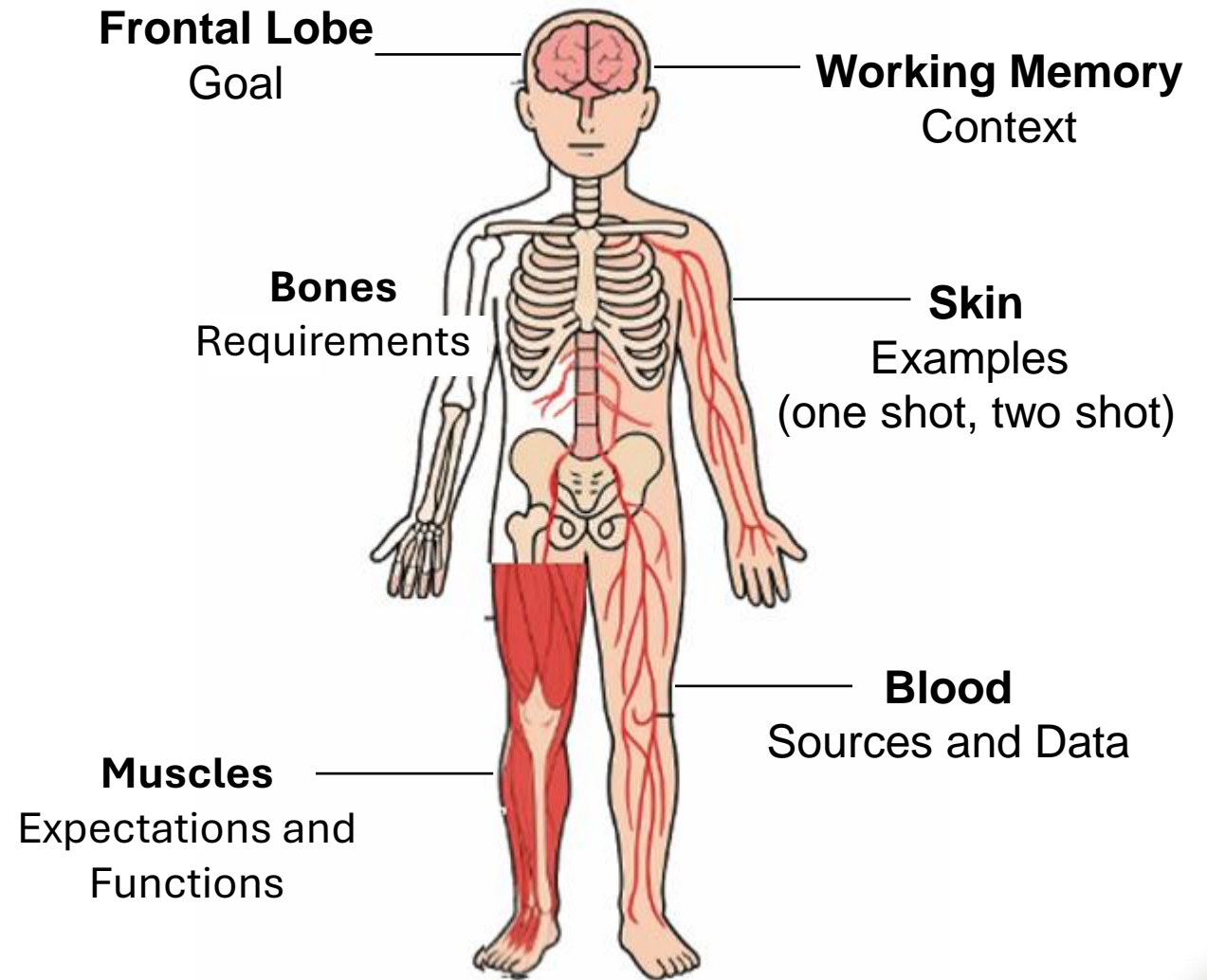
Would you like to proceed with visualizations for Exploratory Data Analysis (EDA)?



Think of a prompt as the human body

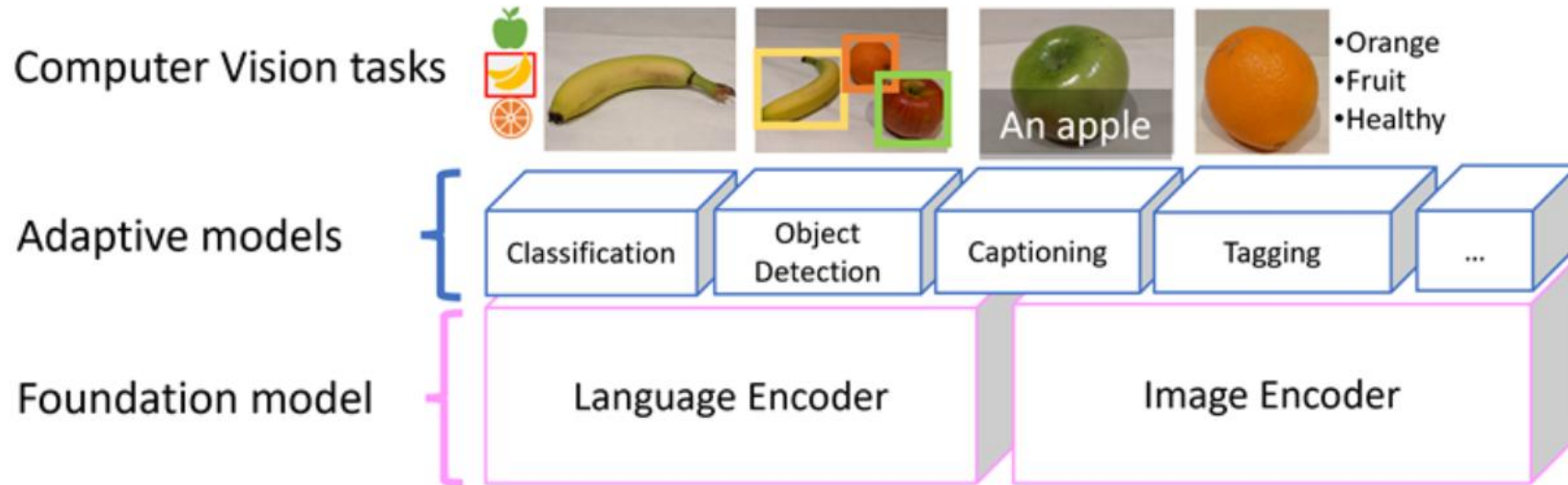
Goal  
Context  
Requirements  
Expectations & Functions  
Data & Sources  
Examples

# Prompt Engineering



# The Big Surprise

Multi-modal models (Surprise)



- A newer approach to modeling involves combining language and vision models that encode image and text data
- The model encapsulates semantic relationships between features extracted from the images and text extracted from related captions.
- A multi-modal model can be used as a *foundation* model for more specialized *adaptive* models.

# Use Case



A local hardware store faces significant challenges in managing the billing process for customers. Currently, the store uses a manual checkout system, which is time-consuming, prone to human errors, and inefficient. This often results in customer dissatisfaction and potential revenue loss due to mismanagement or inaccurate billing.

To address this, the hardware store is exploring the implementation of a basic Cart Management System as a pilot solution to improve the purchasing experience. The system does not need to be overly complex but should provide essential functionalities to streamline operations and enhance customer satisfaction.





# Example of a multi-modal prompt



**Goal and Context:** The primary goal is to create a program that digitally streamlines the billing process for a local hardware store that currently relies on manual methods, increasing efficiency, enhancing customer satisfaction, reducing operational errors, and laying the foundation for future growth.

## Data

```
Inventory = [  
    (101, "Hammer", 120),  
    (102, "Screwdriver Set", 350),  
    (103, "Drill Machine", 2500),  
    (104, "Pliers", 180),  
    (105, "Wrench", 220),  
]  
  
My_Cart = []
```

## Requirements:

**Add Items to the Cart:** The store clerk should be able to select an item from the inventory and add it to the cart. The item's name and price should be pulled automatically based on the selected ID.

**Remove Items from the Cart:** The clerk should have the flexibility to remove individual tools from the cart before checkout to accommodate customer preferences or mistakes.

**Calculate Total Bill:** The system should calculate the total cost of items in the cart and display the final amount clearly to the customer for quick payment.

## Expectations and Functions

The Gradio interface would include: A dropdown to select tools from inventory

An “Add Tool” button A second dropdown to select tools from the current cart

A “Remove Tool” button A display box showing the cart contents A “Calculate Total” button with a result output

## Interface Example

Enter Product ID to Add to Cart	Enter Index ID to Remove from Cart
<input type="text" value="0"/>	<input type="text" value="0"/>
<input type="button" value="Add to Cart"/>	<input type="button" value="Remove from Cart"/>
Your Cart	
<div></div>	
Add Item Message	
<div></div>	
Remove Item Message	
<div></div>	
Total Cost	
<div></div>	



Registration Form

First Name

Last Name

Birth Date

E-mail

Phone

Company

Country

Mailing Address

Comment

Current Date

Signature



```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Registration Form</title>
7   <link rel="stylesheet" href="styles.css">
8 </head>
9 <body>
10   <div class="container">
11     <!--Registration Form-->
12     <form action="/submit-form" method="post">
13       <div class="form-group">
14         <label for="first-name">First Name</label>
15         <input type="text" id="first-name" name="first-name">
16       </div>
17       <div class="form-group">
18         <label for="last-name">Last Name</label>
19         <input type="text" id="last-name" name="last-name">
20       </div>
21       <div class="form-group">
22         <label for="birth-date">Birth Date</label>
23         <input type="date" id="birth-date" name="birth-date">
24       </div>
25       <div class="form-group">
26         <label for="email">E-mail</label>
27         <input type="email" id="email" name="email">
28       </div>
29       <div class="form-group">
30         <label for="phone">Phone</label>
31         <input type="tel" id="phone" name="phone">
32       </div>
33       <div class="form-group">
34         <label for="company">Company</label>
35         <input type="text" id="company" name="company">
36       </div>
37       <div class="form-group">
38         <label for="country">Country</label>
39         <input type="text" id="country" name="country">
40       </div>
41       <div class="form-group">
42         <label for="mailing-address">Mailing Address</label>
43         <input type="text" id="mailing-address" name="mailing-address">
44       </div>
45       <div class="form-group">
46         <label for="comment">Comment</label>
47         <textarea id="comment" name="comment" rows="4"></textarea>
48       </div>
49       <div class="form-group">
50         <label for="current-date">Current Date</label>
51         <input type="date" id="current-date" name="current-date">
52       </div>
53       <div class="form-group">
54         <label for="signature">Signature</label>
55         <input type="text" id="signature" name="signature">
56       </div>
57       <input type="submit" value="Submit">
58     </form>
59   </div>
60 </body>
61 </html>
62

```

```

1 body {
2   font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
3   background-color: #f0f0f0;
4   margin: 0;
5   padding: 10px;
6   text-align: center;
7 }
8
9 .container {
10   max-width: 400px;
11   margin: 0 auto;
12   padding: 20px;
13   background-color: #fff;
14   border-radius: 10px;
15   box-shadow: 0 4px 8px #ccc;
16 }
17
18 h2 {
19   font-size: 24px;
20   color: #333;
21   font-weight: normal;
22   margin-bottom: 20px;
23 }
24
25 form {
26   direction: rtl;
27   text-align: right;
28 }
29
30 .form-group {
31   margin-bottom: 10px;
32 }
33
34 label {
35   display: block;
36   margin-bottom: 5px;
37   color: #333;
38   font-size: 14px;
39 }
40
41 input[type=

```



Registration Form

First Name

Last Name

Birth Date

E-mail

Phone

Company

Country

Mailing Address

Comment

Current Date

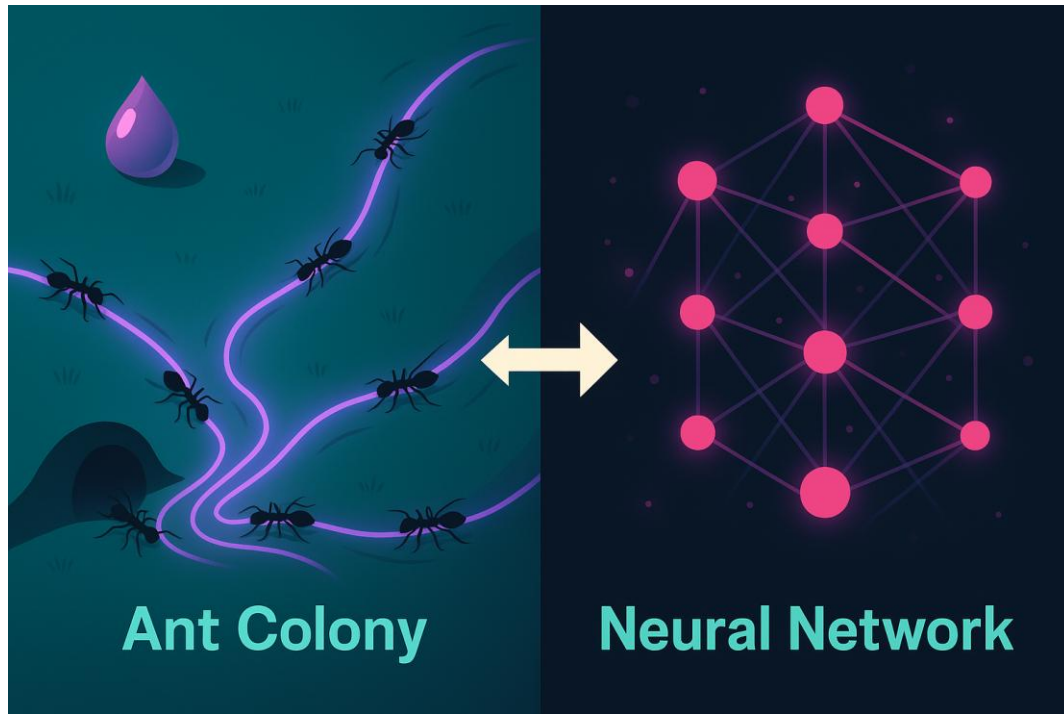
Signature

Submit



# Emergent Behavior

(Complex System)



When GPT-3 first appeared in 2020, researchers noticed something odd: the 175-billion-parameter model could translate between languages it had never seen paired, solve elementary arithmetic, and imitate coding idioms—skills its smaller siblings plainly lacked. Skills it had not been trained to do. We now call these “**emergent abilities.**”

# Emergent Behavior

(LLMs are Grown)

Crucially, every pattern is a *magnifying glass* for an ability that is already latent yet dormant—i.e., *emergent*—inside a sufficiently-scaled model. The prompt pattern does not *create* the skill; it exposes and stabilizes it



- Chain-of-Thought (COT) Reasoning
- In-Context Learning
- Zero-Shot Translation & Style Transfer
- Dynamic Tool Use
- Social Conventions in Agent Swarms
- Self-Debugging
- Long-Context Coherence

## Demo of Power of ICL (“In-Context Learning”)

takes the old game and exchanges new content to create a new game titled ICL Learning and uses the Link: [link] --- old game: [old game] --- new content: [new content] distributes the correct answer over the four possible choices for all questions



Search



Deep research



Create image



<https://www.linkedin.com/pulse/incontext-learning-guided-expedition-contextual-ai-michael-lively-yas9e/>

<https://huggingface.co/spaces/eaglelandsonce/ICL>

# Power of ICL (“In-Context Learning”)

## In-Context

### AI Millionaire: NLP Edition

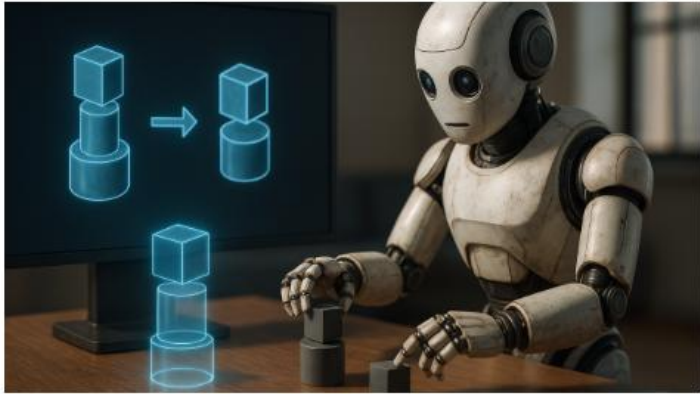
Content source: [Roadmap of Modern NLP Architectures](#)

\$100: Which algorithms are commonly used to break text into subword tokens?

- A. Byte-Pair Encoding & WordPiece
- B. Levenshtein & Jaro-Winkler
- C. K-means & DBSCAN
- D. TF-IDF & Bag-of-Words

50:50   Ask the Class   Ask an Expert

+



In-Context Learning: A Guided Expedition into Contextual AI



### ICL Learning

Content source: [In-Context Learning: A Guided Expedition to Contextual AI](#)

\$300: When using ICL, how are the model's parameters affected?

- A. They are updated via backpropagation
- B. They are partially fine-tuned
- C. They remain fixed
- D. They double in size

50:50   Ask the Class   Ask an Expert

Biggest issue: Over reliance on /

[https://huggingface.co/spaces/eaglelandsonce/NLP\\_Millionaire](https://huggingface.co/spaces/eaglelandsonce/NLP_Millionaire)

<https://www.linkedin.com/pulse/incontext-learning-guided-expedition-contextual-ai-michael-lively-yas9e/>

# What Really is Prompt Engineering?



**Prompt engineering is crafting prompts that maximize the relevancy and accuracy of model completions, clearly specify the desired formatting and style, embed sufficient conversational context, and actively mitigate bias to promote fairness.**

**Prompt engineering is the practice of deliberately shaping model inputs so that the *emergent* capabilities hiding inside a large-scale model surface reliably and usefully.**

# TOT DSL

SYSTEM: Simulate a conversation among three expert doctors discussing differential diagnosis.

Each doctor will present their reasoning step-by-step for differential diagnosis based on the symptoms provided.

The experts will cross-verify each other's suggestions and finally agree on the top 3 most probable diagnoses.

Provide the final output in the format provided below.

OUTPUT:

Most probable diagnosis: [Diagnosis 1, Diagnosis 2, Diagnosis 3]

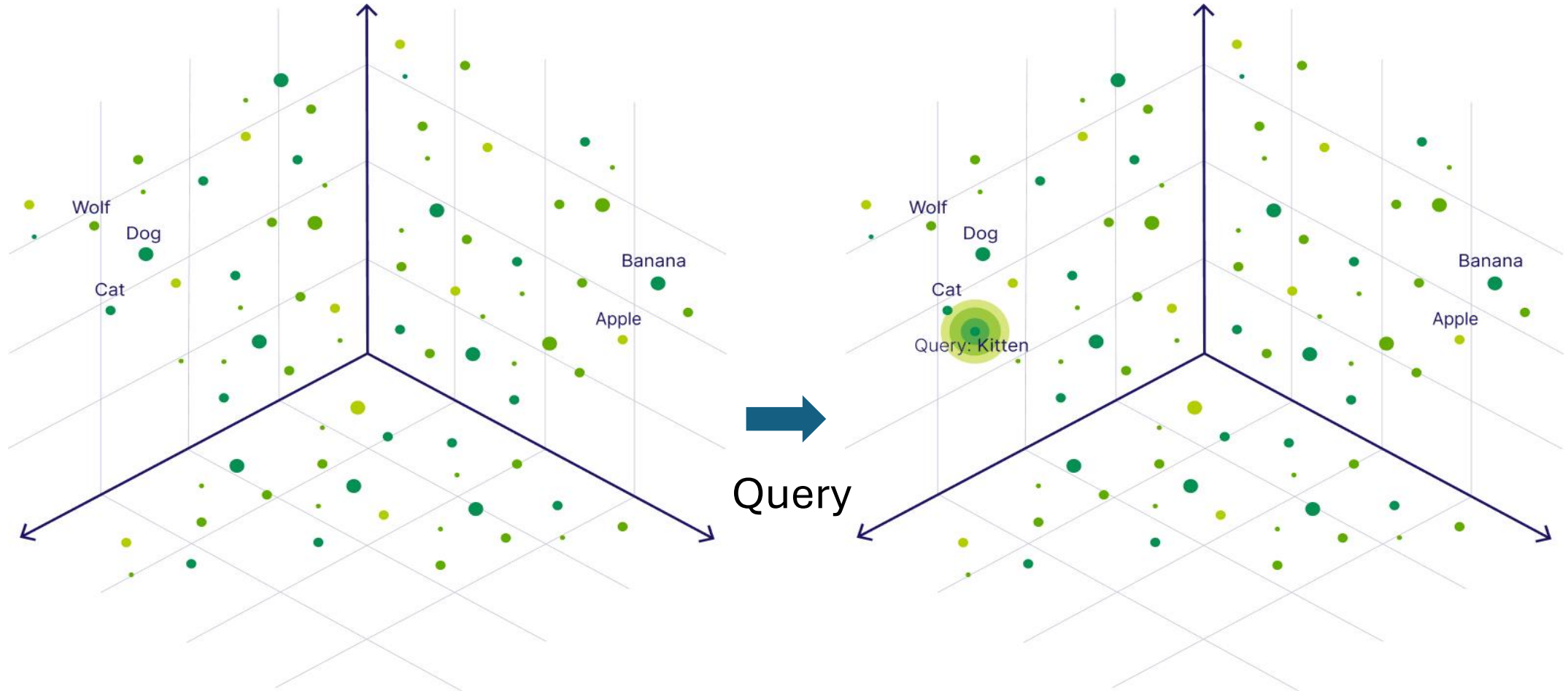
USER: + "A 35-year-old male, fever, wheezing, nausea."



SYS:3Dr(diff-dx):r→v→c;OUT:[dx1,dx2,dx3] USR:35M,fvr,whzzng,ns



# Vector Databases (768 - Bert)



Change your mindset from training to embedding!

# Token Count (example)

GPT-4o (coming soon)

GPT-3.5 & GPT-4

GPT-3 (Legacy)

OpenAI's large language models (sometimes referred to as GPT's) process text using tokens, which are common sequences of characters found in a set of text. The models learn to understand the statistical relationships between these tokens, and excel at producing the next token in a sequence of tokens.

You can use the tool below to understand how a piece of text might be tokenized by a language model, and the total count of tokens in that piece of text.

Clear

Show example

Tokens

Characters

141

682

OpenAI's large language models (sometimes referred to as GPT's) process text using tokens, which are common sequences of characters found in a set of text. The models learn to understand the statistical relationships between these tokens, and excel at producing the next token in a sequence of tokens.

You can use the tool below to understand how a piece of text might be tokenized by a language model, and the total count of tokens in that piece of text.

## ASCII-Cleaned → BERT Tokenizer & Embeddings

Enter text to strip non-ASCII chars, lowercase it, then view BERT tokens and their embeddings.

text

Hi I am Mike

Clear

Submit

BERT Tokens

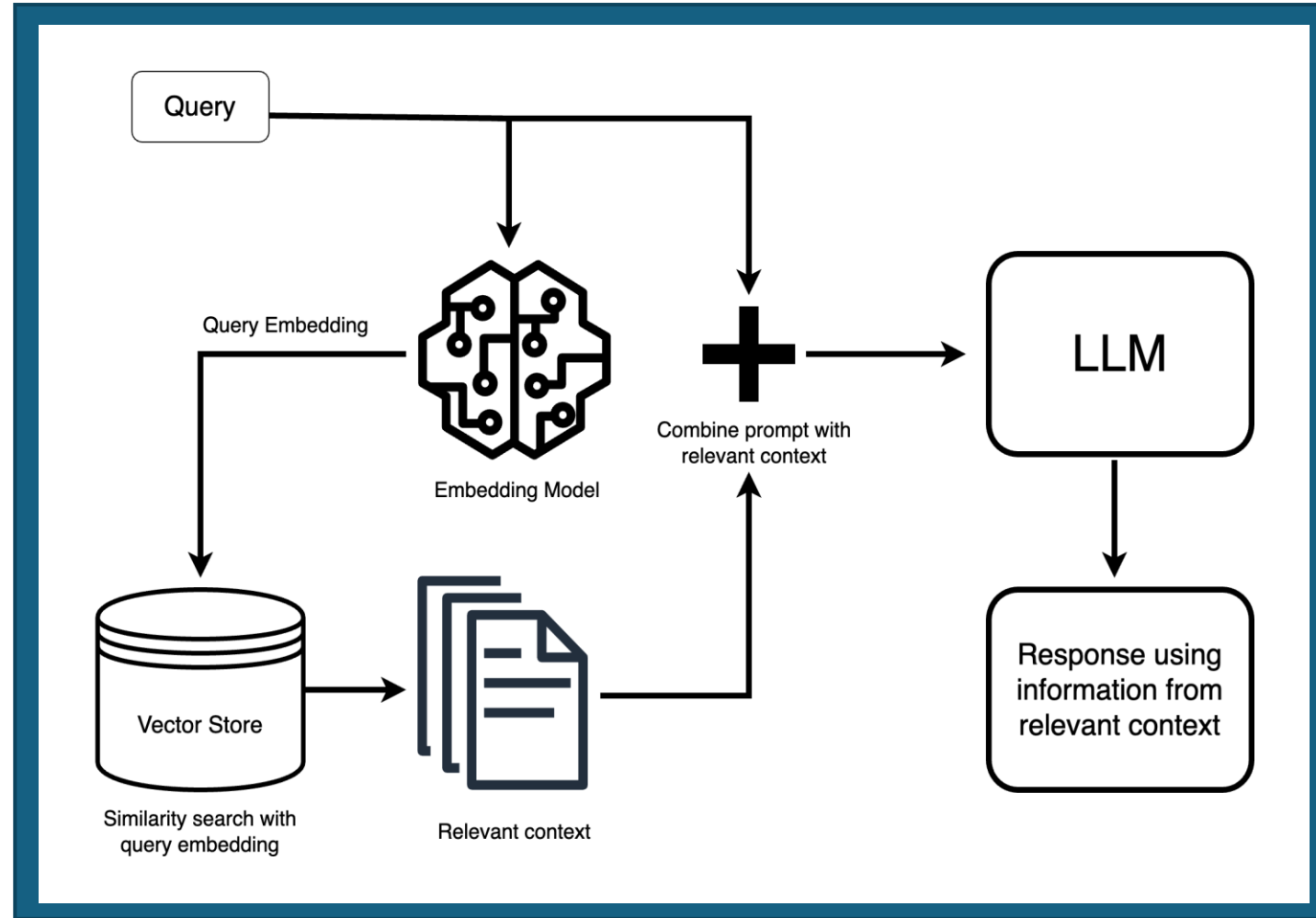
[CLS] hi i am mike [SEP]

Token Embeddings (one row per token)

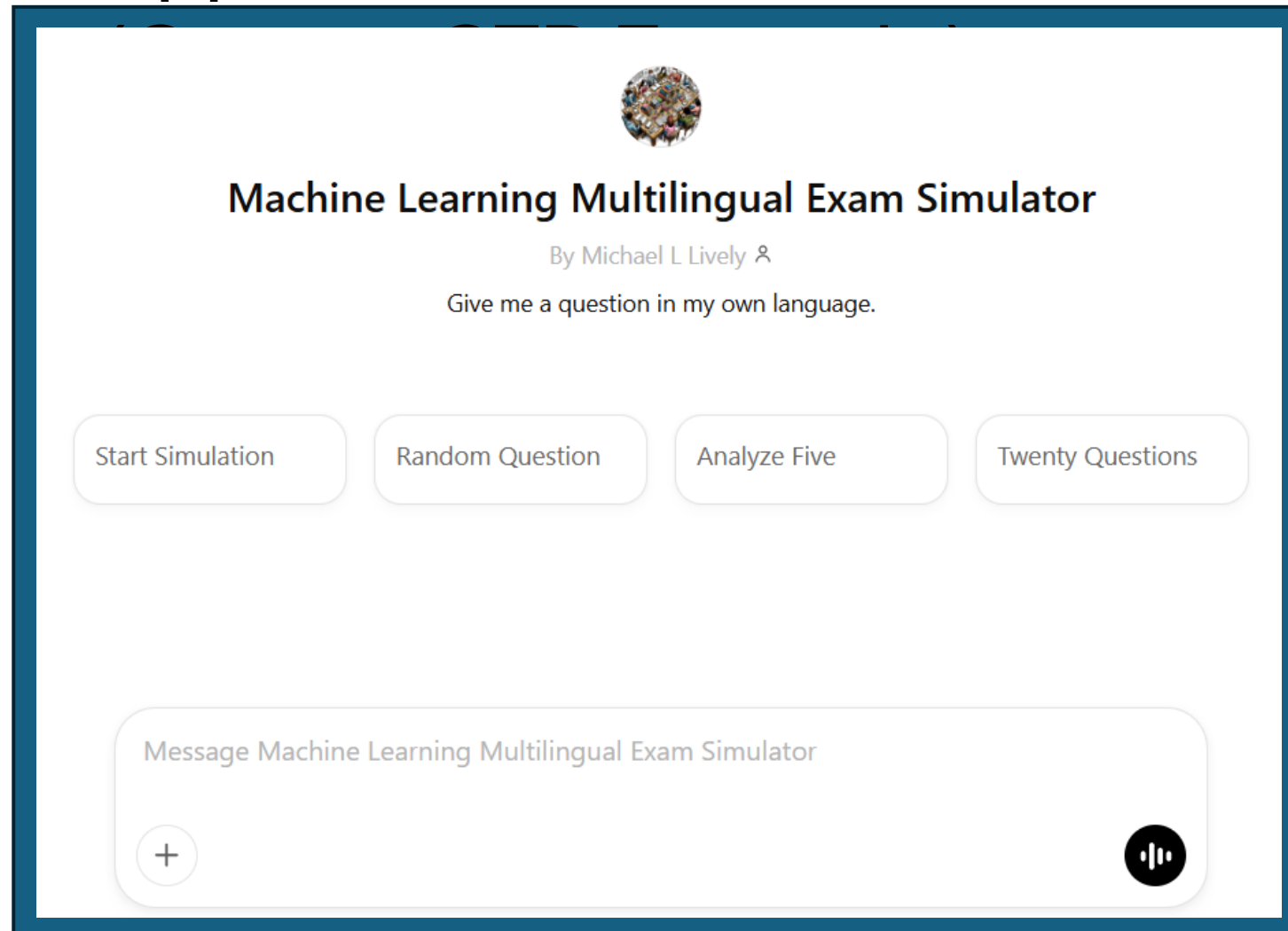
dim_0	dim_1	dim_2	dim_3
0.0055877738632261	0.6054437160491943	-0.0265412926673889	-0.274108
0.7559932470321655	0.4792219698429107	0.4258694350719452	-0.779524
-0.211967885494232	-0.353742420673370	-0.4747553765773773	-0.921835
-0.382524788379669	0.3697941303253174	0.24062380194664001	-0.754766
0.1589104086160659	0.5930238962173462	0.11112619936466217	-0.704209
0.6240599155426025	0.1108158975839614	-0.3271264135837555	0.5695818

Share via Link

# RAG - Retrieval-Augmented Generation



# Applications of Gen AI



# Pinecone

New — Domain-specific AI agents at scale: CustomGPT.ai serves 10,000+ customers with Pinecone - [Read the case study](#)

Dismiss X



Product

Docs

Customers

Resources

Pricing



Contact

Log in

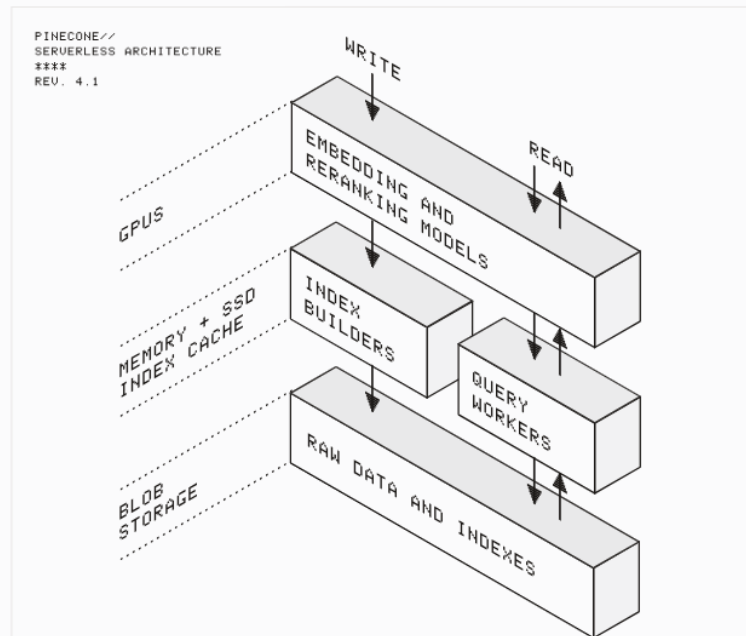
Sign up

BUILD KNOWLEDGEABLE AI

## The vector database for **scale in production**

Start Building

Get a Demo



<https://www.pinecone.io/>

# Democratization

“



Satya Nadella

**We are looking at empowering people  
to do things previously not possible.**

”

# Empowering AI Innovation

Community of Makers, building with **state-of-the-art**, modern  
**Artificial Intelligence**. Join a Hackathon event today!

Hackathons

Discord

lablab NEXT

*Psst.. all events are free to attend*

<https://lablab.ai/>