

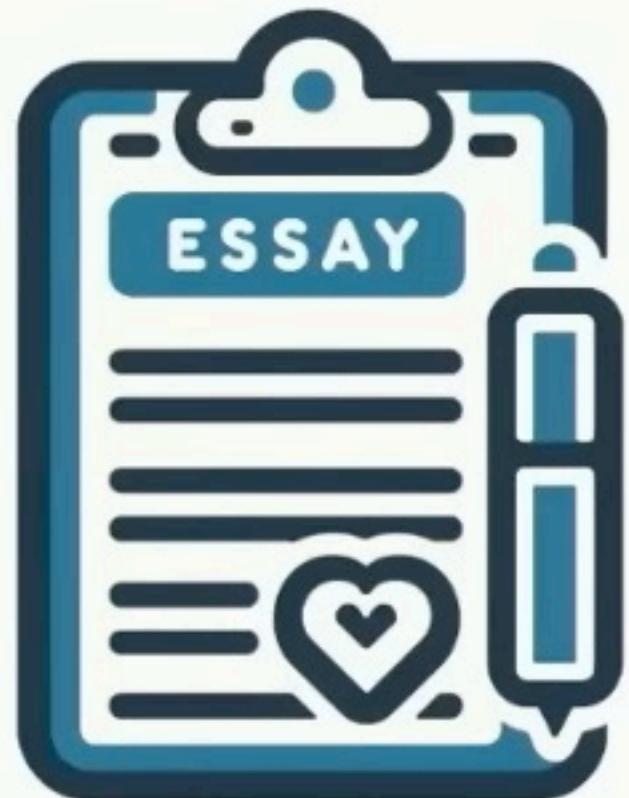
Agentic Reasoning

Andrew Ng

LLM-based agents

Non-agentic workflow (zero-shot):

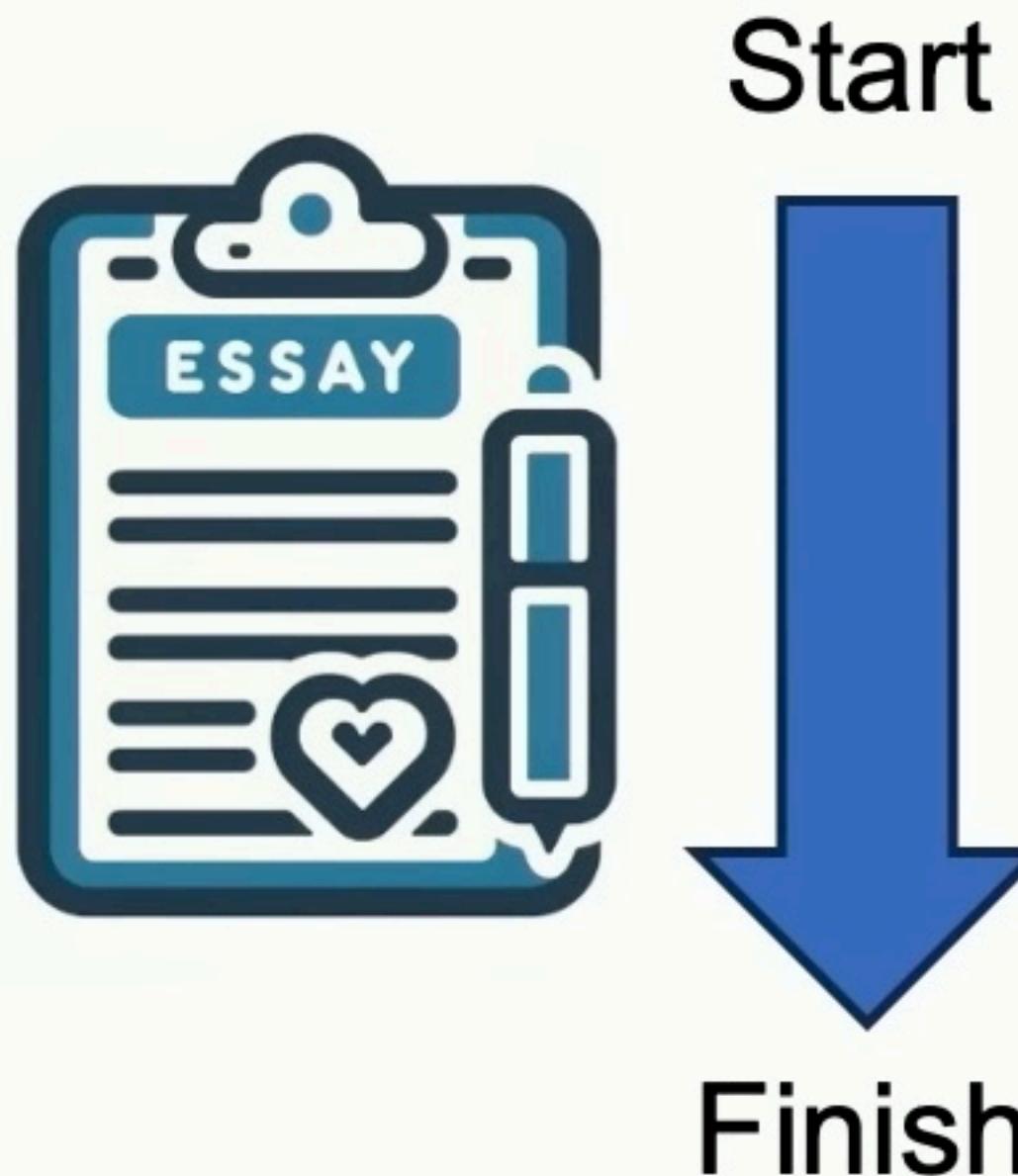
Please type out an essay on topic X from start to finish in one go, without using backspace.



LLM-based agents

Non-agentic workflow (zero-shot):

Please type out an essay on topic X from start to finish in one go, without using backspace.



Agentic workflow:

Write an essay outline on topic X

Do you need any web research?

Write a first draft.

Consider what parts need revision or more research.

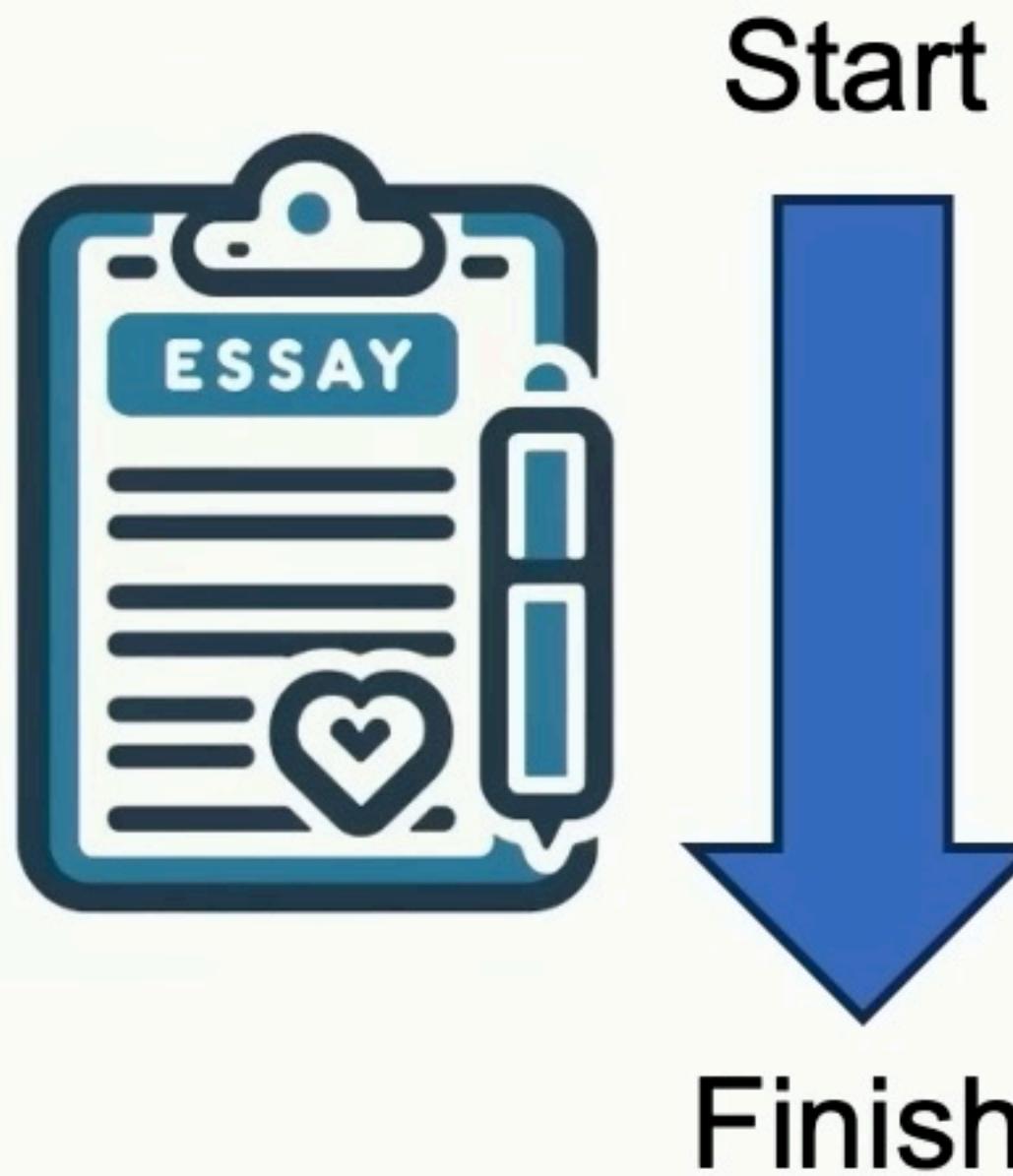
Revise your draft.

....

LLM-based agents

Non-agentic workflow (zero-shot):

Please type out an essay on topic X from start to finish in one go, without using backspace.



Agentic workflow:

Write an essay outline on topic X

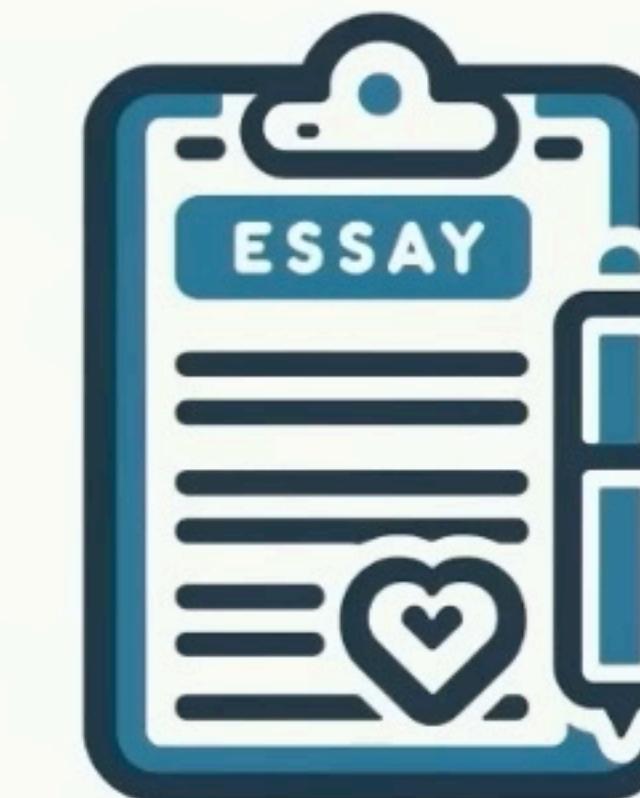
Do you need any web research?

Write a first draft.

Consider what parts need revision or more research.

Revise your draft.

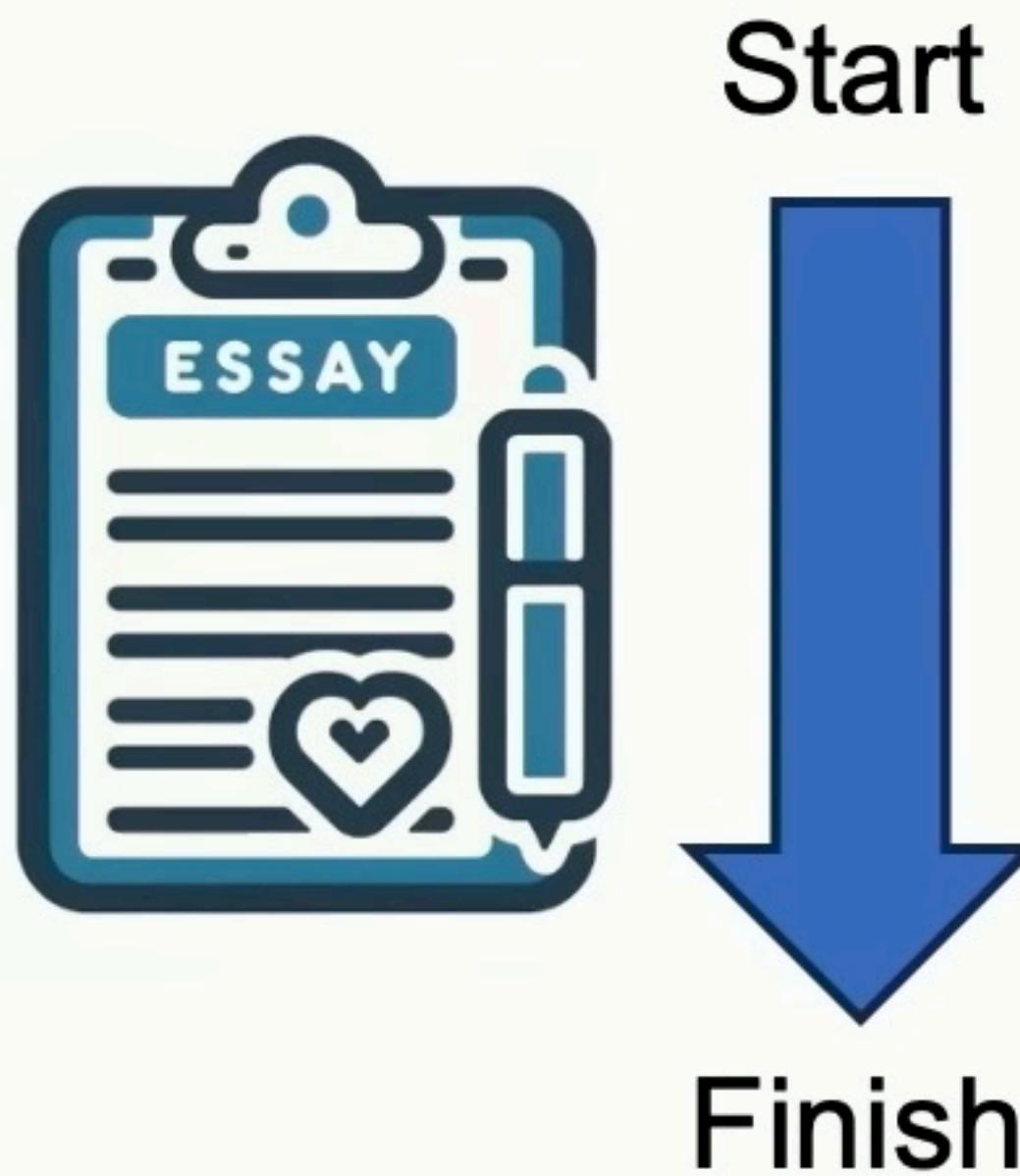
....



LLM-based agents

Non-agentic workflow (zero-shot):

Please type out an essay on topic X from start to finish in one go, without using backspace.



Agentic workflow:

Write an essay outline on topic X

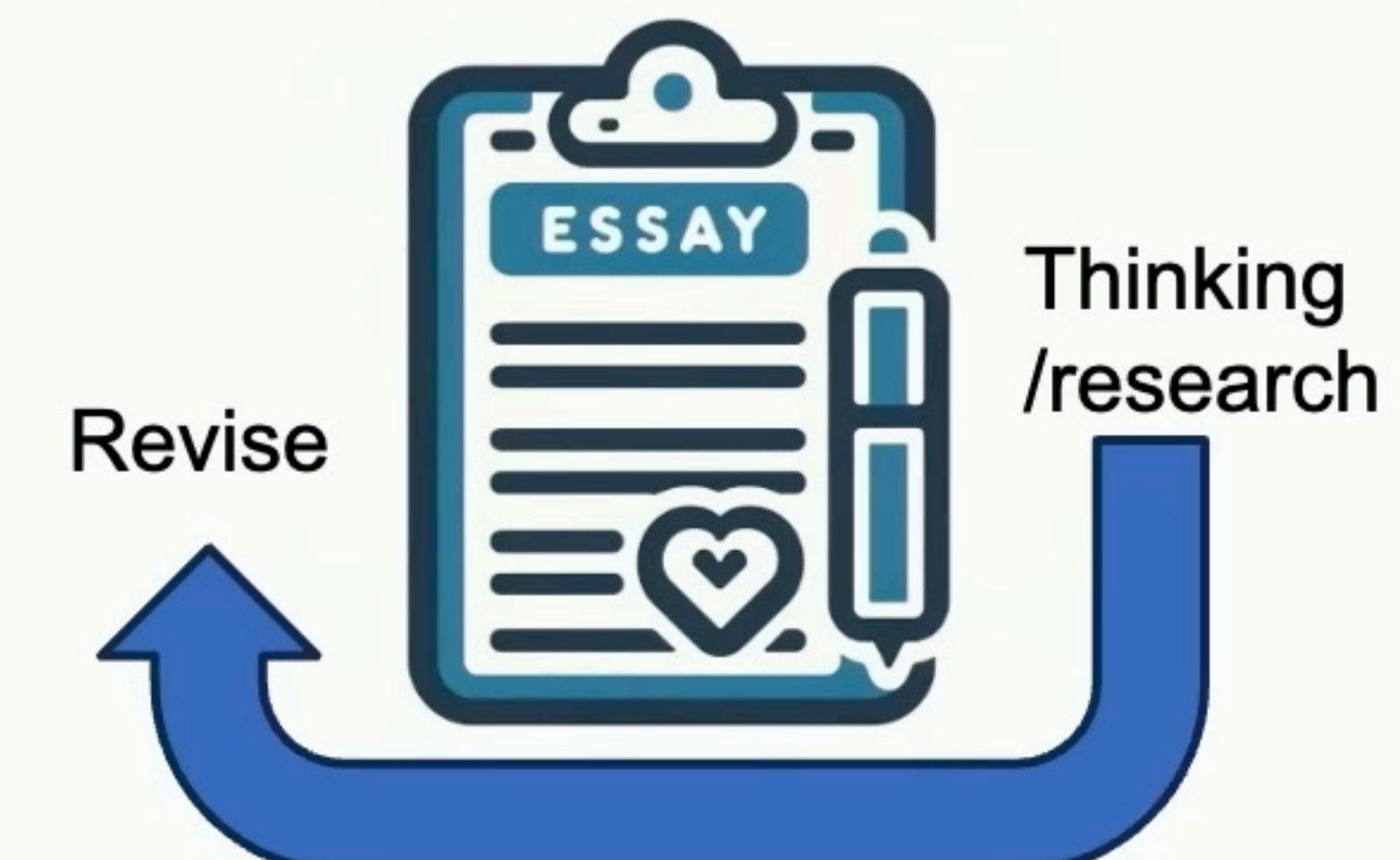
Do you need any web research?

Write a first draft.

Consider what parts need revision or more research.

Revise your draft.

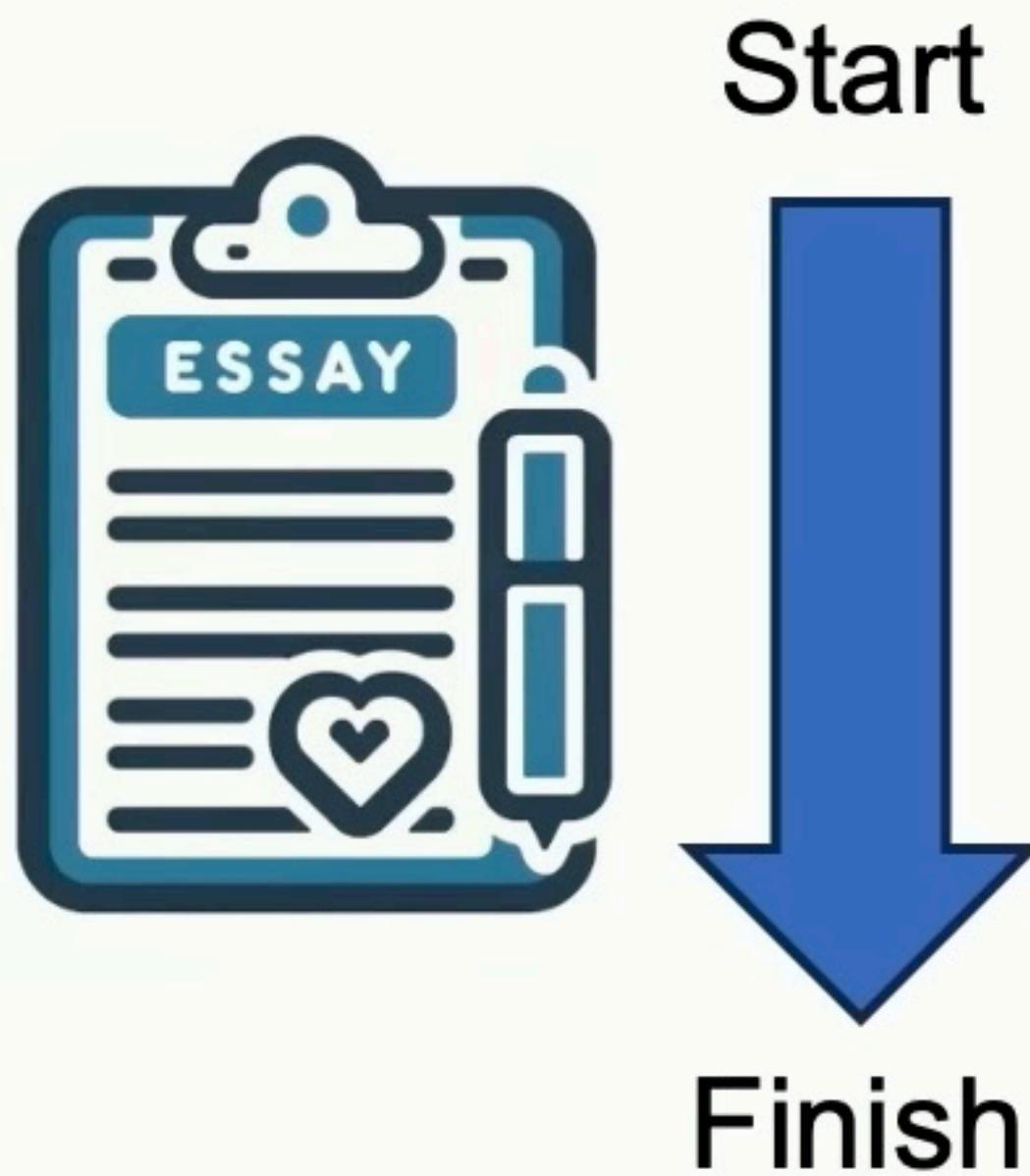
....



LLM-based agents

Non-agentic workflow (zero-shot):

Please type out an essay on topic X from start to finish in one go, without using backspace.



Agentic workflow:

Write an essay outline on topic X

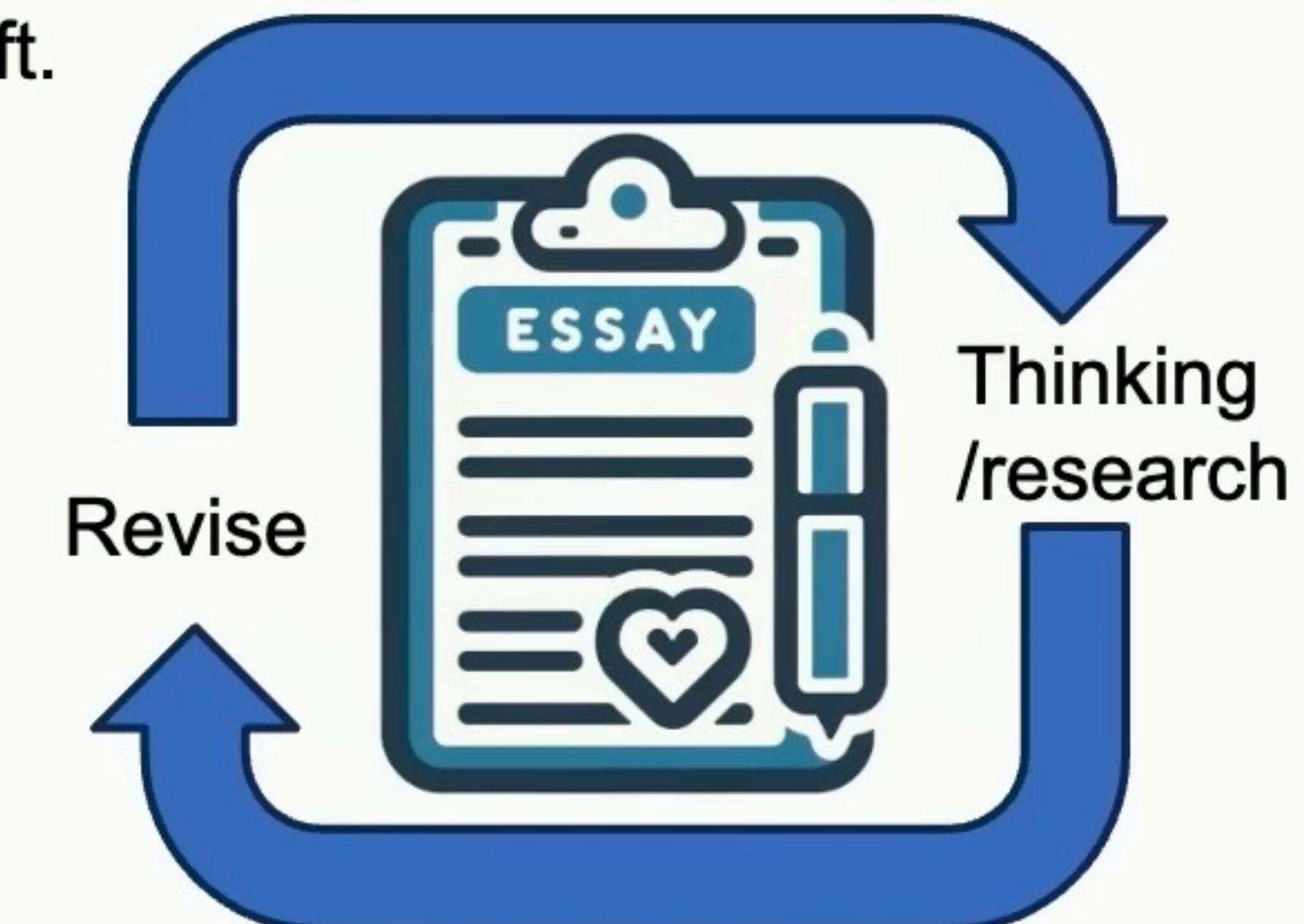
Do you need any web research?

Write a first draft.

Consider what parts need revision or more research.

Revise your draft.

....



Coding benchmark (HumanEval)

Example problem:

```
def solution(lst):
    """Given a non-empty list of integers, return the sum of all of the odd elements
    that are in even positions.
```

Examples

```
solution([5, 8, 7, 1]) ==>12
solution([3, 3, 3, 3, 3]) ==>9
solution([30, 13, 24, 321]) ==>0
"""
```

Coding benchmark (HumanEval)

Example problem:

```
def solution(lst):
    """Given a non-empty list of integers, return the sum of all of the odd elements
    that are in even positions.
```

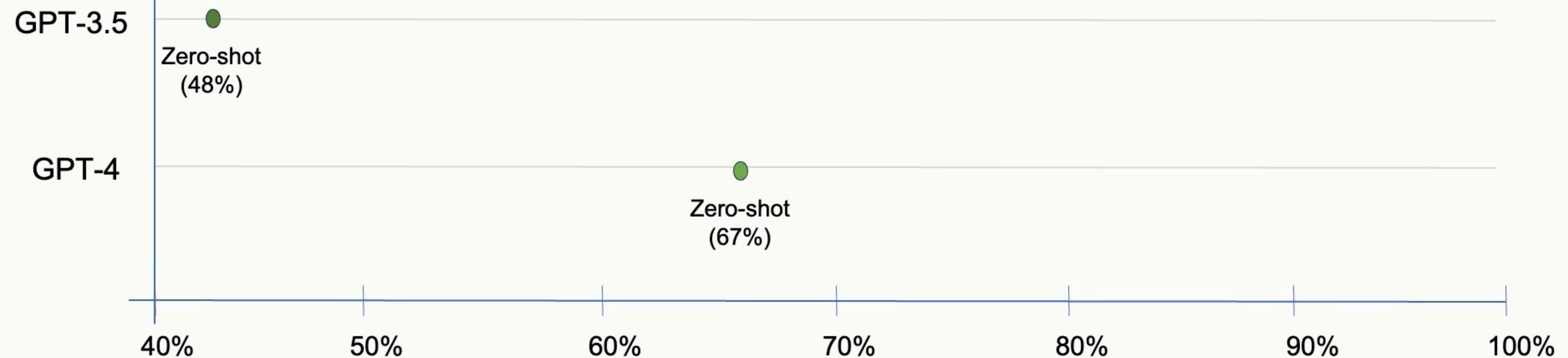
Examples

```
solution([5, 8, 7, 1]) ==>12
solution([3, 3, 3, 3, 3]) ==>9
solution([30, 13, 24, 321]) ==>0
"""
```

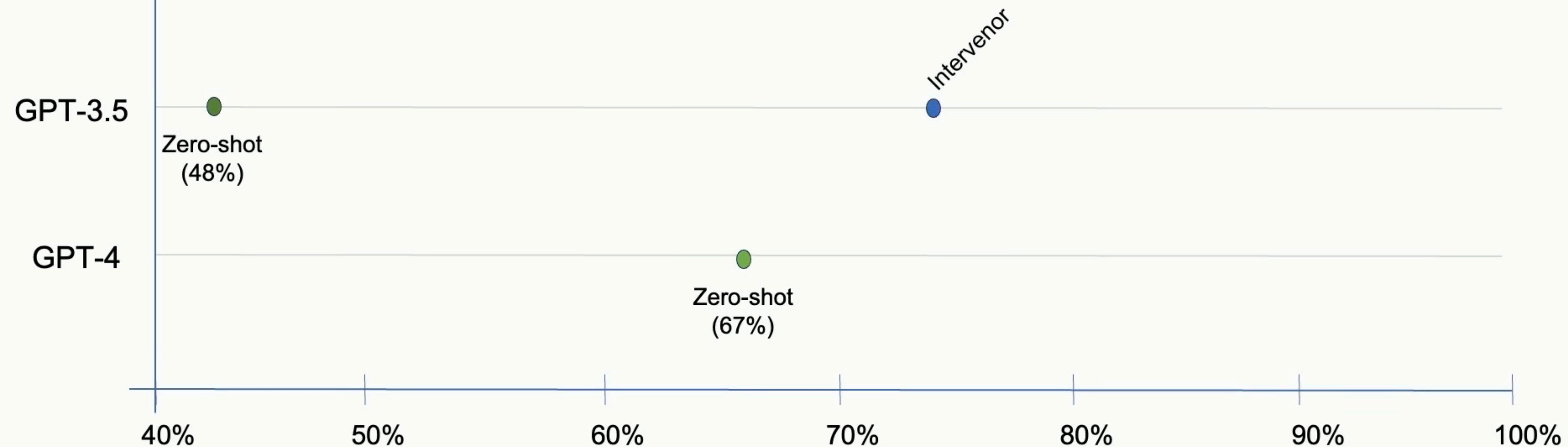
Solution:

```
return sum(lst[i] for i in range(0,len(lst)) if i % 2 == 0 and lst[i] % 2 == 1)
```

Coding benchmark (HumanEval)



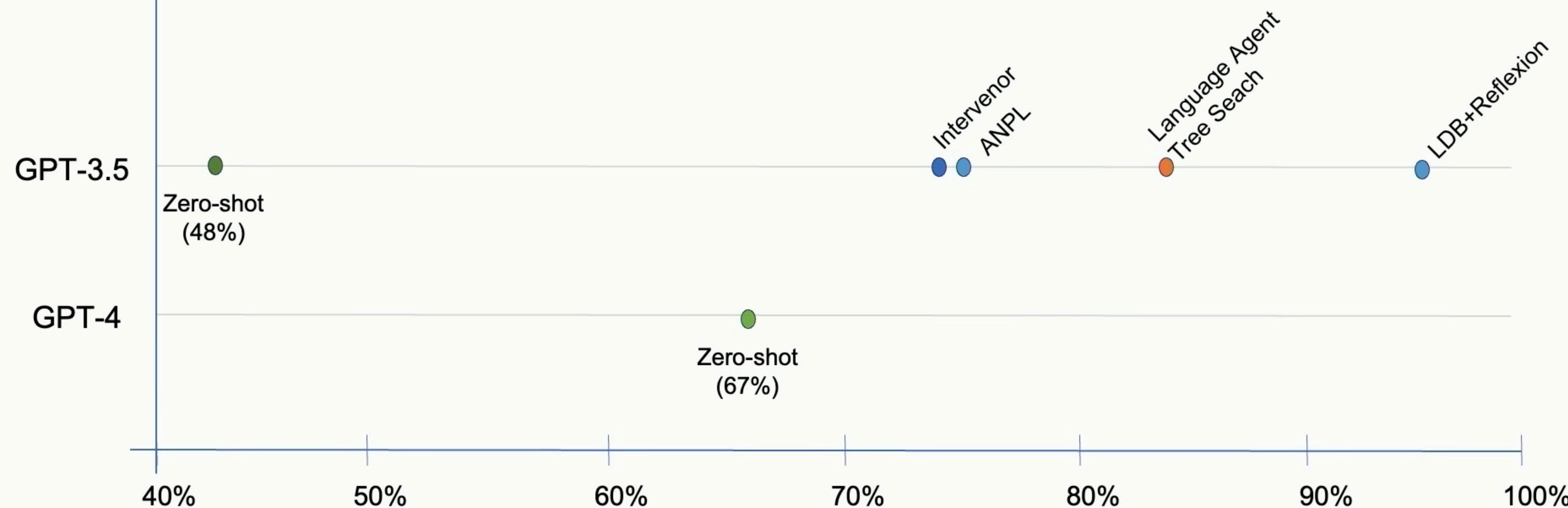
Coding benchmark (HumanEval)



[Thanks to Joaquin Dominguez and John Santerre for help with analysis.]

Andrew Ng

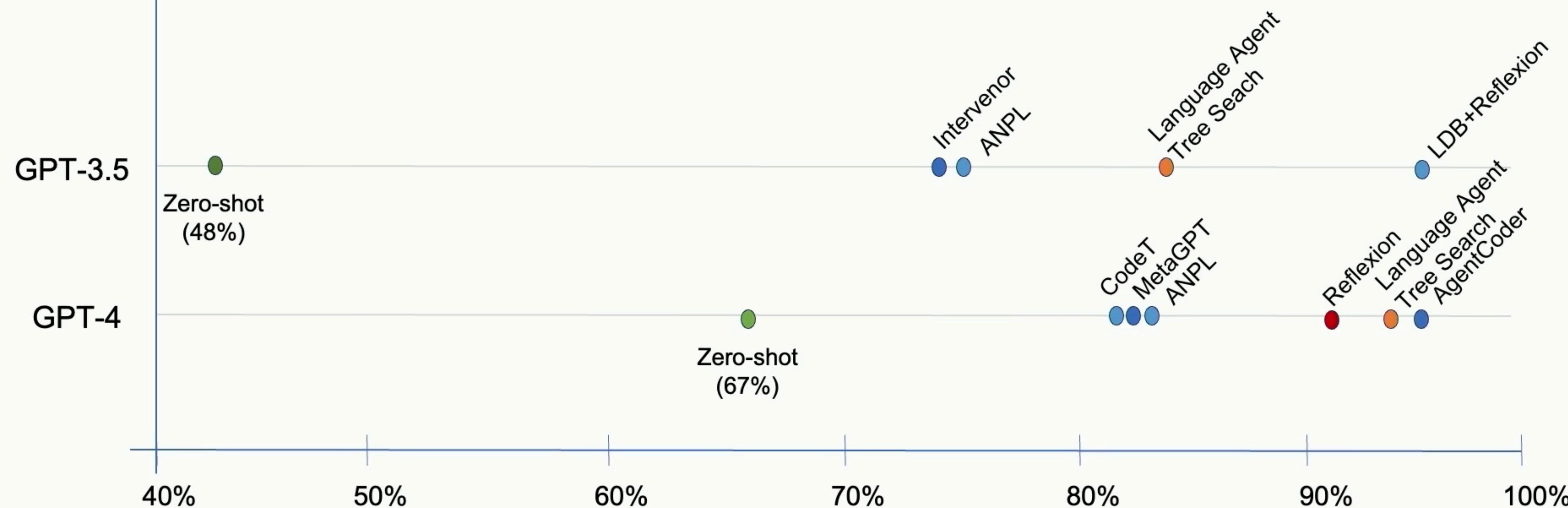
Coding benchmark (HumanEval)



[Thanks to Joaquin Dominguez and John Santerre for help with analysis.]

Andrew Ng

Coding benchmark (HumanEval)



Agentic Reasoning Design Patterns

1. Reflection



robust technology

2. Tool use



emerging technology

3. Planning

4. Multi-agent collaboration

1. Reflection



Please write code for {task}

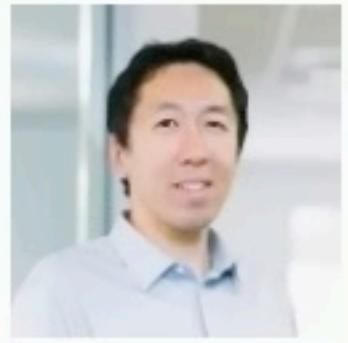


Coder Agent
(LLM)

Recommended reading:

- Self-Refine: Iterative Refinement with Self-Feedback, Madaan et al. (2023)
- Reflexion: Language Agents with Verbal Reinforcement Learning, Shinn et al., (2023)

1. Reflection



Please write code for {task}



Coder Agent
(LLM)

Recommended reading:

- Self-Refine: Iterative Refinement with Self-Feedback, Madaan et al. (2023)
- Reflexion: Language Agents with Verbal Reinforcement Learning, Shinn et al., (2023)

1. Reflection



Please write code for {task}

def do_task(x): ...



Coder Agent
(LLM)

Recommended reading:

- Self-Refine: Iterative Refinement with Self-Feedback, Madaan et al. (2023)
- Reflexion: Language Agents with Verbal Reinforcement Learning, Shinn et al., (2023)

1. Reflection



Please write code for {task}

def do_task(x) : ...



Coder Agent
(LLM)

Here's code intended for {task}:

```
def do_task (x) :  
    ...
```

Check the code carefully for correctness, style and efficiency, and give constructive criticism for how to improve it.

Recommended reading:

- Self-Refine: Iterative Refinement with Self-Feedback, Madaan et al. (2023)
- Reflexion: Language Agents with Verbal Reinforcement Learning, Shinn et al., (2023)

1. Reflection



Please write code for {task}

def do_task(x) : ...



Coder Agent
(LLM)

Here's code intended for {task}:

```
def do_task (x) :  
    ...
```

Check the code carefully for correctness, style and efficiency, and give constructive criticism for how to improve it.

Recommended reading:

- Self-Refine: Iterative Refinement with Self-Feedback, Madaan et al. (2023)
- Reflexion: Language Agents with Verbal Reinforcement Learning, Shinn et al., (2023)

1. Reflection



Please write code for {task}

def do_task(x) : ...



Coder Agent
(LLM)

Here's code intended for {task}:

```
def do_task (x) :  
    ...
```

Check the code carefully for correctness, style and efficiency, and give constructive criticism for how to improve it.

Recommended reading:

- Self-Refine: Iterative Refinement with Self-Feedback, Madaan et al. (2023)
- Reflexion: Language Agents with Verbal Reinforcement Learning, Shinn et al., (2023)

1. Reflection



Please write code for {task}

def do_task(x) : ...



Coder Agent
(LLM)

Here's code intended for {task}:

```
def do_task (x) :  
    ...
```

Check the code carefully for correctness, style and efficiency, and give constructive criticism for how to improve it.

Recommended reading:

- Self-Refine: Iterative Refinement with Self-Feedback, Madaan et al. (2023)
- Reflexion: Language Agents with Verbal Reinforcement Learning, Shinn et al., (2023)

1. Reflection



Please write code for {task}

Here's code intended for {task}:

```
def do_task (x) :  
    ...
```

Check the code carefully for correctness, style and efficiency, and give constructive criticism for how to improve it.

def do_task(x) : ...



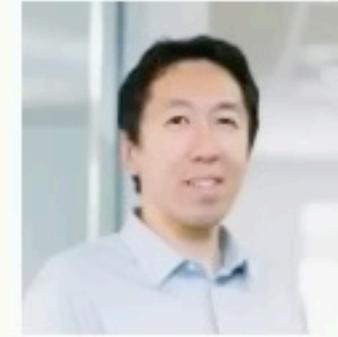
Coder Agent
(LLM)

There's a bug on line 5. Fix it by ...

Recommended reading:

- Self-Refine: Iterative Refinement with Self-Feedback, Madaan et al. (2023)
- Reflexion: Language Agents with Verbal Reinforcement Learning, Shinn et al., (2023)

1. Reflection



Please write code for {task}

Here's code intended for {task}:

```
def do_task (x) :  
    ...
```

Check the code carefully for correctness, style and efficiency, and give constructive criticism for how to improve it.

def do_task(x) : ...

def do_task_v2(x) :



Coder Agent
(LLM)

There's a bug on line 5. Fix it by ...

It failed Unit Test 3. Try changing ...

Recommended reading:

- Self-Refine: Iterative Refinement with Self-Feedback, Madaan et al. (2023)
- Reflexion: Language Agents with Verbal Reinforcement Learning, Shinn et al., (2023)

1. Reflection



Please write code for {task}

Here's code intended for {task}:

```
def do_task (x) :  
    ...
```

Check the code carefully for correctness, style and efficiency, and give constructive criticism for how to improve it.

def do_task(x) : ...

def do_task_v2(x) :



Coder Agent
(LLM)

There's a bug on line 5. Fix it by ...

It failed Unit Test 3. Try changing ...

Recommended reading:

- Self-Refine: Iterative Refinement with Self-Feedback, Madaan et al. (2023)
- Reflexion: Language Agents with Verbal Reinforcement Learning, Shinn et al., (2023)

1. Reflection



Please write code for {task}

Here's code intended for {task}:

```
def do_task (x) :  
    ...
```

Check the code carefully for correctness, style and efficiency, and give constructive criticism for how to improve it.

def do_task(x) : ...

def do_task_v2(x) :

def do_task_v3(x) :



Coder Agent
(LLM)

There's a bug on line 5. Fix it by ...

It failed Unit Test 3. Try changing ...

Recommended reading:

- Self-Refine: Iterative Refinement with Self-Feedback, Madaan et al. (2023)
- Reflexion: Language Agents with Verbal Reinforcement Learning, Shinn et al., (2023)

1. Reflection



Please write code for {task}

Here's code intended for {task}:

```
def do_task (x):  
    ...
```

Check the code carefully for correctness, style and efficiency, and give constructive criticism for how to improve it.

def do_task(x): ...

def do_task_v2(x):

def do_task_v3(x):



Coder Agent
(LLM)

There's a bug on line 5. Fix it by ...

It failed Unit Test 3. Try changing ...

Recommended reading:

- Self-Refine: Iterative Refinement with Self-Feedback, Madaan et al. (2023)
- Reflexion: Language Agents with Verbal Reinforcement Learning, Shinn et al., (2023)

1. Reflection



Please write code for {task}

def do_task(x) : ...

def do_task_v2(x) :

def do_task_v3(x) :



Coder Agent
(LLM)

Here's code intended for {task}:

```
def do_task (x) :  
    ...
```

Check the code carefully for correctness, style and efficiency, and give constructive criticism for how to improve it.

There's a bug on line 5. Fix it by ...

It failed Unit Test 3. Try changing ...

Recommended reading:

- Self-Refine: Iterative Refinement with Self-Feedback, Madaan et al. (2023)
- Reflexion: Language Agents with Verbal Reinforcement Learning, Shinn et al., (2023)

1. Reflection



Please write code for {task}

Here's code intended for {task}:

```
def do_task (x) :  
    ...
```

Check the code carefully for correctness, style and efficiency, and give constructive criticism for how to improve it.

def do_task(x) : ...

def do_task_v2(x) :



Coder Agent
(LLM)

There's a bug on line 5. Fix it by ...

It failed Unit Test 3. Try changing ...



Critic Agent
(LLM)

Recommended reading:

- Self-Refine: Iterative Refinement with Self-Feedback, Madaan et al. (2023)
- Reflexion: Language Agents with Verbal Reinforcement Learning, Shinn et al., (2023)

1. Reflection



Please write code for {task}

Here's code intended for {task}:

```
def do_task (x) :  
    ...
```

Check the code carefully for correctness, style and efficiency, and give constructive criticism for how to improve it.

def do_task(x) : ...

def do_task_v2(x) :

def do_task_v3(x) :



Coder Agent
(LLM)

There's a bug on line 5. Fix it by ...

It failed Unit Test 3. Try changing ...



Critic Agent
(LLM)

Recommended reading:

- Self-Refine: Iterative Refinement with Self-Feedback, Madaan et al. (2023)
- Reflexion: Language Agents with Verbal Reinforcement Learning, Shinn et al., (2023)

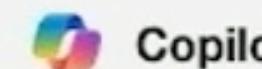
2. Tool use

Web search tool



You

What is the best coffee maker according to reviewers?



Copilot

Searching for best coffee maker according to reviewers

Code execution tool



NG

You

If I invest \$100 at compound 7% interest for 12 years, what do I have at the end?

```
principal = 100  
interest_rate = 0.07  
years = 12  
value = principal*(1 + interest_rate)**years
```

Analysis

- Code Execution
- Wolfram Alpha
- Bearly Code Interpreter

Research

- Search engine
- Web browsing
- Wikipedia

Productivity

- Email
- Calendar
- Cloud Storage

Images

- Image generation (e.g., Dall-E)
- Image captioning
- Object detection

Recommended reading:

- Gorilla: Large Language Model Connected with Massive APIs, Patil et al. (2023)
- MM-REACT: Prompting ChatGPT for Multimodal Reasoning and Action, Yang et al. (2023)

3. Planning

Request: Please generate an image where a girl is reading a book, and her pose is the same as the boy in the image example.jpg, then please describe the new image with your voice.



example.jpg

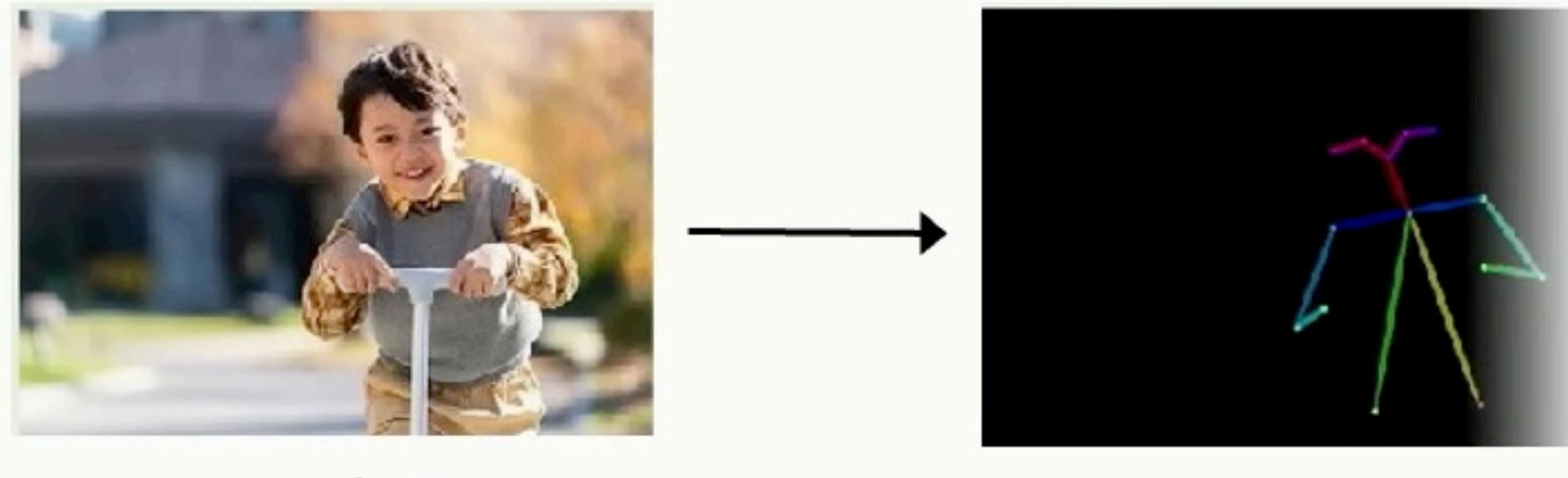
[Example adapted from HuggingGPT paper]

Recommended reading:

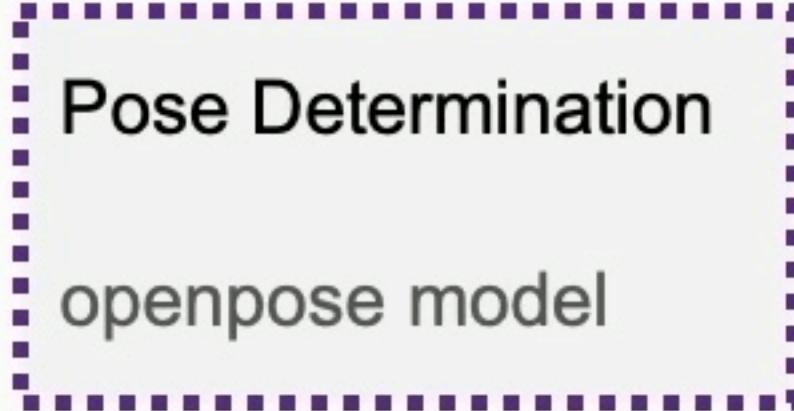
- Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, Wei et al., (2022)
- HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face, Shen et al. (2023)

3. Planning

Request: Please generate an image where a girl is reading a book, and her pose is the same as the boy in the image example.jpg, then please describe the new image with your voice.



example.jpg



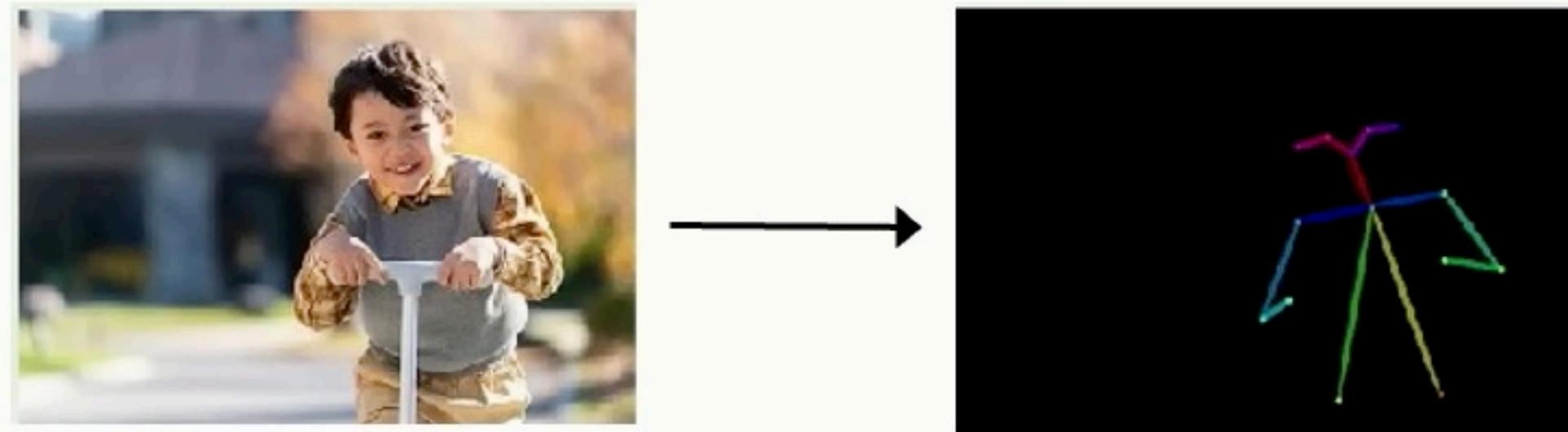
[Example adapted from HuggingGPT paper]

Recommended reading:

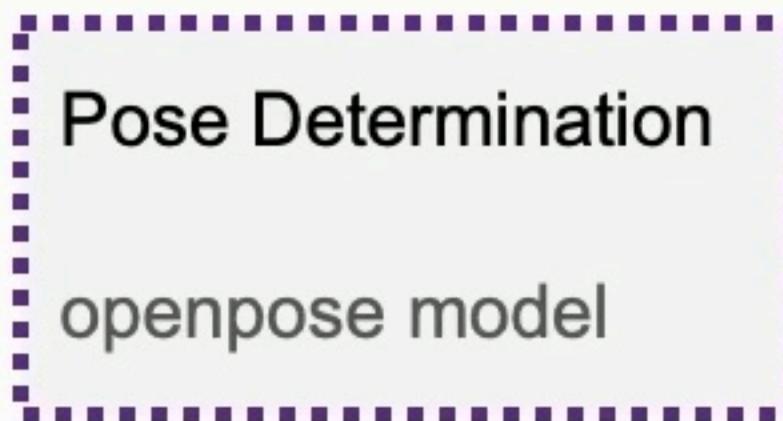
- Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, Wei et al., (2022)
- HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face, Shen et al. (2023)

3. Planning

Request: Please generate an image where a girl is reading a book, and her pose is the same as the boy in the image example.jpg, then please describe the new image with your voice.



example.jpg



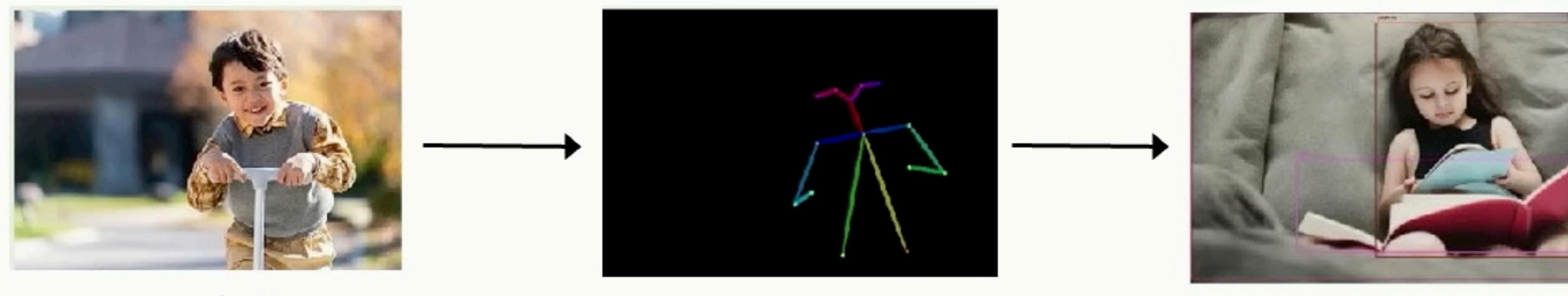
[Example adapted from HuggingGPT paper]

Recommended reading:

- Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, Wei et al., (2022)
- HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face, Shen et al. (2023)

3. Planning

Request: Please generate an image where a girl is reading a book, and her pose is the same as the boy in the image example.jpg, then please describe the new image with your voice.



example.jpg

Pose Determination
openpose model

Pose-to-Image
google/vit model

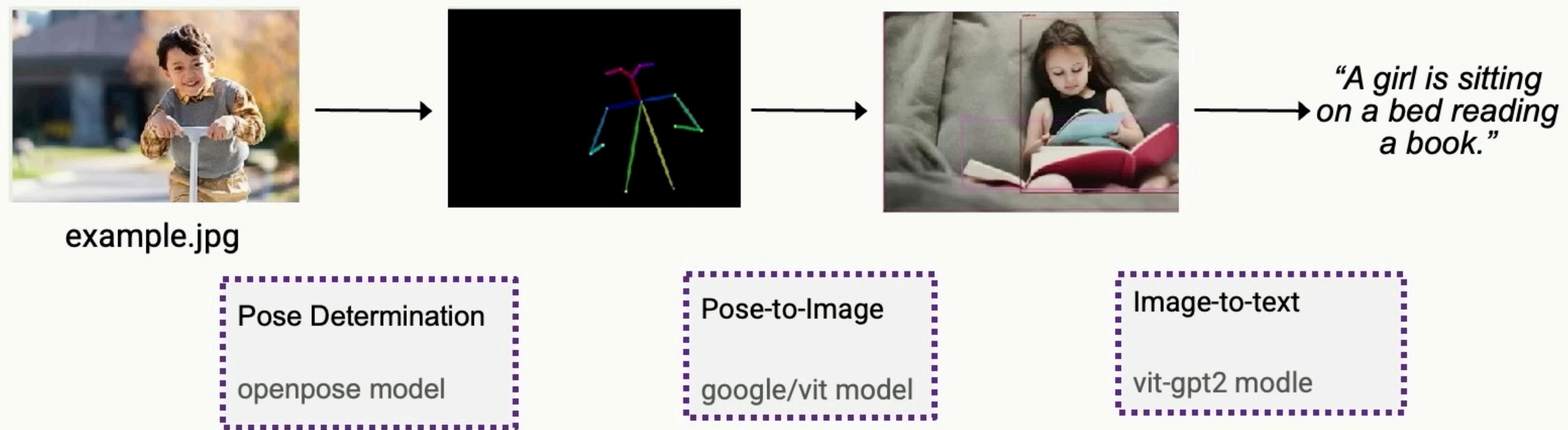
[Example adapted from HuggingGPT paper]

Recommended reading:

- Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, Wei et al., (2022)
- HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face, Shen et al. (2023)

3. Planning

Request: Please generate an image where a girl is reading a book, and her pose is the same as the boy in the image example.jpg, then please describe the new image with your voice.



[Example adapted from HuggingGPT paper]

Recommended reading:

- Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, Wei et al., (2022)
- HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face, Shen et al. (2023)

4. Multiagent collaboration



Multiagent Debate

Task	Single agent	Multi-agent
Biographies	66.0%	73.8%
MMLU	63.9%	71.1%
Chess move	29.3%	45.2%

(Du et al., 2023)

Recommended reading:

- Communicative Agents for Software Development, Qian et al., (2023)
- AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation, Wu et al. (2023)

Agentic Reasoning Design Patterns

1. Reflection

- Self-Refine: Iterative Refinement with Self-Feedback, Madaan et al. (2023)
- Reflexion: Language Agents with Verbal Reinforcement Learning, Shinn et al., (2023)

2. Tool use

- Gorilla: Large Language Model Connected with Massive APIs, Patil et al. (2023)
- MM-REACT: Prompting ChatGPT for Multimodal Reasoning and Action, Yang et al. (2023)

3. Planning

- Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, Wei et al., (2022)
- HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face, Shen et al. (2023)

4. Multi-agent collaboration

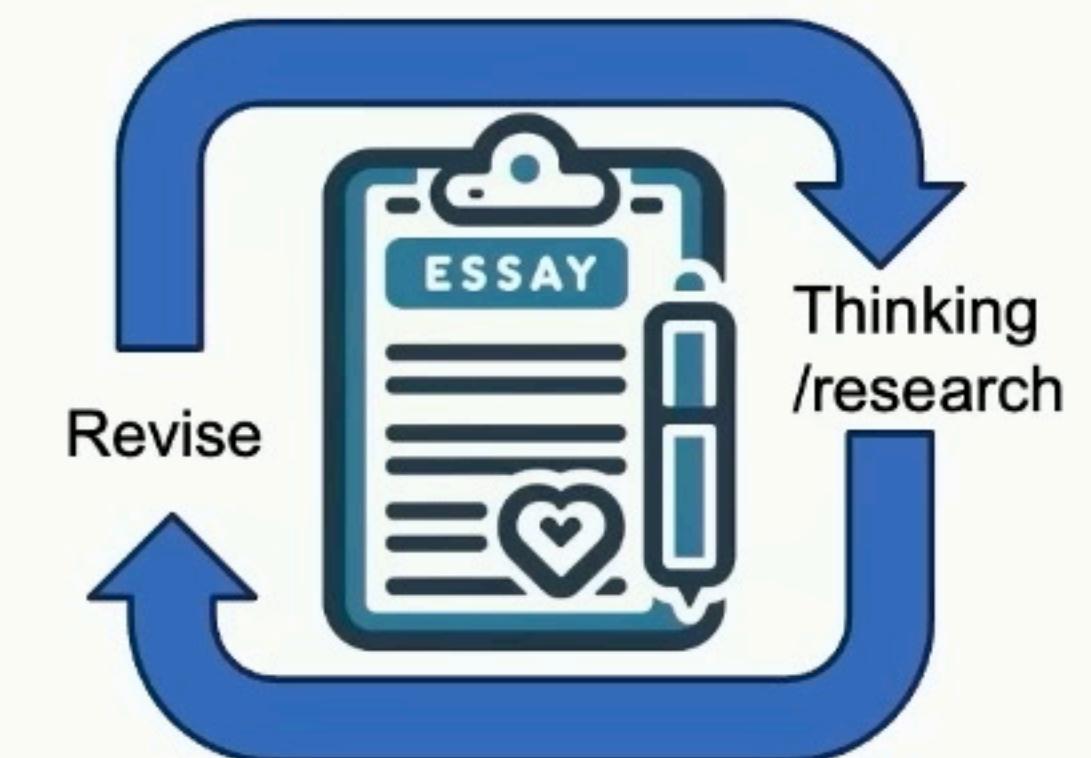
- Communicative Agents for Software Development, Qian et al., (2023)
- AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation, Wu et al. (2023)

Conclusion

The set of tasks that AI can do will expand dramatically because of agentic workflows.

We have to get used to delegating tasks to AI agents and patiently wait for a response.

Fast token generation is important. Generating more tokens even from a lower quality LLM can give good Results.



If you're looking forward to running GPT-5/Claude 4/Gemini 2.0 (zero shot) on your application, you might already be able to get similar performance with agentic reasoning on an earlier model.

Andrew Ng