LLM Agents

Alexander Wettig

`awettig@princeton.edu`

2025/04/09

Credit to Kilian Lieret, Ofir Press, Carlos Jimenez, Shunyu Yao for slides!

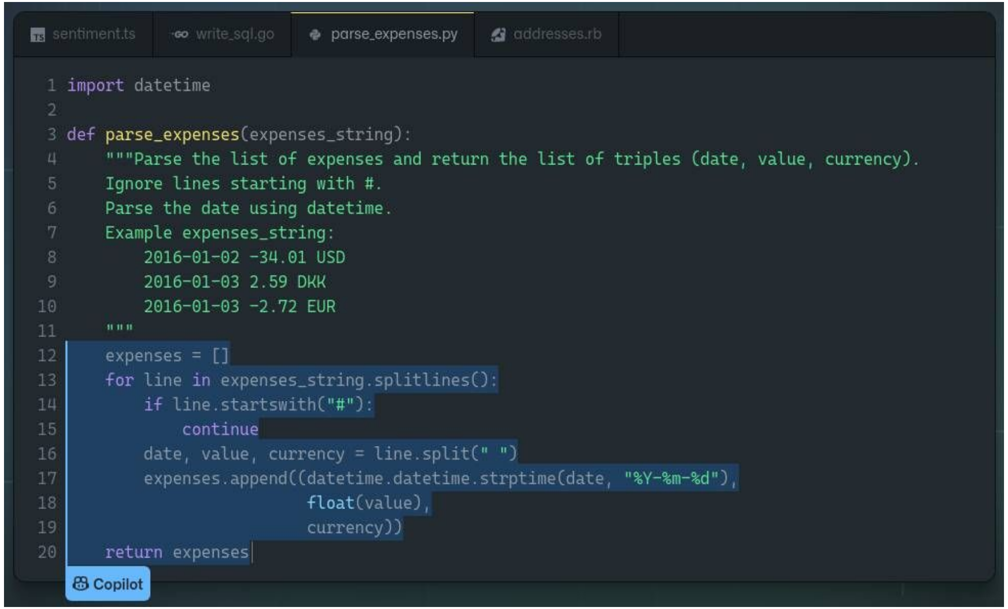
Goals

- Building and evaluating LLM agents
- Focus on applications
  - Software engineering *(our focus today)*
  - GUI navigation
  - Information retrieval
  - Customer assistance
  - ...
  - *Many use cases with real economic value*
- Terminology
  - Agents, tools, environments, scaffolds, frameworks, compound systems

LMs for Software Engineering

Language models are next token predictors

⇒ June 2021: **Github Copilot** powered by OpenAI Codex



```python
import datetime

def parse_expenses(expenses_string):
    """Parse the list of expenses and return the list of triples (date, value, currency).
    Ignore lines starting with #.
    Parse the date using datetime.
    Example expenses_string:
        2016-01-02 -34.01 USD
        2016-01-03 2.59 DKK
        2016-01-03 -2.72 EUR
    """
    expenses = []
    for line in expenses_string.splitlines():
        if line.startswith("#"):
            continue
        date, value, currency = line.split(" ")
        expenses.append((datetime.datetime.strptime(date, "%Y-%m-%d"),
                         float(value),
                         currency))
    return expenses
```

LMs for Software Engineering

Language models are next token predictors

⇒ June 2021: **Github Copilot** powered by OpenAI Codex

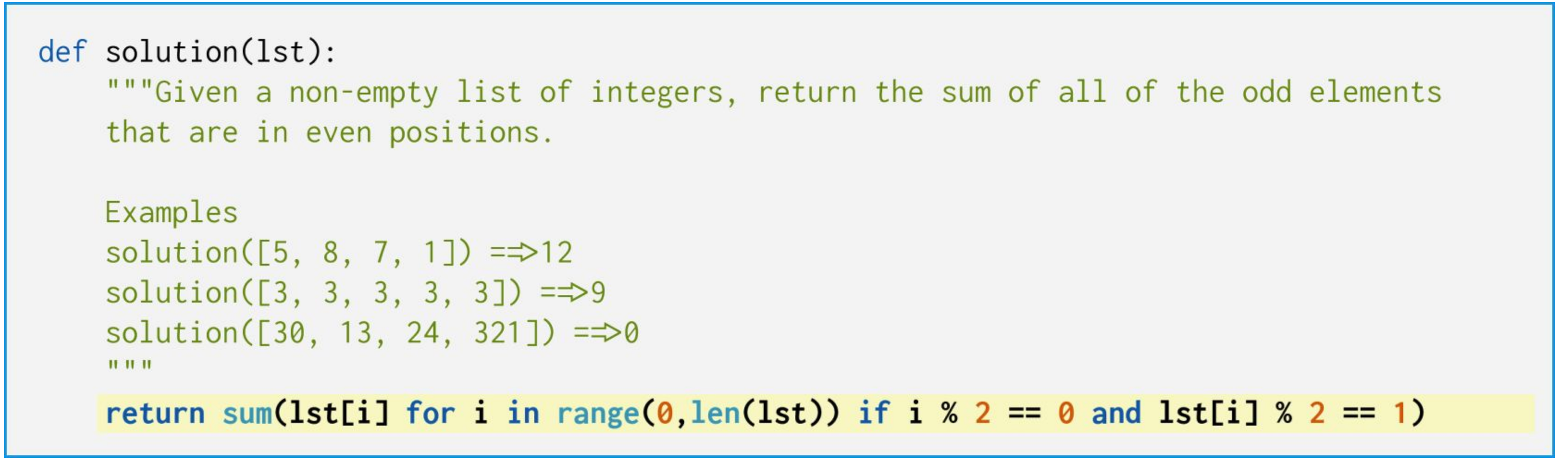
⇒ July 2021: OpenAI releases paper on Codex...

Evaluating Large Language Models Trained on Code

... introducing new benchmark, **HumanEval**

HumanEval

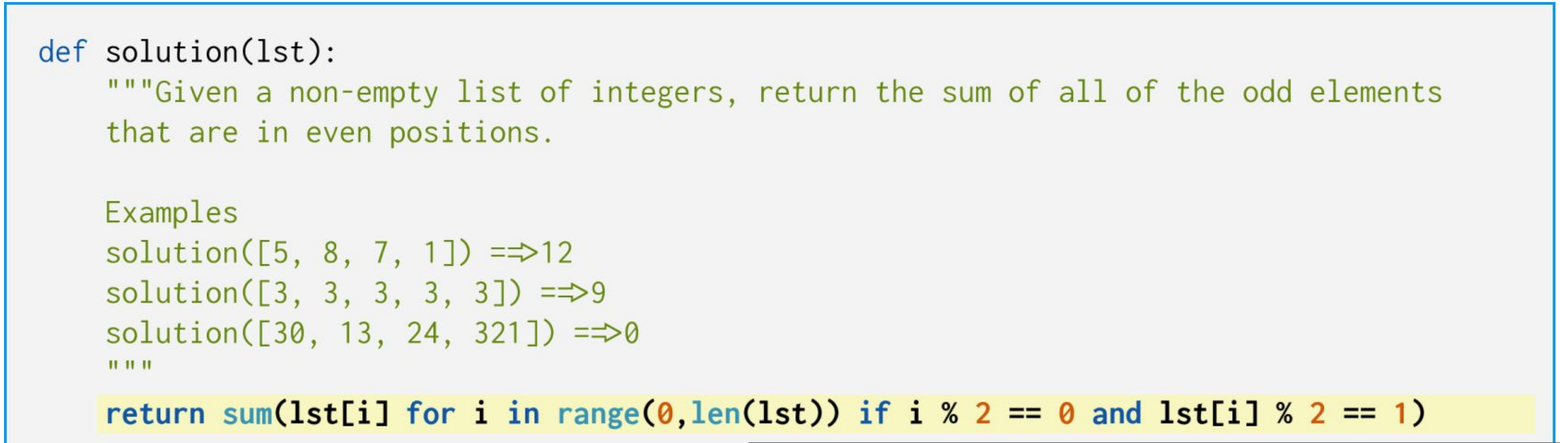
164 handwritten programming problems



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

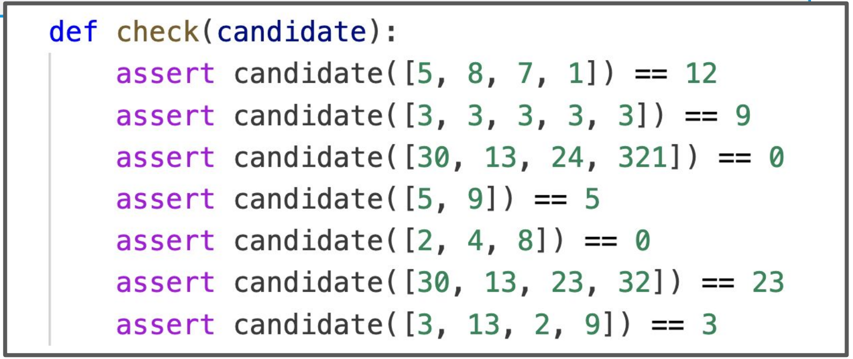
HumanEval

164 handwritten programming problems



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

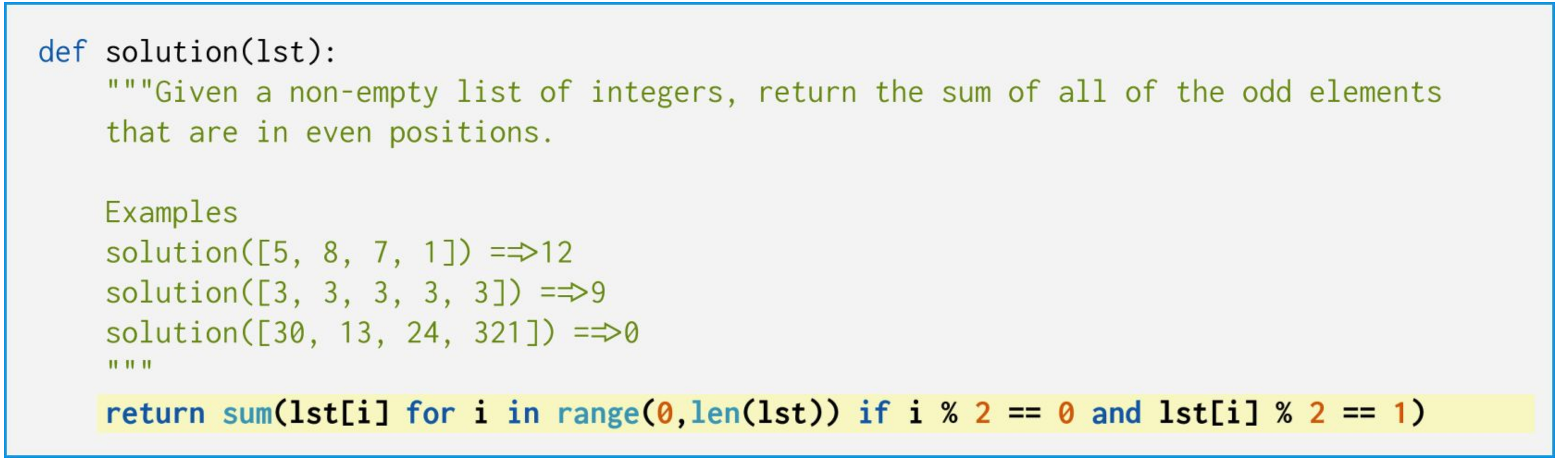
Check correctness of generated code by running unit tests!



```python
def check(candidate):
    assert candidate([5, 8, 7, 1]) == 12
    assert candidate([3, 3, 3, 3, 3]) == 9
    assert candidate([30, 13, 24, 321]) == 0
    assert candidate([5, 9]) == 5
    assert candidate([2, 4, 8]) == 0
    assert candidate([30, 13, 23, 32]) == 23
    assert candidate([3, 13, 2, 9]) == 3
```

HumanEval

164 handwritten programming problems

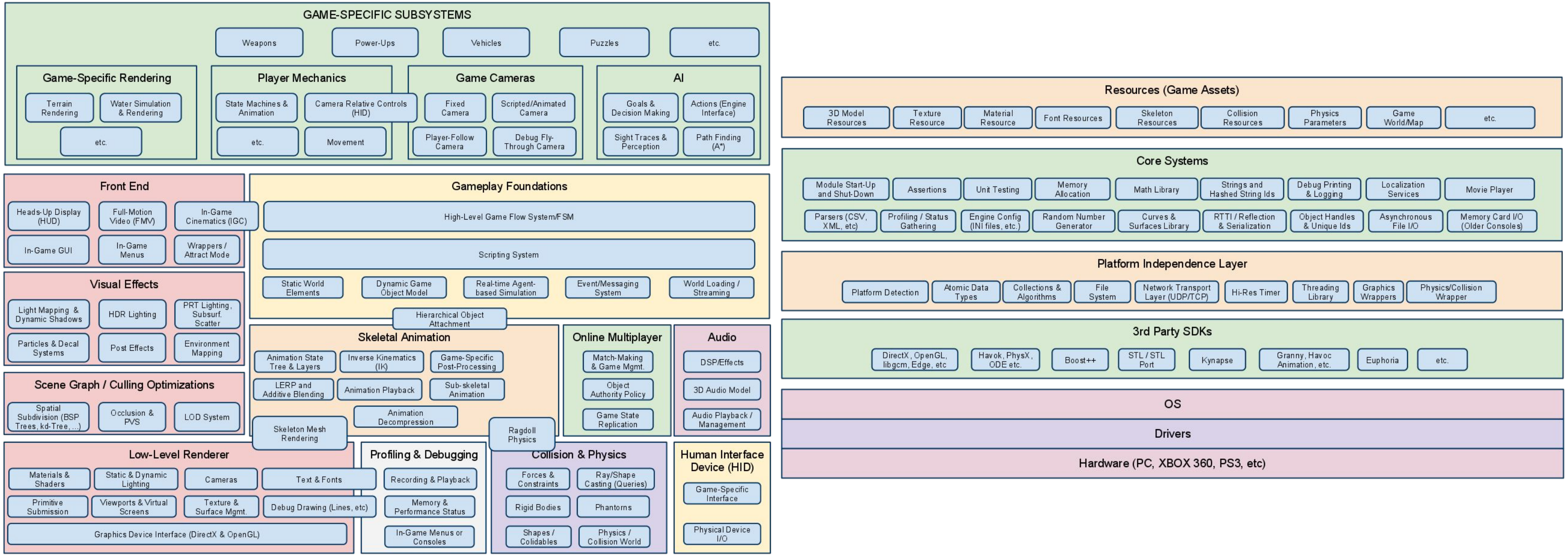


```python
def solution(lst):
    """Given a non-empty list of integers, return the sum of all of the odd elements
    that are in even positions.

    Examples
    solution([5, 8, 7, 1]) ==>12
    solution([3, 3, 3, 3, 3]) ==>9
    solution([30, 13, 24, 321]) ==>0
    """
    return sum(lst[i] for i in range(0,len(lst)) if i % 2 == 0 and lst[i] % 2 == 1)
```

Great benchmark for evaluating **code generation & autocomplete!**

🤔 But does this capture your typical coding work?

A. Complexity of real codebases



GAME-SPECIFIC SUBSYSTEMS

Weapons
Power-Ups
Vehicles
Puzzles
etc.

Game-Specific Rendering

Terrain Rendering
Water Simulation & Rendering
etc.

Player Mechanics

State Machines & Animation
Camera Relative Controls (HID)
etc.
Movement

Game Cameras

Fixed Camera
Scripted/Animated Camera
Player-Follow Camera
Debug Fly-Through Camera

AI

Goals & Decision Making
Actions (Engine Interface)
Sight Traces & Perception
Path Finding (A*)

Front End

Heads-Up Display (HUD)
Full-Motion Video (FMV)
In-Game Cinematics (IGC)
In-Game GUI
In-Game Menus
Wrappers / Attract Mode

Visual Effects

Light Mapping & Dynamic Shadows
HDR Lighting
PRT Lighting, Subsurf. Scatter
Particles & Decal Systems
Post Effects
Environment Mapping

Scene Graph / Culling Optimizations

Spatial Subdivision (BSP Trees, kd-Tree, ...)
Occlusion & PVS
LOD System

Gameplay Foundations

High-Level Game Flow System/FSM
Scripting System
Static World Elements
Dynamic Game Object Model
Real-time Agent-based Simulation
Event/Messaging System
World Loading / Streaming
Hierarchical Object Attachment

Skeletal Animation

Animation State Tree & Layers
Inverse Kinematics (IK)
Game-Specific Post-Processing
LERP and Additive Blending
Animation Playback
Sub-skeletal Animation
Animation Decompression
Skeleton Mesh Rendering
Ragdoll Physics

Online Multiplayer

Match-Making & Game Mgmt.
Object Authority Policy
Game State Replication

Audio

DSP/Effects
3D Audio Model
Audio Playback / Management

Low-Level Renderer

Materials & Shaders
Static & Dynamic Lighting
Cameras
Text & Fonts
Primitive Submission
Viewports & Virtual Screens
Texture & Surface Mgmt.
Debug Drawing (Lines, etc)
Graphics Device Interface (DirectX & OpenGL)

Profiling & Debugging

Recording & Playback
Memory & Performance Status
In-Game Menus or Consoles

Collision & Physics

Forces & Constraints
Ray/Shape Casting (Queries)
Rigid Bodies
Phantoms
Shapes / Collidables
Physics / Collision World

Human Interface Device (HID)

Game-Specific Interface
Physical Device I/O

Resources (Game Assets)

3D Model Resources
Texture Resource
Material Resource
Font Resources
Skeleton Resources
Collision Resources
Physics Parameters
Game World/Map
etc.

Core Systems

Module Start-Up and Shut-Down
Assertions
Unit Testing
Memory Allocation
Math Library
Strings and Hashed String Ids
Debug Printing & Logging
Localization Services
Movie Player
Parsers (CSV, XML, etc)
Profiling / Status Gathering
Engine Config (INI files, etc.)
Random Number Generator
Curves & Surfaces Library
RTTI / Reflection & Serialization
Object Handles & Unique Ids
Asynchronous File I/O
Memory Card I/O (Older Consoles)

Platform Independence Layer

Platform Detection
Atomic Data Types
Collections & Algorithms
File System
Network Transport Layer (UDP/TCP)
Hi-Res Timer
Threading Library
Graphics Wrappers
Physics/Collision Wrapper

3rd Party SDKs

DirectX, OpenGL, libgcm, Edge, etc
Havok, PhysX, ODE etc.
Boost++
STL / STL Port
Kynapse
Granny, Havoc Animation, etc.
Euphoria
etc.

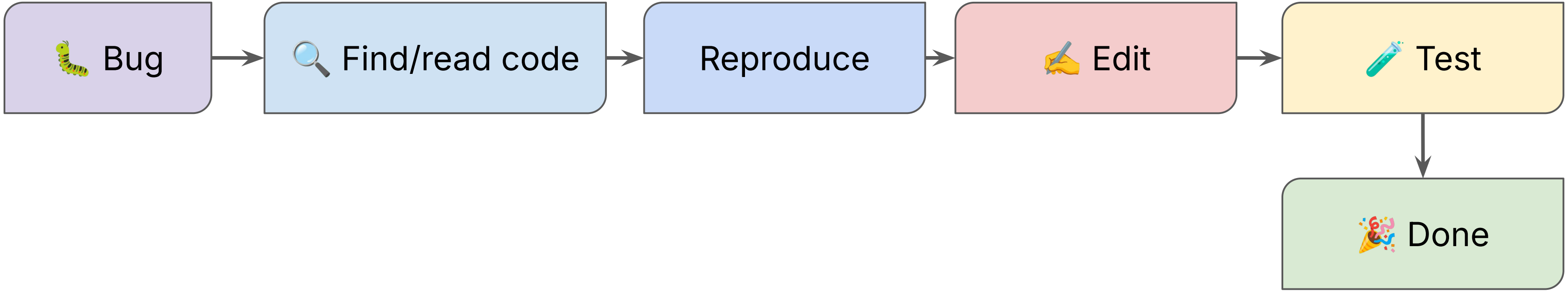
OS

Drivers

Hardware (PC, XBOX 360, PS3, etc)

http://www.gameenginebook.com/figures.html

B. Complexity of development workflows



🐛 Bug

🔍 Find/read code

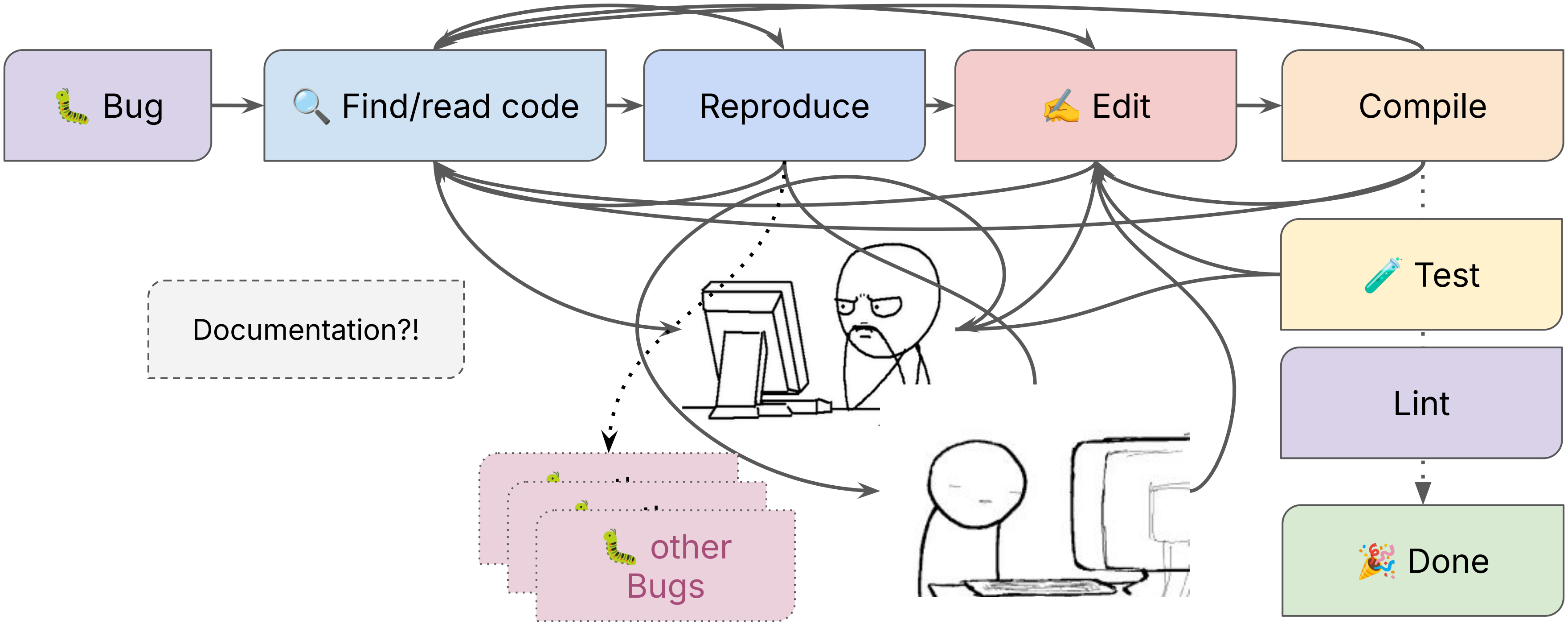
Reproduce

✍️ Edit

🧪 Test

🎉 Done

B. Complexity of development workflows



🐛 Bug

🔍 Find/read code

Reproduce

✍️ Edit

Compile

🧪 Test

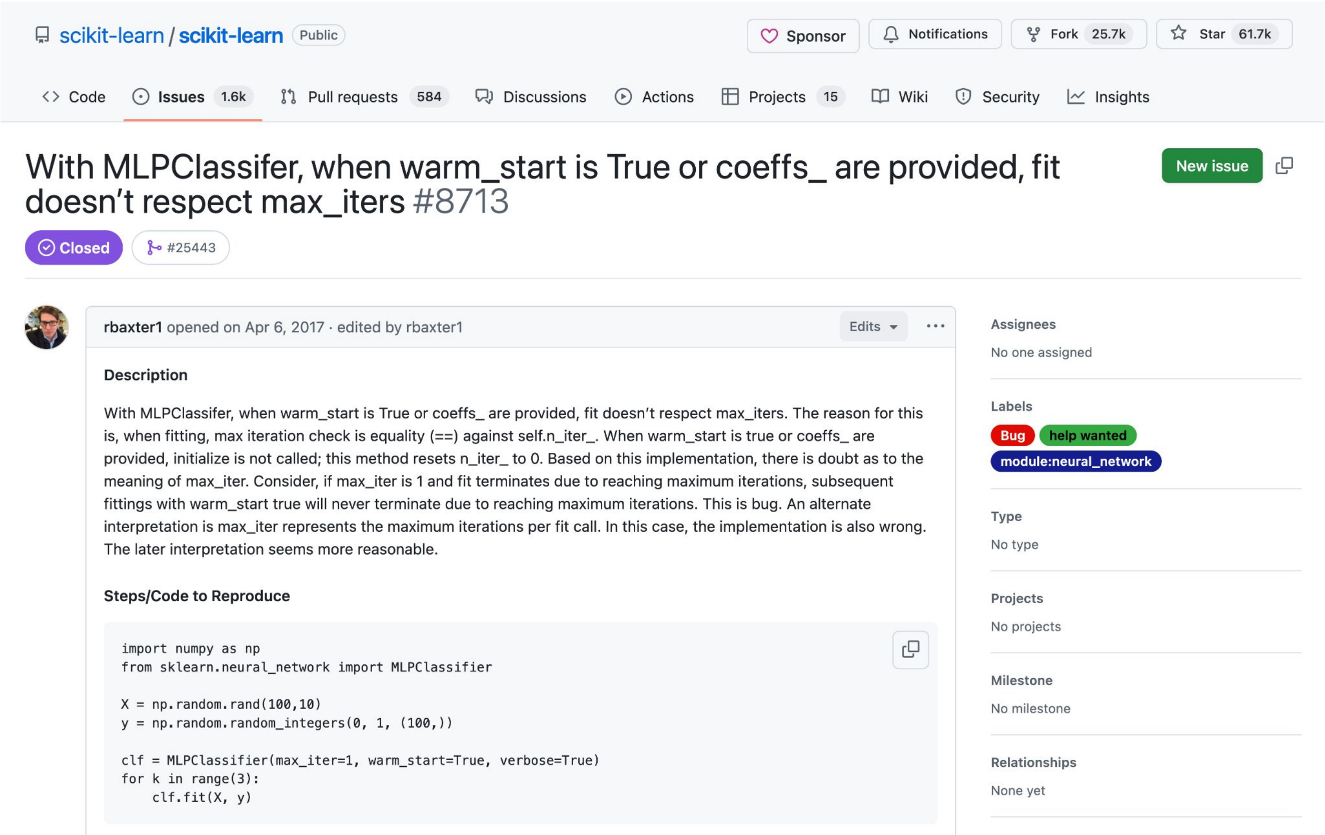
Lint

🎉 Done

Documentation?!

🐛 other Bugs

SWE-bench: Can Language Models Resolve Real-World GitHub Issues?



With MLPClassifer, when warm_start is True or coeffs_ are provided, fit doesn't respect max_iters #8713

rbaxter1 opened on Apr 6, 2017 · edited by rbaxter1

Description

With MLPClassifer, when warm_start is True or coeffs_ are provided, fit doesn't respect max_iters. The reason for this is, when fitting, max iteration check is equality (==) against self.n_iter_. When warm_start is true or coeffs_ are provided, initialize is not called; this method resets n_iter_ to 0. Based on this implementation, there is doubt as to the meaning of max_iter. Consider, if max_iter is 1 and fit terminates due to reaching maximum iterations, subsequent fittings with warm_start true will never terminate due to reaching maximum iterations. This is bug. An alternate interpretation is max_iter represents the maximum iterations per fit call. In this case, the implementation is also wrong. The later interpretation seems more reasonable.

Steps/Code to Reproduce

```
import numpy as np
from sklearn.neural_network import MLPClassifier

X = np.random.rand(100,10)
y = np.random.random_integers(0, 1, (100,))

clf = MLPClassifier(max_iter=1, warm_start=True, verbose=True)
for k in range(3):
    clf.fit(X, y)
```

Assignees
No one assigned

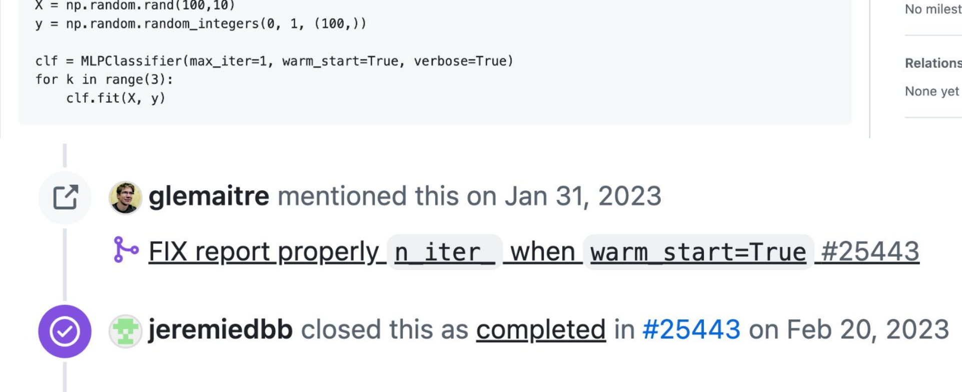
Labels
Bug
help wanted
module:neural_network

Type
No type

Projects
No projects

Milestone
No milestone

Relationships
None yet



```
X = np.random.rand(100,10)
y = np.random.random_integers(0, 1, (100,))

clf = MLPClassifier(max_iter=1, warm_start=True, verbose=True)
for k in range(3):
    clf.fit(X, y)
```

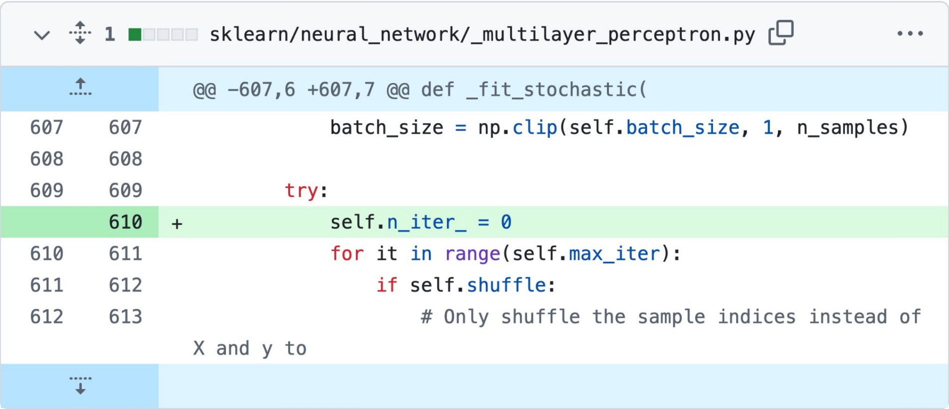
glemaitre mentioned this on Jan 31, 2023

FIX report properly `n_iter_` when `warm_start=True` #25443

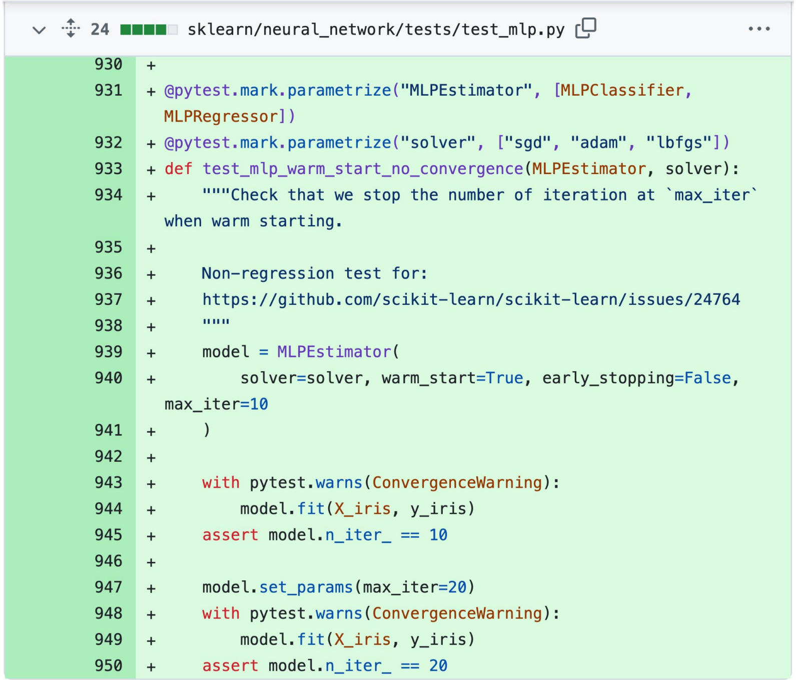
jeremiedbb closed this as completed in #25443 on Feb 20, 2023

With test changes

Pull request



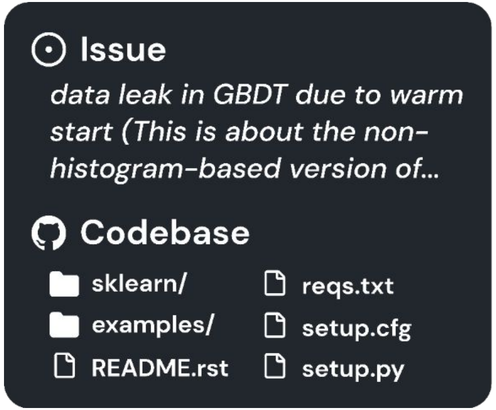
```
@@ -607,6 +607,7 @@ def _fit_stochastic(
             batch_size = np.clip(self.batch_size, 1, n_samples)

         try:
+            self.n_iter_ = 0
             for it in range(self.max_iter):
                 if self.shuffle:
                     # Only shuffle the sample indices instead of X and y to
```



```
+
+@pytest.mark.parametrize("MLPEstimator", [MLPClassifier, MLPRegressor])
+@pytest.mark.parametrize("solver", ["sgd", "adam", "lbfgs"])
+def test_mlp_warm_start_no_convergence(MLPEstimator, solver):
+    """Check that we stop the number of iteration at `max_iter` when warm starting.
+
+    Non-regression test for:
+    https://github.com/scikit-learn/scikit-learn/issues/24764
+    """
+    model = MLPEstimator(
+        solver=solver, warm_start=True, early_stopping=False, max_iter=10
+    )
+
+    with pytest.warns(ConvergenceWarning):
+        model.fit(X_iris, y_iris)
+    assert model.n_iter_ == 10
+
+    model.set_params(max_iter=20)
+    with pytest.warns(ConvergenceWarning):
+        model.fit(X_iris, y_iris)
+    assert model.n_iter_ == 20
```

SWE-bench: Can Language Models Resolve Real-World GitHub Issues?

SWE-bench: Can Language Models Resolve Real-World GitHub Issues?



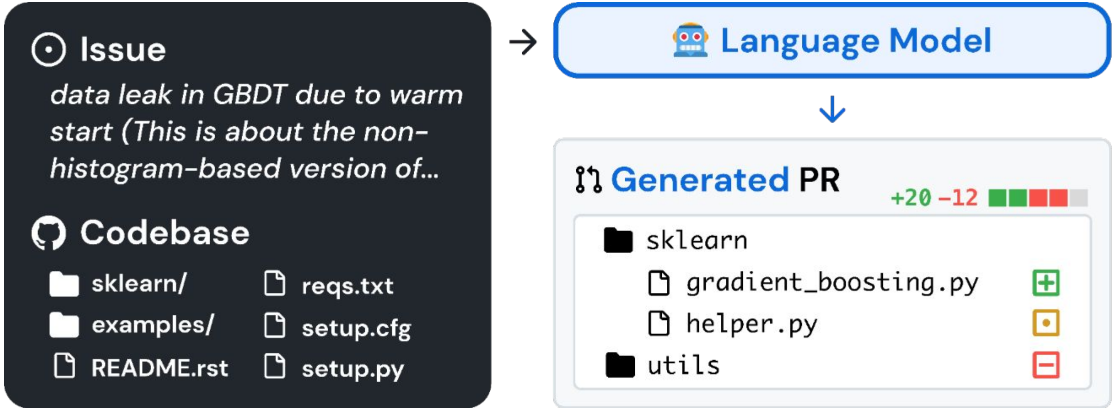
Issue

data leak in GBDT due to warm start (This is about the non-histogram-based version of...

Codebase

- sklearn/
- examples/
- README.rst
- reqs.txt
- setup.cfg
- setup.py

SWE-bench: Can Language Models Resolve Real-World GitHub Issues?



Issue

data leak in GBDT due to warm start (This is about the non-histogram-based version of...

Codebase

- sklearn/
- examples/
- README.rst
- reqs.txt
- setup.cfg
- setup.py

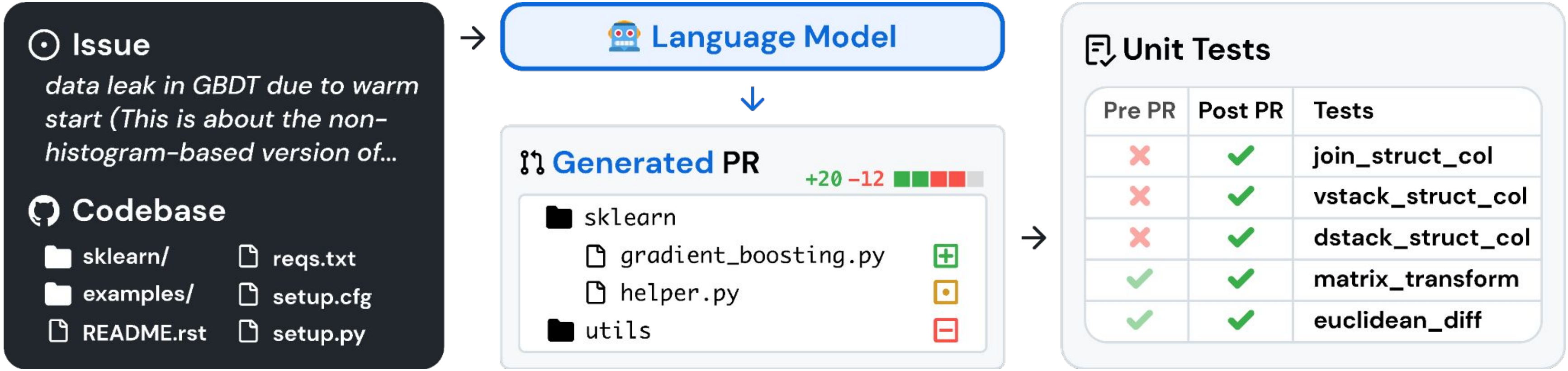
→ **Language Model**

↓

Generated PR +20 −12 🟩🟩🟥🟥⬜

```
sklearn
    gradient_boosting.py    ⊞
    helper.py               ⊡
utils                       ⊟
```

SWE-bench: Can Language Models Resolve Real-World GitHub Issues?



Issue

data leak in GBDT due to warm start (This is about the non-histogram-based version of...

Codebase

sklearn/
examples/
README.rst
reqs.txt
setup.cfg
setup.py

Language Model

Generated PR +20 −12

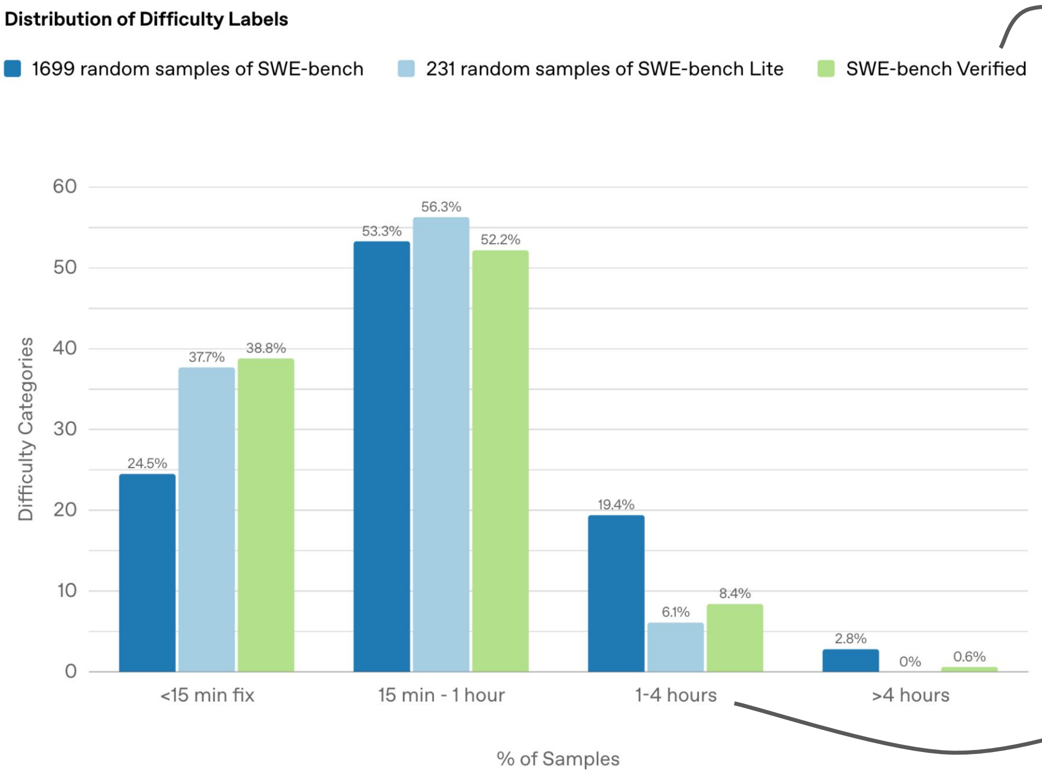
```
sklearn
    gradient_boosting.py
    helper.py
utils
```

Unit Tests

Pre PR	Post PR	Tests
✗	✓	join_struct_col
✗	✓	vstack_struct_col
✗	✓	dstack_struct_col
✓	✓	matrix_transform
✓	✓	euclidean_diff

Many new challenges:

- Understanding large code bases
- Long input contexts
- Edits to multiple files



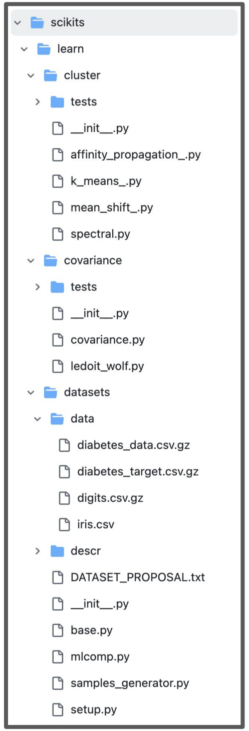
High-quality subset of SWE-bench (well-specified issues and tests)

Tracks real software developer activity

openai.com/index/introducing-swe-bench-verified/

RAG: Simplest approach for SWE-bench

= retrieval augmented-generation



- scikits
  - learn
    - cluster
      - tests
      - __init__.py
      - affinity_propagation_.py
      - k_means_.py
      - mean_shift_.py
      - spectral.py
    - covariance
      - tests
      - __init__.py
      - covariance.py
      - ledoit_wolf.py
    - datasets
      - data
        - diabetes_data.csv.gz
        - diabetes_target.csv.gz
        - digits.csv.gz
        - iris.csv
      - descr
      - DATASET_PROPOSAL.txt
      - __init__.py
      - base.py
      - mlcomp.py
      - samples_generator.py
      - setup.py

🐛 Issue

Retrieve relevant files

Use as prompt

Add to prompt

🤖LLM
instruction-tuned

```
--- a/sklearn/linear_model/huber.py
+++ b/sklearn/linear_model/huber.py
@@ -231,6 +231,9 @@ class HuberRegressor(LinearModel, RegressorMixin,
    BaseEstimator):
       self.tol = tol

+   def _validate_data(self, X, y=None):
+       return X.astype(np.float64) if X.dtype == np.bool else X
+
    def fit(self, X, y, sample_weight=None):
       """Fit the model according to the given training data.
```

▼ **Instructions** • 1 line

You will be provided with a partial code base and an issue statement explaining a problem to resolve.

▶ **Issue** • 67 lines

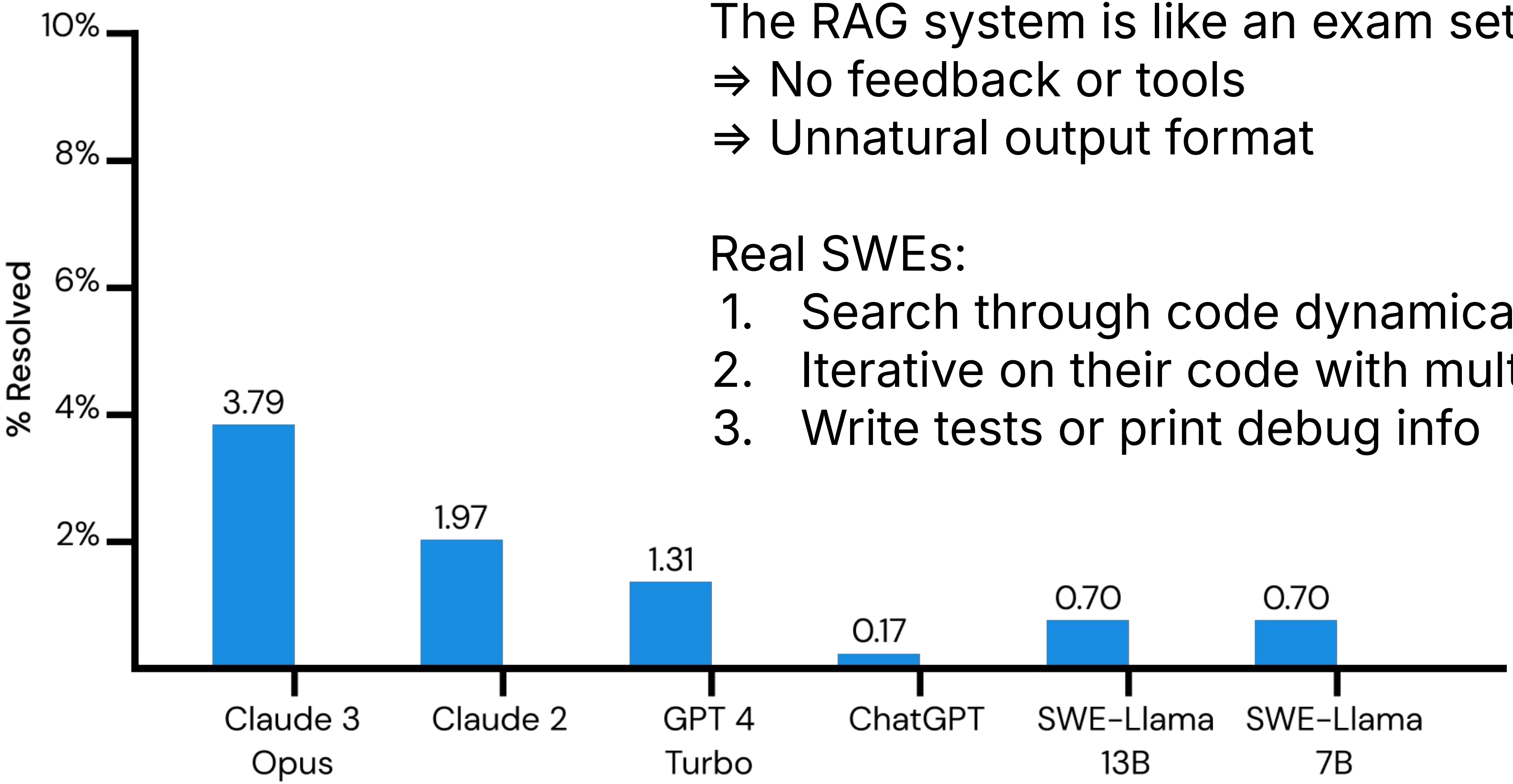
▼ **Code (retrieved documents)** • 1431 lines

  ▶ **`README.rst`** • 132 lines

  ▶ **`sphinx/ext/napoleon/docstring.py`** • 1295 lines

▶ **Additional Instructions** • 57 lines

RAG performs poorly on SWE-bench

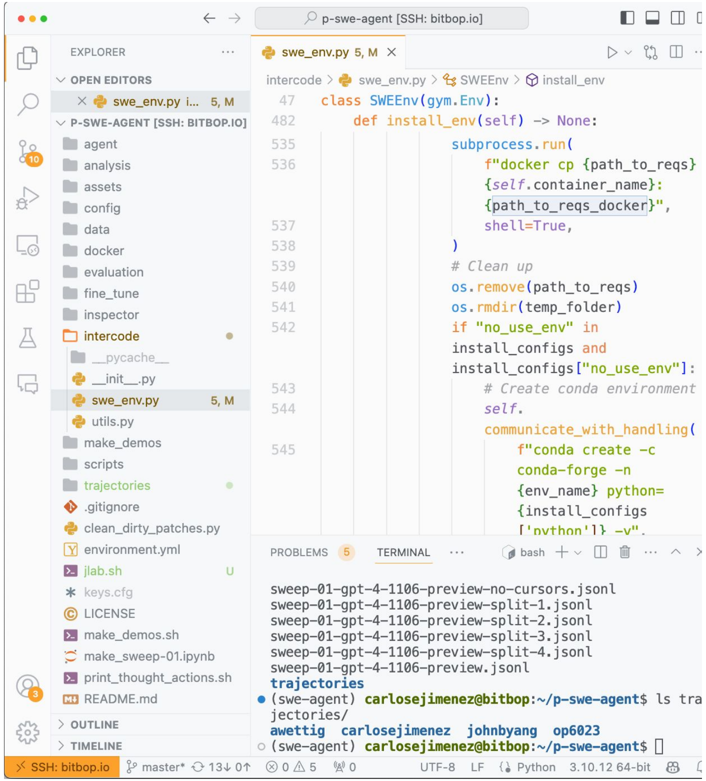


The RAG system is like an exam setting
⇒ No feedback or tools
⇒ Unnatural output format

Real SWEs:

1. Search through code dynamically
2. Iterative on their code with multiple edits
3. Write tests or print debug info

We want LLM-Computer interaction



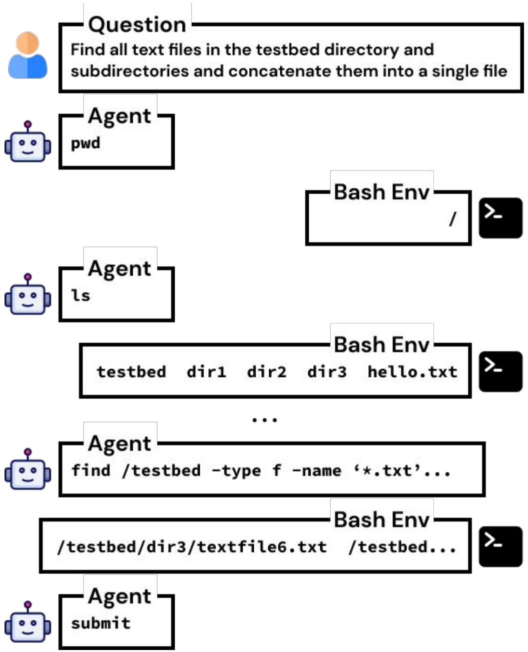
```python
class SWEEnv(gym.Env):
    def install_env(self) -> None:
                subprocess.run(
                    f"docker cp {path_to_reqs}
                    {self.container_name}:
                    {path_to_reqs_docker}",
                    shell=True,
                )
                # Clean up
                os.remove(path_to_reqs)
                os.rmdir(temp_folder)
                if "no_use_env" in
                install_configs and
                install_configs["no_use_env"]:
                    # Create conda environment
                    self.
                    communicate_with_handling(
                        f"conda create -c
                        conda-forge -n
                        {env_name} python=
                        {install_configs
                        ['python']} -v",
```

```
sweep-01-gpt-4-1106-preview-no-cursors.jsonl
sweep-01-gpt-4-1106-preview-split-1.jsonl
sweep-01-gpt-4-1106-preview-split-2.jsonl
sweep-01-gpt-4-1106-preview-split-3.jsonl
sweep-01-gpt-4-1106-preview-split-4.jsonl
sweep-01-gpt-4-1106-preview.jsonl
trajectories
(swe-agent) carlosejimenez@bitbop:~/p-swe-agent$ ls tra
jectories/
awettig  carlosejimenez  johnbyang  op6023
(swe-agent) carlosejimenez@bitbop:~/p-swe-agent$ 
```

Humans use tools like VSCode / vim

- Lots of visual information
- High granularity of commands

We want LLM-Computer interaction



Question

Find all text files in the testbed directory and subdirectories and concatenate them into a single file

Agent

```
pwd
```

Bash Env

```
/
```

Agent

```
ls
```

Bash Env

```
testbed  dir1  dir2  dir3  hello.txt
```

...

Agent

```
find /testbed -type f -name '*.txt'...
```

Bash Env

```
/testbed/dir3/textfile6.txt  /testbed...
```

Agent

```
submit
```

(Yang et. al 2023)

Better approach:
Connect LLM to terminal environment
LM output → bash
bash output → LM

Basically a **chat**, but instead of
Human ⇔ ChatBot
we have
ChatBot ⇔ Computer

Observation

Bug: with MLPClassifier,
when warm_start is True, fit doesn’t respect max_iters
...

Observation

Bug: with MLPClassifier,
when warm_start is True, fit doesn’t respect max_iters
...

Action

```
open train.py
```

Observation

```
Bug: with MLPClassifier,
when warm_start is True, fit doesn't respect max_iters
...
```

Action

```
open train.py
```

Observation

```
import numpy as np
from scipy.special improt expit

def inplace_identity(X):
"""Simply leave the input array
unchanged. Paramters
...
```

Observation

```
Bug: with MLPClassifier,
when warm_start is True, fit doesn’t respect max_iters
...
```

Action

```
open train.py
```

Observation

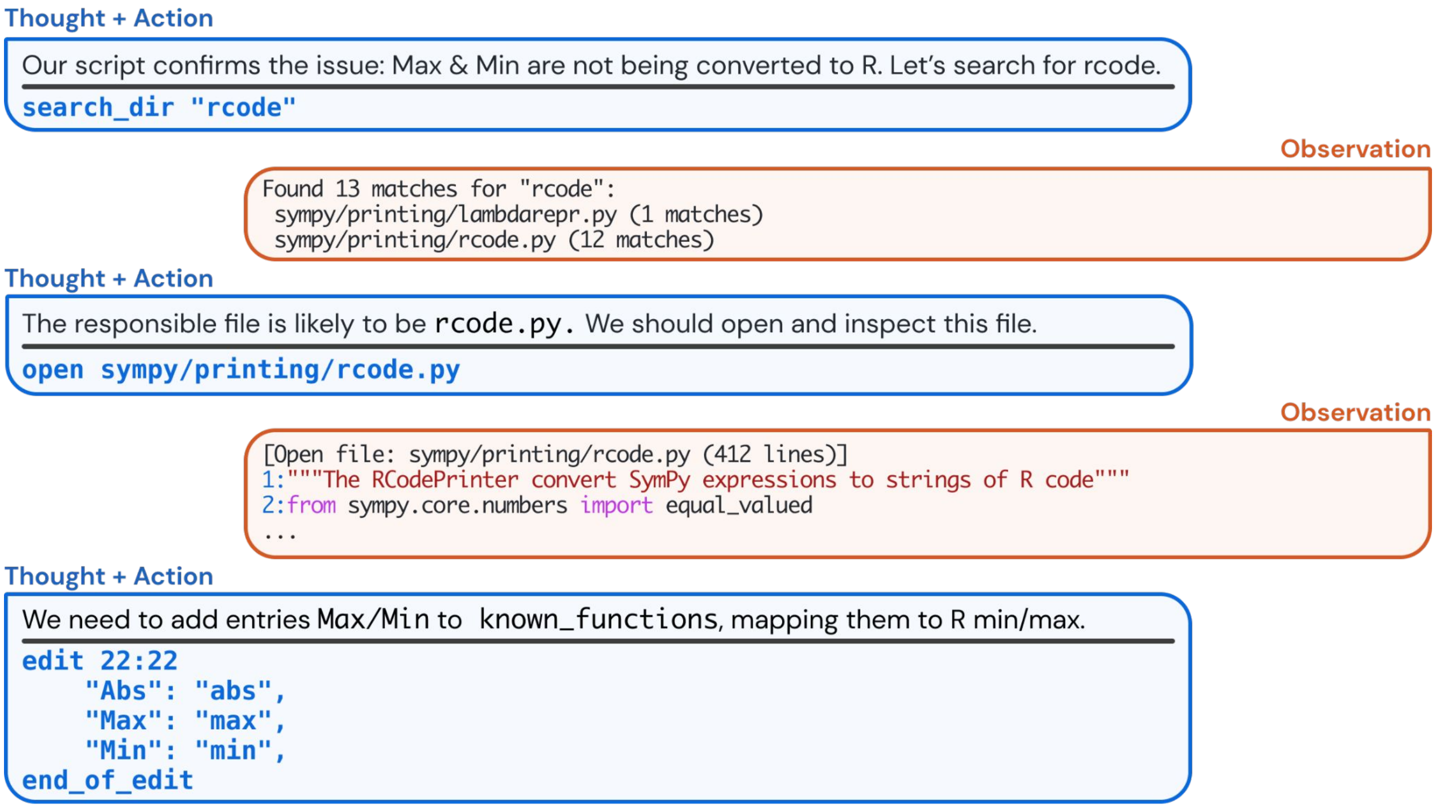
```
import numpy as np
from scipy.special improt expit

def inplace_identity(X):
“””Simply leave the input array
unchanged. Paramters
...
```

Action

```
edit_line 8
delta[Z == 0] = 0
```

conversation ⇒ **trajectory**

Usually LM provides both a verbalized **Thought** and the **Action**



Thought + Action

Our script confirms the issue: Max & Min are not being converted to R. Let's search for rcode.

```
search_dir "rcode"
```

Observation

```
Found 13 matches for "rcode":
 sympy/printing/lambdarepr.py (1 matches)
 sympy/printing/rcode.py (12 matches)
```

Thought + Action

The responsible file is likely to be `rcode.py`. We should open and inspect this file.

```
open sympy/printing/rcode.py
```

Observation

```
[Open file: sympy/printing/rcode.py (412 lines)]
1:"""The RCodePrinter convert SymPy expressions to strings of R code"""
2:from sympy.core.numbers import equal_valued
...
```

Thought + Action

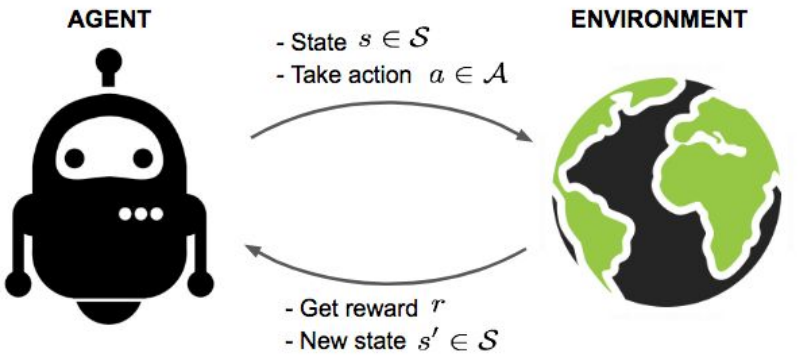
We need to add entries `Max/Min` to `known_functions`, mapping them to R min/max.

```
edit 22:22
    "Abs": "abs",
    "Max": "max",
    "Min": "min",
end_of_edit
```

This was introduced by Yao et al., 2022 “**ReAct:** Synergizing **Re**asoning and **Act**ing in Language Models”

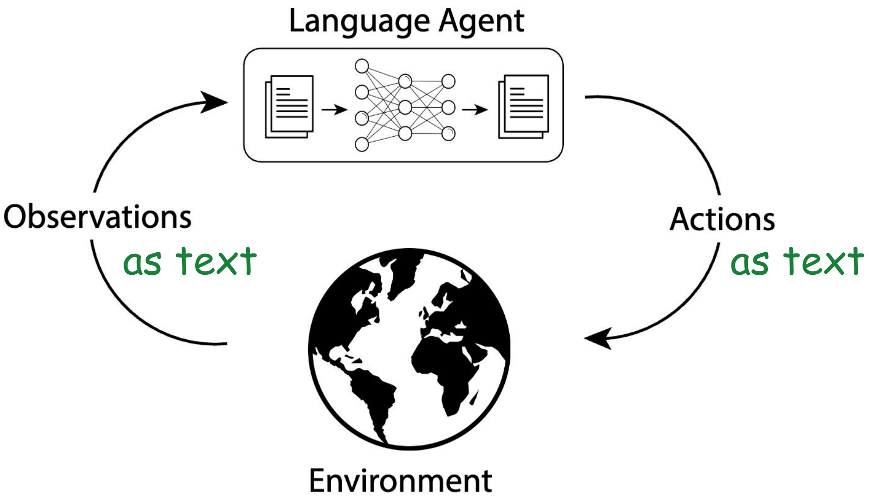
So what is an LLM agent?

Traditional RL view


AGENT
ENVIRONMENT
- State $s \in \mathcal{S}$
- Take action $a \in \mathcal{A}$
- Get reward r
- New state $s' \in \mathcal{S}$

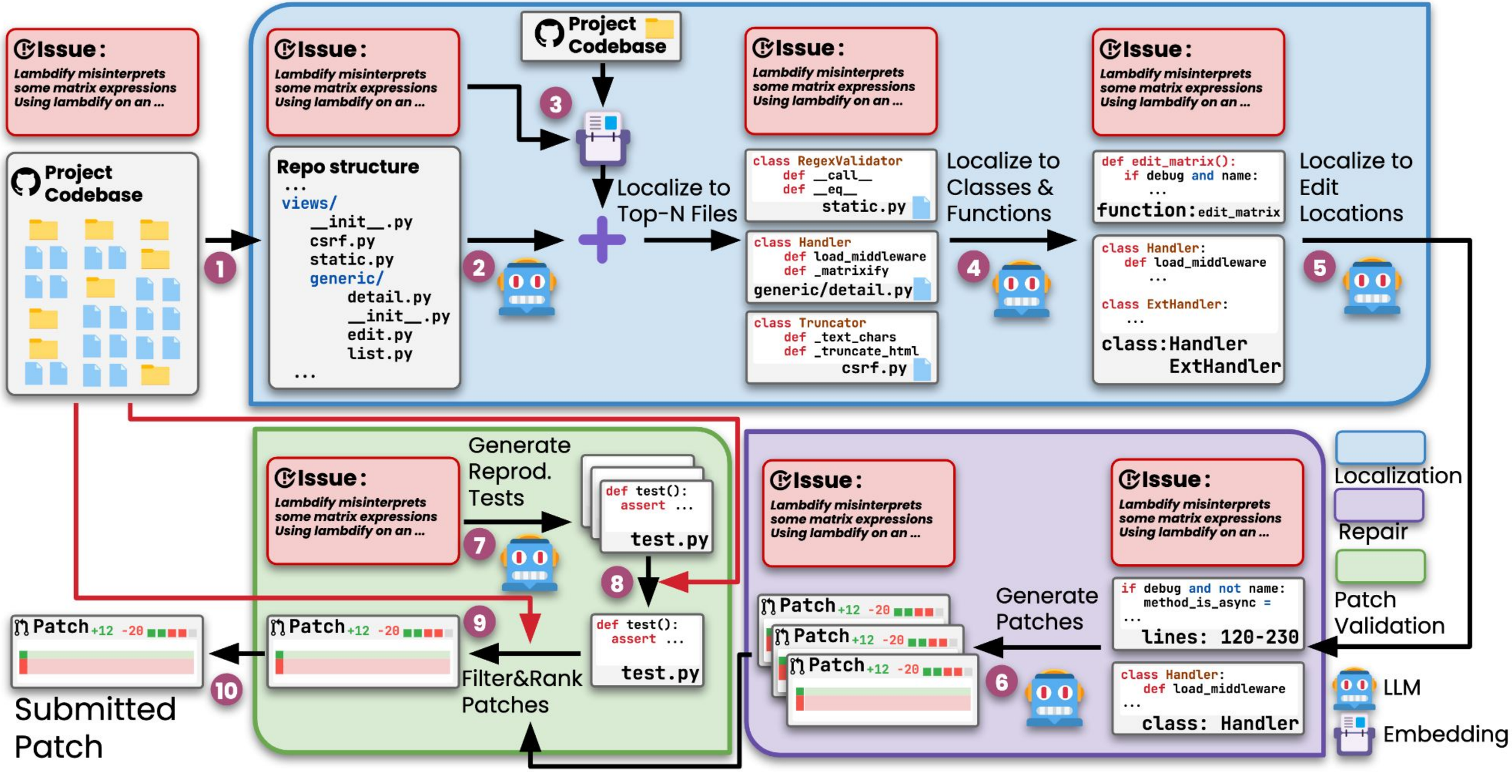
lilianweng.github.io/posts/2018-02-19-rl-overview/

Language / LM / LLM agents



Sumers et al., 2024. Cognitive Architectures for Language Agents

What is NOT an LLM agent?



```
...
views/
    __init__.py
    csrf.py
    static.py
    generic/
        detail.py
        __init__.py
        edit.py
        list.py
...
```

```
class RegexValidator
    def __call__
    def __eq__
            static.py
```

```
class Handler
    def load_middleware
    def _matrixify
generic/detail.py
```

```
class Truncator
    def _text_chars
    def _truncate_html
            csrf.py
```

```
def edit_matrix():
    if debug and name:
        ...
function: edit_matrix
```

```
class Handler:
    def load_middleware
        ...
class ExtHandler:
    ...
class: Handler
       ExtHandler
```

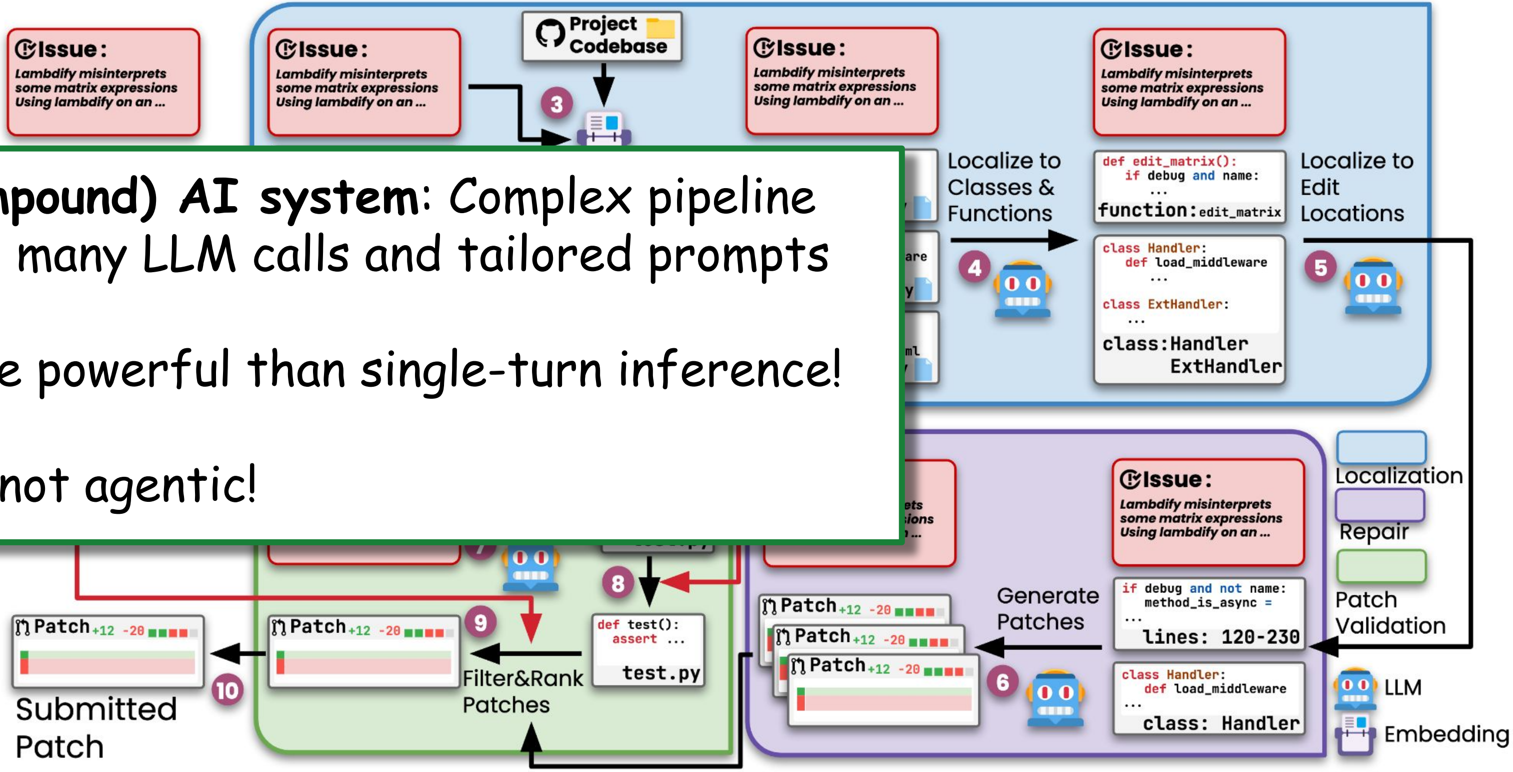
```
def test():
    assert ...
        test.py
```

```
if debug and not name:
    method_is_async =
...
  lines: 120-230
```

```
class Handler:
    def load_middleware
...
  class: Handler
```

Xia et al., 2024, **Agentless**: Demystifying LLM-based Software Engineering Agents

What is NOT an LLM agent?



(compound) AI system: Complex pipeline with many LLM calls and tailored prompts

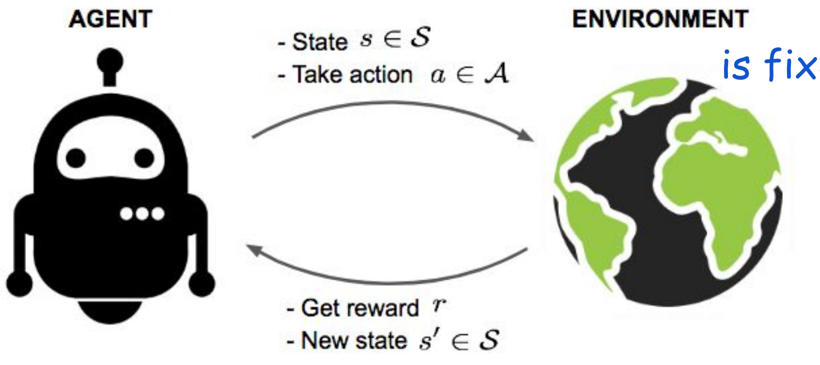
More powerful than single-turn inference!

But not agentic!

Xia et al., 2024, **Agentless**: Demystifying LLM-based Software Engineering Agents

So what is an LLM agent?

Traditional RL view

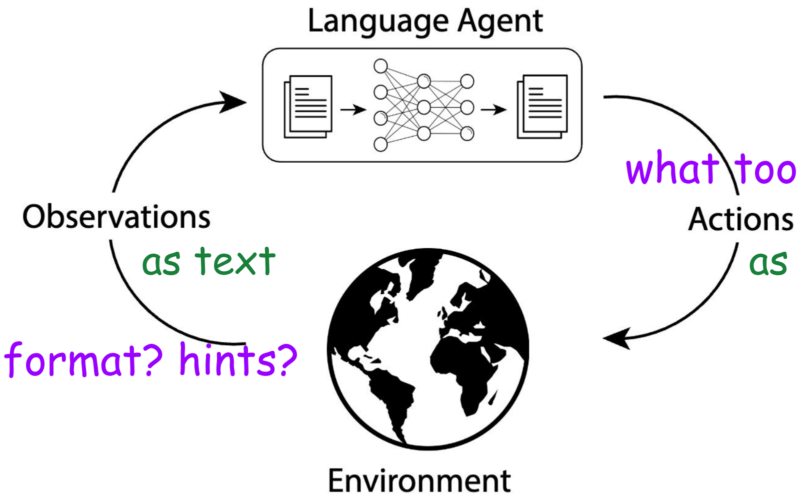

AGENT
- State $s \in \mathcal{S}$
- Take action $a \in \mathcal{A}$
ENVIRONMENT
- Get reward r
- New state $s' \in \mathcal{S}$

is fixed

lilianweng.github.io/posts/2018-02-19-rl-overview/

maximize reward of agent policy

Language / LM / LLM agents



what tools?

as text

as text

format? hints?

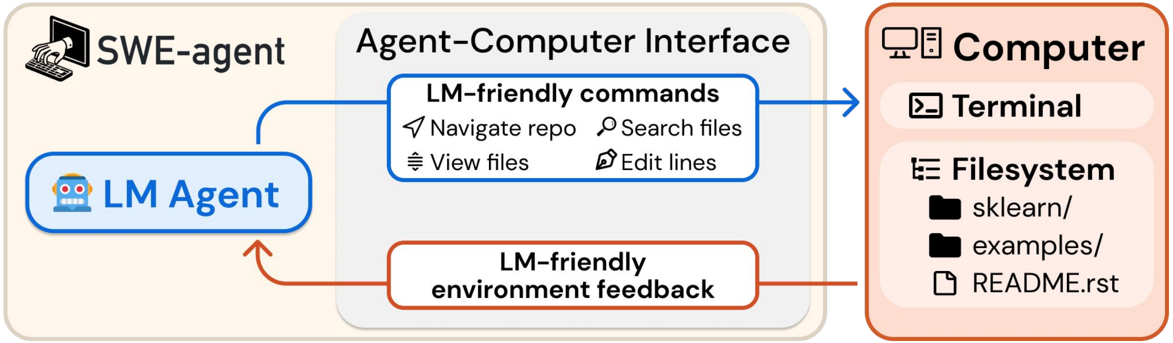
Sumers et al., 2024. Cognitive Architectures for Language Agents

can also tune environment

SWE-agent

= Terminal +
LM-friendly tools

The choice of tools and control flow is also called **scaffold** or ***agent framework***



SWE-agent

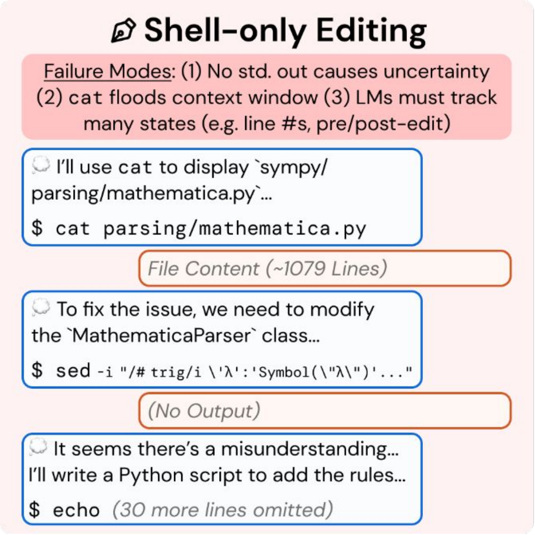
Agent-Computer Interface

LM Agent

LM-friendly commands
Navigate repo
Search files
View files
Edit lines

LM-friendly environment feedback

Computer
Terminal
Filesystem
sklearn/
examples/
README.rst



Shell-only Editing

Failure Modes: (1) No std. out causes uncertainty (2) cat floods context window (3) LMs must track many states (e.g. line #s, pre/post-edit)

I'll use cat to display `sympy/parsing/mathematica.py`...

```
$ cat parsing/mathematica.py
```

File Content (~1079 Lines)

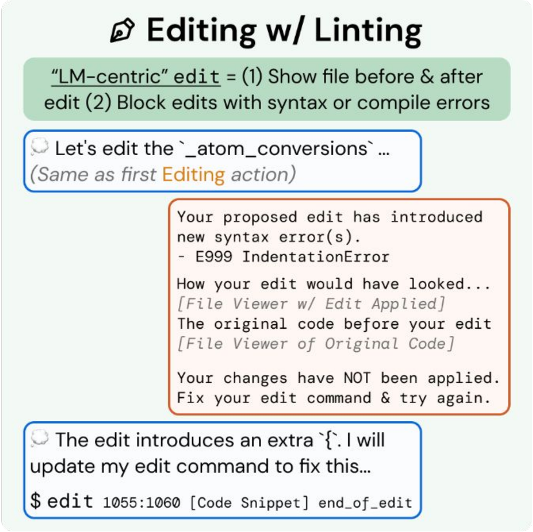
To fix the issue, we need to modify the `MathematicaParser` class...

```
$ sed -i "/# trig/i \'λ':'Symbol(\"λ\")'..."
```

(No Output)

It seems there's a misunderstanding... I'll write a Python script to add the rules...

```
$ echo (30 more lines omitted)
```



Editing w/ Linting

"LM-centric" edit = (1) Show file before & after edit (2) Block edits with syntax or compile errors

Let's edit the `_atom_conversions` ...
(Same as first Editing action)

```
Your proposed edit has introduced
new syntax error(s).
- E999 IndentationError
How your edit would have looked...
[File Viewer w/ Edit Applied]
The original code before your edit
[File Viewer of Original Code]

Your changes have NOT been applied.
Fix your edit command & try again.
```

The edit introduces an extra `{`. I will update my edit command to fix this...

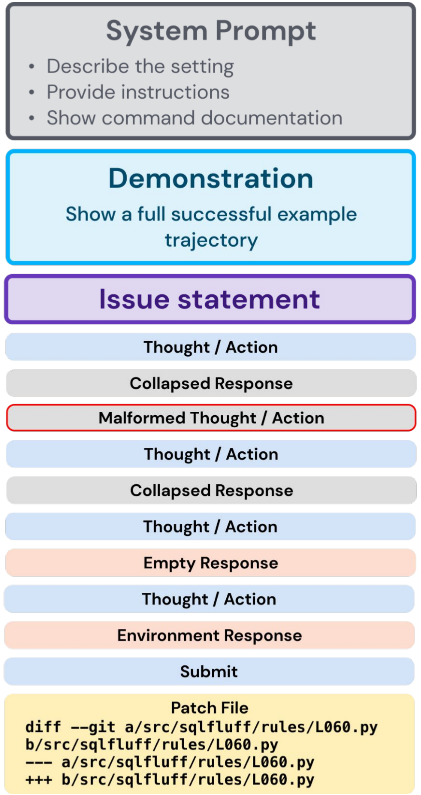
```
$ edit 1055:1060 [Code Snippet] end_of_edit
```

SWE-bench scores **10%** **18%**

What has changed in the last year?

SWE-agent v0.1 (GPT-4)

Score: 18% (April 2024)



System Prompt

- Describe the setting
- Provide instructions
- Show command documentation

Demonstration

Show a full successful example trajectory

Issue statement

Thought / Action

Collapsed Response

Malformed Thought / Action

Thought / Action

Collapsed Response

Thought / Action

Empty Response

Thought / Action

Environment Response

Submit

Patch File

```
diff --git a/src/sqlfluff/rules/L060.py
b/src/sqlfluff/rules/L060.py
--- a/src/sqlfluff/rules/L060.py
+++ b/src/sqlfluff/rules/L060.py
```

SWE-agent v1.0 (Claude 3.7 Sonnet)

Score: 33% (February 2025)

System Prompt

- Explain mission, strategy & give tips
- **Can be very short** with Claude 3.5+
- LLMs has standardized interface for tools

Demonstrations (providing examples trajectories)

- **Not needed** with Claude 3.5+
- Can lock agent into specific patterns (bad if you have high task variance)
- LM does ReAct out of the box

History processing

- Previously: kept only last 5 observations
- **New models can deal better with long context**
- **Multi-agents** as fancy way of managing context (but worth the added complexity & cost?)

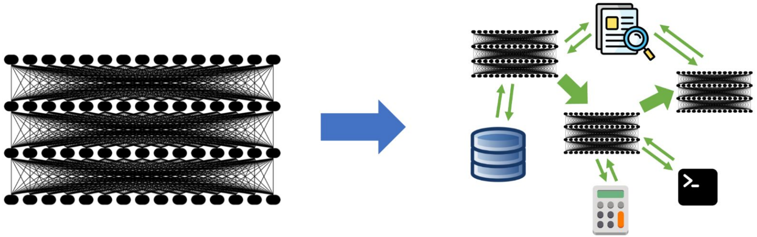
Quick Recap

The LLM Agent interacts with an **environment**

text inputs (env. feedback, tool results) ↔ text outputs (actions, tool calls)

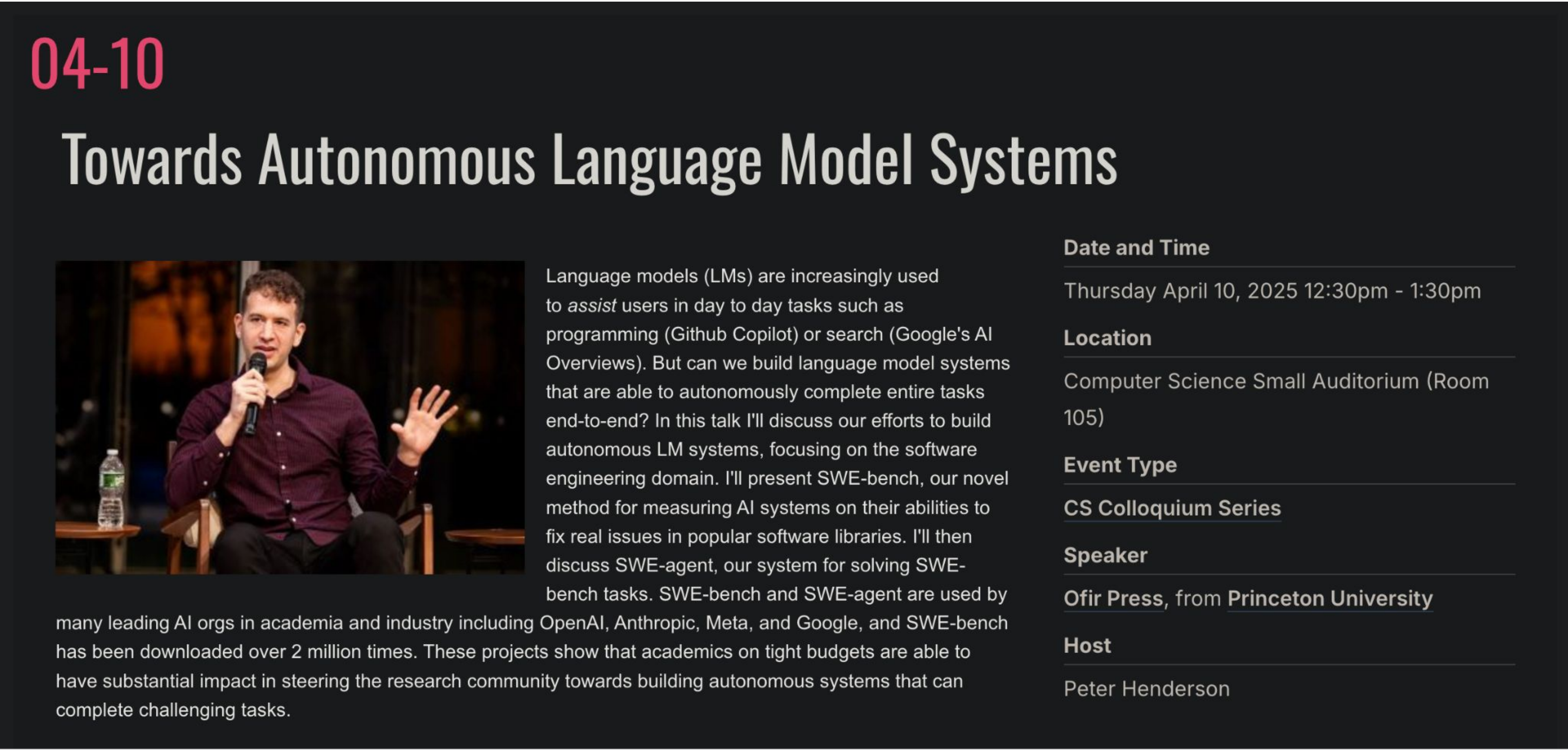
Agent **framework/scaffold** handles control flow
(e.g. retries, self-verification, context window, budget constraints)

AI systems = agents and other compound systems consisting of many LLM calls



Increasingly many new AI results are from compound systems.

bair.berkeley.edu/blog/2024/02/18/compound-ai-systems



04-10

Towards Autonomous Language Model Systems

Language models (LMs) are increasingly used to *assist* users in day to day tasks such as programming (Github Copilot) or search (Google's AI Overviews). But can we build language model systems that are able to autonomously complete entire tasks end-to-end? In this talk I'll discuss our efforts to build autonomous LM systems, focusing on the software engineering domain. I'll present SWE-bench, our novel method for measuring AI systems on their abilities to fix real issues in popular software libraries. I'll then discuss SWE-agent, our system for solving SWE-bench tasks. SWE-bench and SWE-agent are used by many leading AI orgs in academia and industry including OpenAI, Anthropic, Meta, and Google, and SWE-bench has been downloaded over 2 million times. These projects show that academics on tight budgets are able to have substantial impact in steering the research community towards building autonomous systems that can complete challenging tasks.

Date and Time

Thursday April 10, 2025 12:30pm - 1:30pm

Location

Computer Science Small Auditorium (Room 105)

Event Type

CS Colloquium Series

Speaker

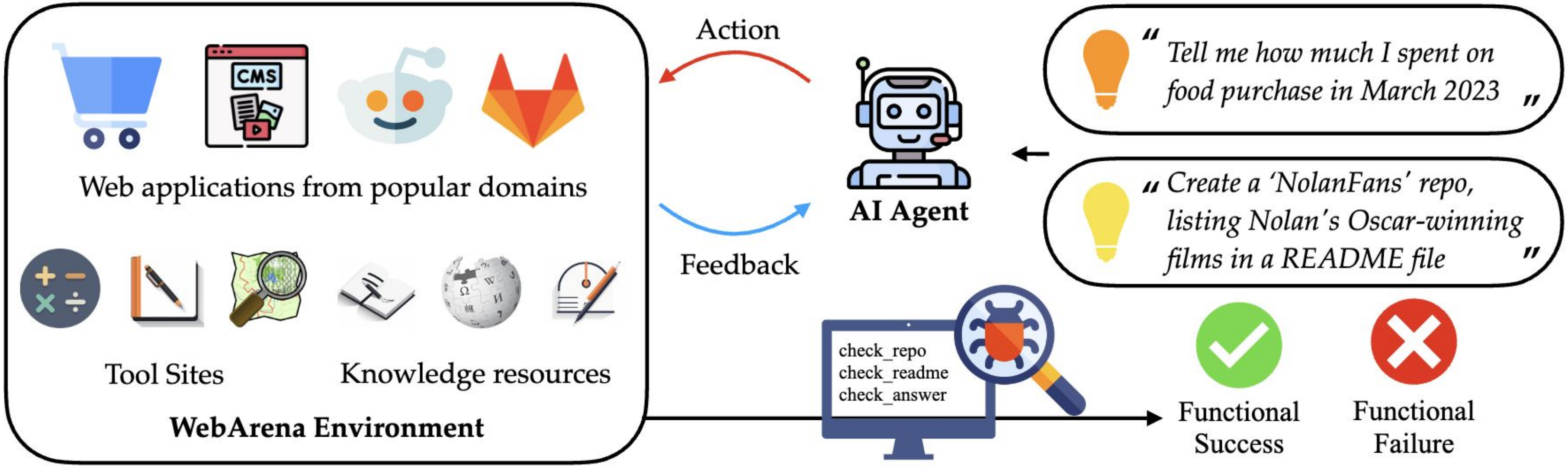
Ofir Press, from **Princeton University**

Host

Peter Henderson

LLM agents: other applications and developments

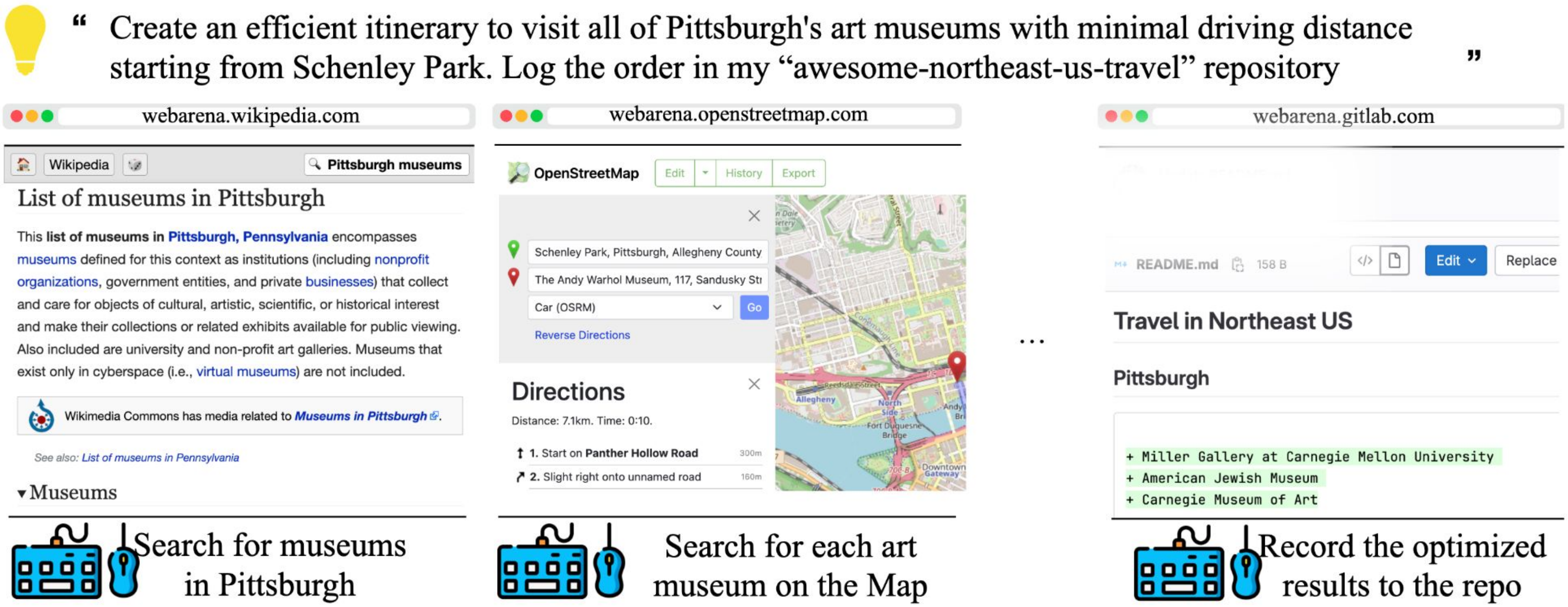
Other benchmarks: **WebArena**


Web applications from popular domains
Tool Sites
Knowledge resources
WebArena Environment
Action
Feedback
AI Agent
"Tell me how much I spent on food purchase in March 2023"
"Create a 'NolanFans' repo, listing Nolan's Oscar-winning films in a README file"
check_repo
check_readme
check_answer
Functional Success
Functional Failure

webarena.dev

Zhou et al., 2023, WebArena: A Realistic Web Environment for Building Autonomous Agents

Other benchmarks: **WebArena**



“ Create an efficient itinerary to visit all of Pittsburgh's art museums with minimal driving distance starting from Schenley Park. Log the order in my “awesome-northeast-us-travel” repository ”

List of museums in Pittsburgh

This **list of museums in Pittsburgh, Pennsylvania** encompasses museums defined for this context as institutions (including nonprofit organizations, government entities, and private businesses) that collect and care for objects of cultural, artistic, scientific, or historical interest and make their collections or related exhibits available for public viewing. Also included are university and non-profit art galleries. Museums that exist only in cyberspace (i.e., virtual museums) are not included.

Wikimedia Commons has media related to *Museums in Pittsburgh*.

See also: List of museums in Pennsylvania

▾ Museums

Search for museums in Pittsburgh

Schenley Park, Pittsburgh, Allegheny County

The Andy Warhol Museum, 117, Sandusky Str

Car (OSRM)

Directions

Distance: 7.1km. Time: 0:10.

1. Start on **Panther Hollow Road** 300m
2. Slight right onto unnamed road 160m

Search for each art museum on the Map

…

Travel in Northeast US

Pittsburgh

```
+ Miller Gallery at Carnegie Mellon University
+ American Jewish Museum
+ Carnegie Museum of Art
```

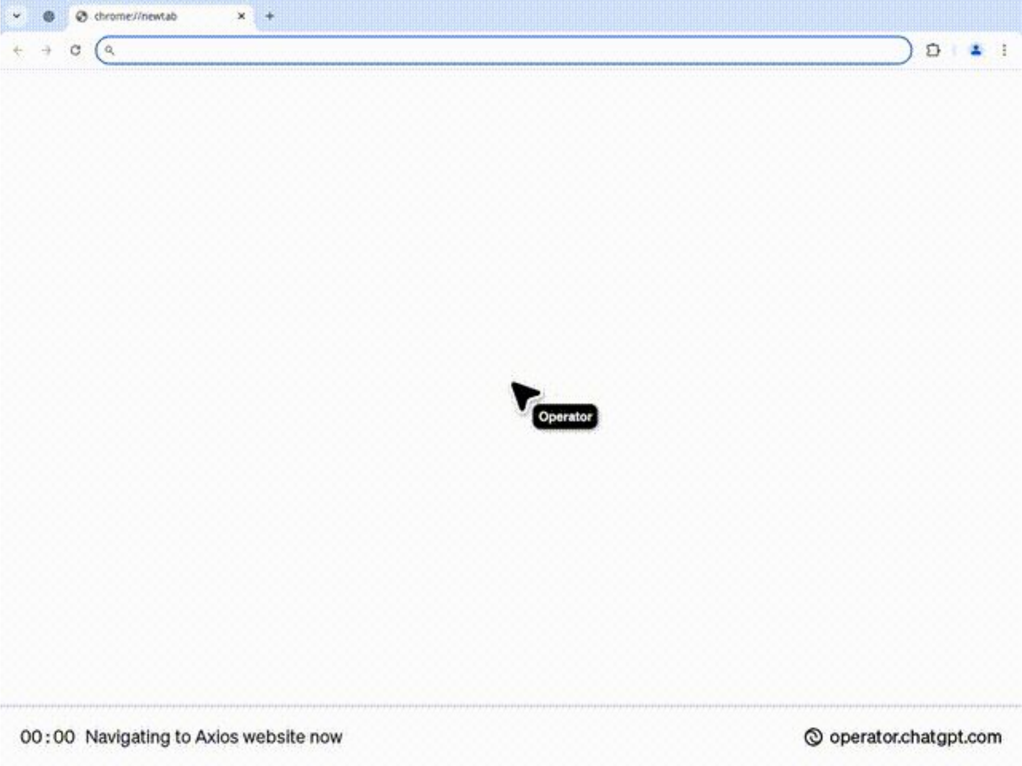
Record the optimized results to the repo

webarena.dev

Zhou et al., 2023. WebArena: A Realistic Web Environment for Building Autonomous Agents

Product: **OpenAI Operator**

Similar: Claude Computer Use

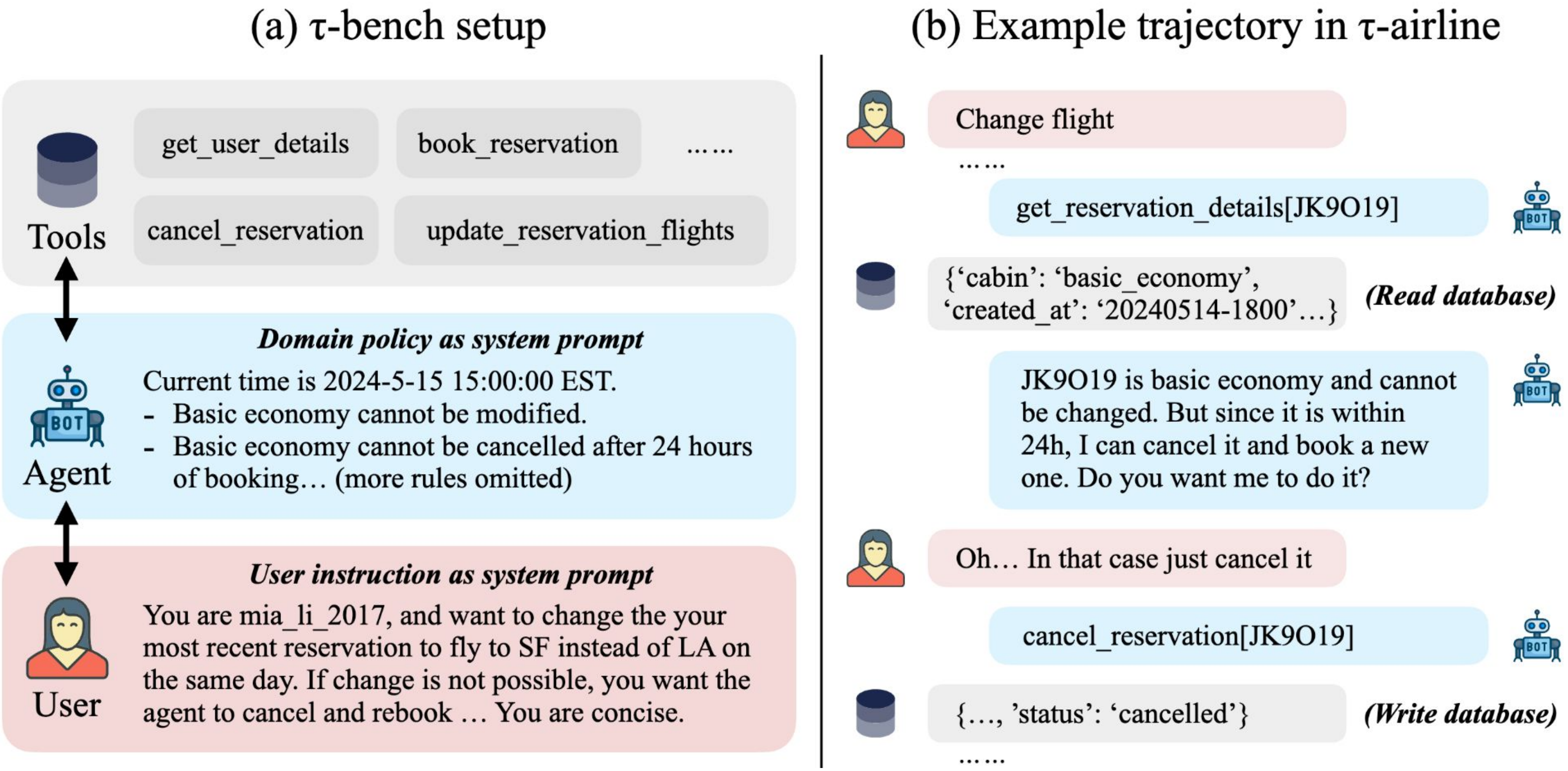


00:00 Navigating to Axios website now

operator.chatgpt.com

every.to/chain-of-thought/we-tried-openai-s-new-agent-here-s-what-we-found

Other benchmarks: **τ-bench**



Domain policy as system prompt

Current time is 2024-5-15 15:00:00 EST.
- Basic economy cannot be modified.
- Basic economy cannot be cancelled after 24 hours of booking… (more rules omitted)

User instruction as system prompt

You are mia_li_2017, and want to change the your most recent reservation to fly to SF instead of LA on the same day. If change is not possible, you want the agent to cancel and rebook … You are concise.

Change flight

……

get_reservation_details[JK9O19]

{'cabin': 'basic_economy', 'created_at': '20240514-1800'…}

JK9O19 is basic economy and cannot be changed. But since it is within 24h, I can cancel it and book a new one. Do you want me to do it?

Oh… In that case just cancel it

cancel_reservation[JK9O19]

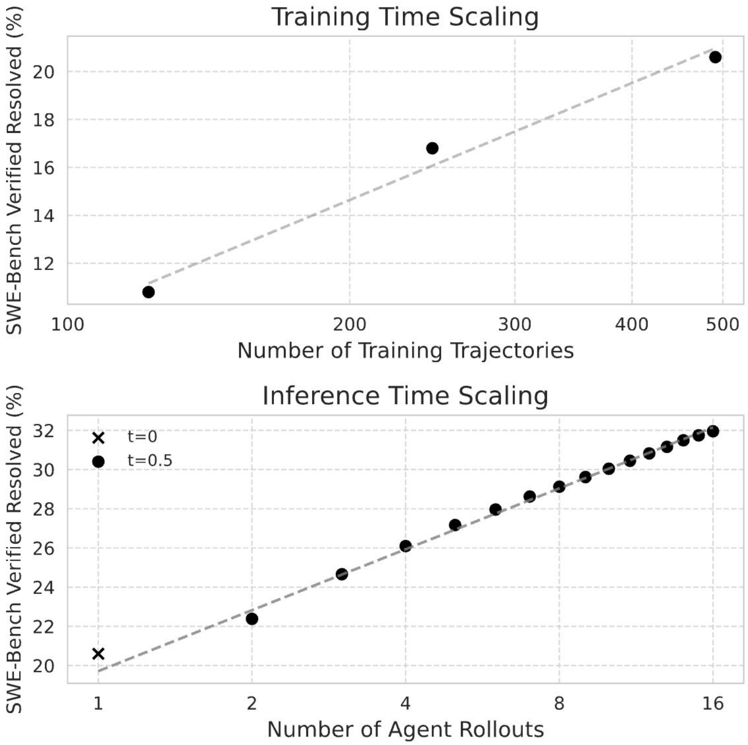
{…, 'status': 'cancelled'}

……

Yao et al., τ-bench, 2024: A Benchmark for Tool-Agent-User Interaction in Real-World Domains

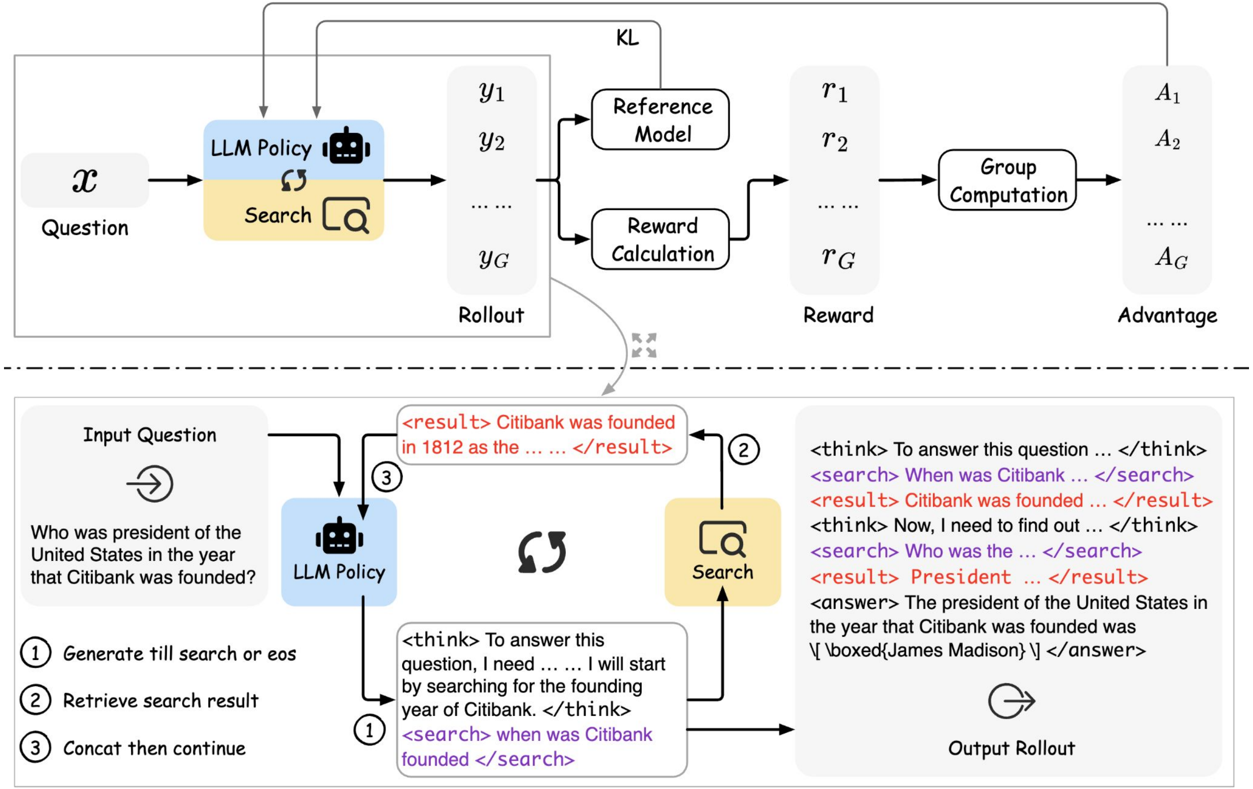
LLM agent training: **SWE-gym**

- Collect more SWE-bench issues for fine-tuning LLMs
- Use data to fine-tune models both as agent and as verifier
- Best system with 32B LLM achieves 32% on SWE-bench Verified



Pan et al., 2024. Training Software Engineering Agents and Verifiers with SWE-Gym

LLM agent training: **ReSearch**



Who was president of the United States in the year that Citibank was founded?

<result> Citibank was founded in 1812 as the … … </result>

<think> To answer this question, I need … … I will start by searching for the founding year of Citibank. </think> <search> when was Citibank founded </search>

① Generate till search or eos

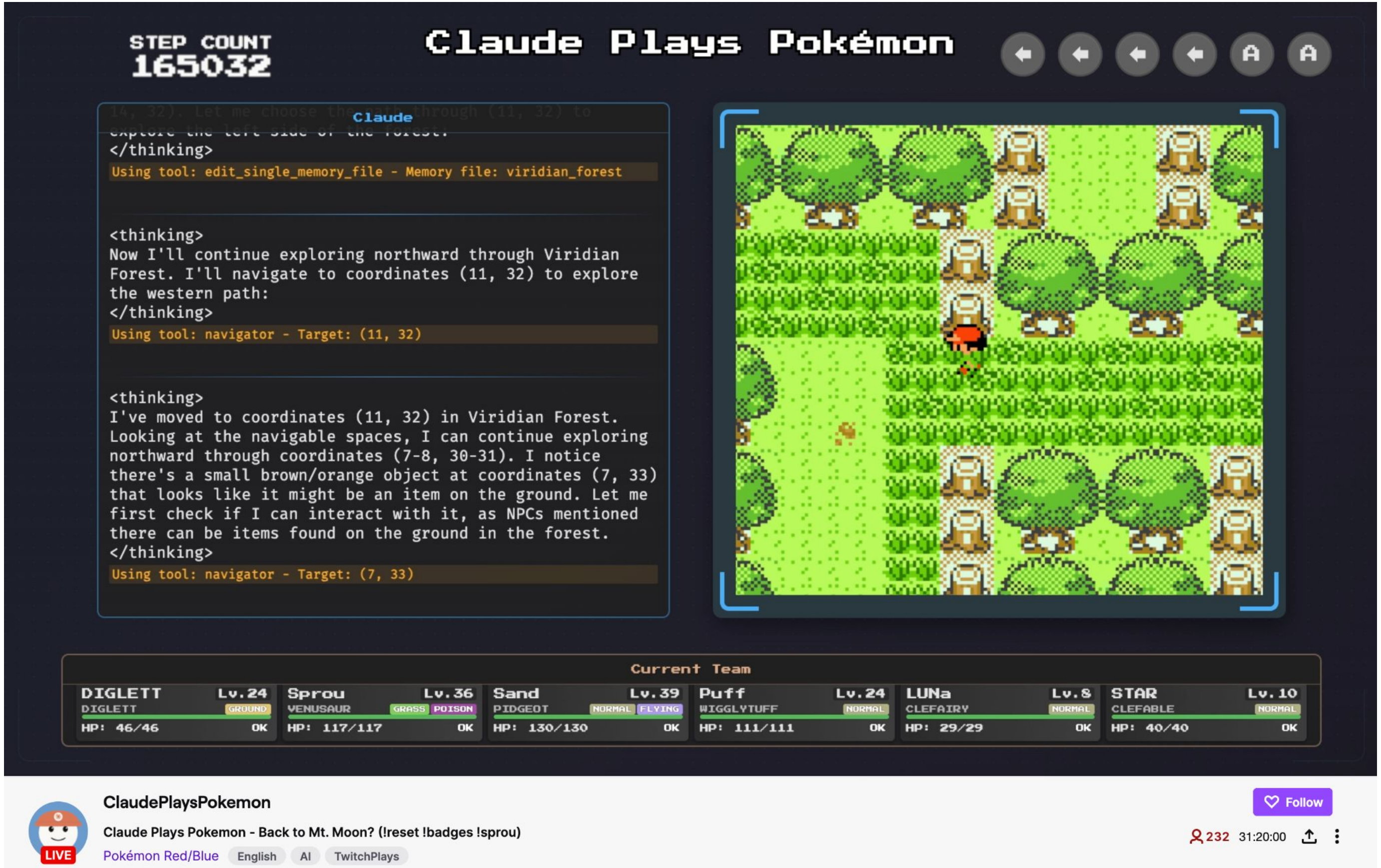
② Retrieve search result

③ Concat then continue

<think> To answer this question … </think>
<search> When was Citibank … </search>
<result> Citibank was founded … </result>
<think> Now, I need to find out … </think>
<search> Who was the … </search>
<result> President … </result>
<answer> The president of the United States in the year that Citibank was founded was \[\boxed{James Madison} \] </answer>

Chen et al., 2025. ReSearch: Learning to Reason with Search for LLMs via Reinforcement Learning

Research Demo: **Claude Plays Pokemon**



STEP COUNT
165032

Claude Plays Pokémon

```
14, 32). Let me choose the path through (11, 32) to
explore the left side of the forest:
</thinking>
Using tool: edit_single_memory_file - Memory file: viridian_forest

<thinking>
Now I'll continue exploring northward through Viridian
Forest. I'll navigate to coordinates (11, 32) to explore
the western path:
</thinking>
Using tool: navigator - Target: (11, 32)

<thinking>
I've moved to coordinates (11, 32) in Viridian Forest.
Looking at the navigable spaces, I can continue exploring
northward through coordinates (7-8, 30-31). I notice
there's a small brown/orange object at coordinates (7, 33)
that looks like it might be an item on the ground. Let me
first check if I can interact with it, as NPCs mentioned
there can be items found on the ground in the forest.
</thinking>
Using tool: navigator - Target: (7, 33)
```

Current Team

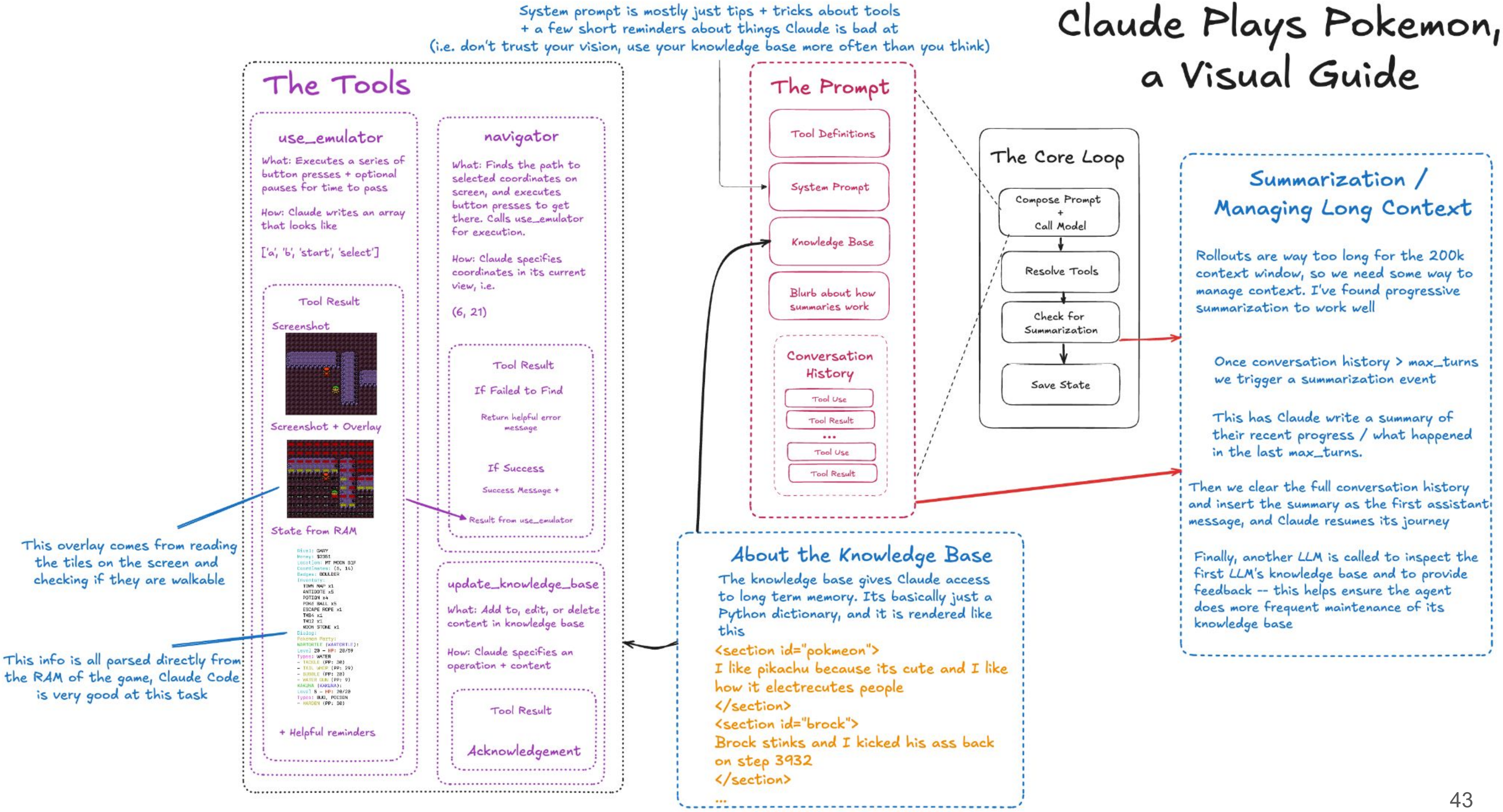
DIGLETT Lv.24	Sprou Lv.36	Sand Lv.39	Puff Lv.24	LUNa Lv.8	STAR Lv.10
DIGLETT GROUND	VENUSAUR GRASS POISON	PIDGEOT NORMAL FLYING	WIGGLYTUFF NORMAL	CLEFAIRY NORMAL	CLEFABLE NORMAL
HP: 46/46 OK	HP: 117/117 OK	HP: 130/130 OK	HP: 111/111 OK	HP: 29/29 OK	HP: 40/40 OK

ClaudePlaysPokemon

Claude Plays Pokemon - Back to Mt. Moon? (!reset !badges !sprou)

Pokémon Red/Blue English AI TwitchPlays

Claude Plays Pokemon, a Visual Guide



System prompt is mostly just tips + tricks about tools
+ a few short reminders about things Claude is bad at
(i.e. don't trust your vision, use your knowledge base more often than you think)

The Tools

use_emulator

What: Executes a series of button presses + optional pauses for time to pass

How: Claude writes an array that looks like

['a', 'b', 'start', 'select']

Tool Result

Screenshot

Screenshot + Overlay

State from RAM

```
Rival: GARY
Money: $2351
Location: MT MOON B1F
Coordinates: (5, 14)
Badges: BOULDER
Inventory:
  TOWN MAP x1
  ANTIDOTE x5
  POTION x4
  POKé BALL x5
  ESCAPE ROPE x1
  TM34 x1
  TM12 x1
  MOON STONE x1
Dialog:
Pokemon Party:
WARTORTLE (WARTORTLE):
Level 20 - HP: 28/59
Types: WATER
- TACKLE (PP: 30)
- TAIL WHIP (PP: 29)
- BUBBLE (PP: 28)
- WATER GUN (PP: 9)
KAKUNA (KAKUNA):
Level 5 - HP: 20/20
Types: BUG, POISON
- HARDEN (PP: 30)
```

+ Helpful reminders

This overlay comes from reading the tiles on the screen and checking if they are walkable

This info is all parsed directly from the RAM of the game, Claude Code is very good at this task

navigator

What: Finds the path to selected coordinates on screen, and executes button presses to get there. Calls use_emulator for execution.

How: Claude specifies coordinates in its current view, i.e.

(6, 21)

Tool Result

If Failed to Find

Return helpful error message

If Success

Success Message +

Result from use_emulator

update_knowledge_base

What: Add to, edit, or delete content in knowledge base

How: Claude specifies an operation + content

Tool Result

Acknowledgement

The Prompt

- Tool Definitions
- System Prompt
- Knowledge Base
- Blurb about how summaries work
- Conversation History
  - Tool Use
  - Tool Result
  - ...
  - Tool Use
  - Tool Result

The Core Loop

- Compose Prompt + Call Model
- Resolve Tools
- Check for Summarization
- Save State

Summarization / Managing Long Context

Rollouts are way too long for the 200k context window, so we need some way to manage context. I've found progressive summarization to work well

Once conversation history > max_turns we trigger a summarization event

This has Claude write a summary of their recent progress / what happened in the last max_turns.

Then we clear the full conversation history and insert the summary as the first assistant message, and Claude resumes its journey

Finally, another LLM is called to inspect the first LLM's knowledge base and to provide feedback -- this helps ensure the agent does more frequent maintenance of its knowledge base

About the Knowledge Base

The knowledge base gives Claude access to long term memory. Its basically just a Python dictionary, and it is rendered like this

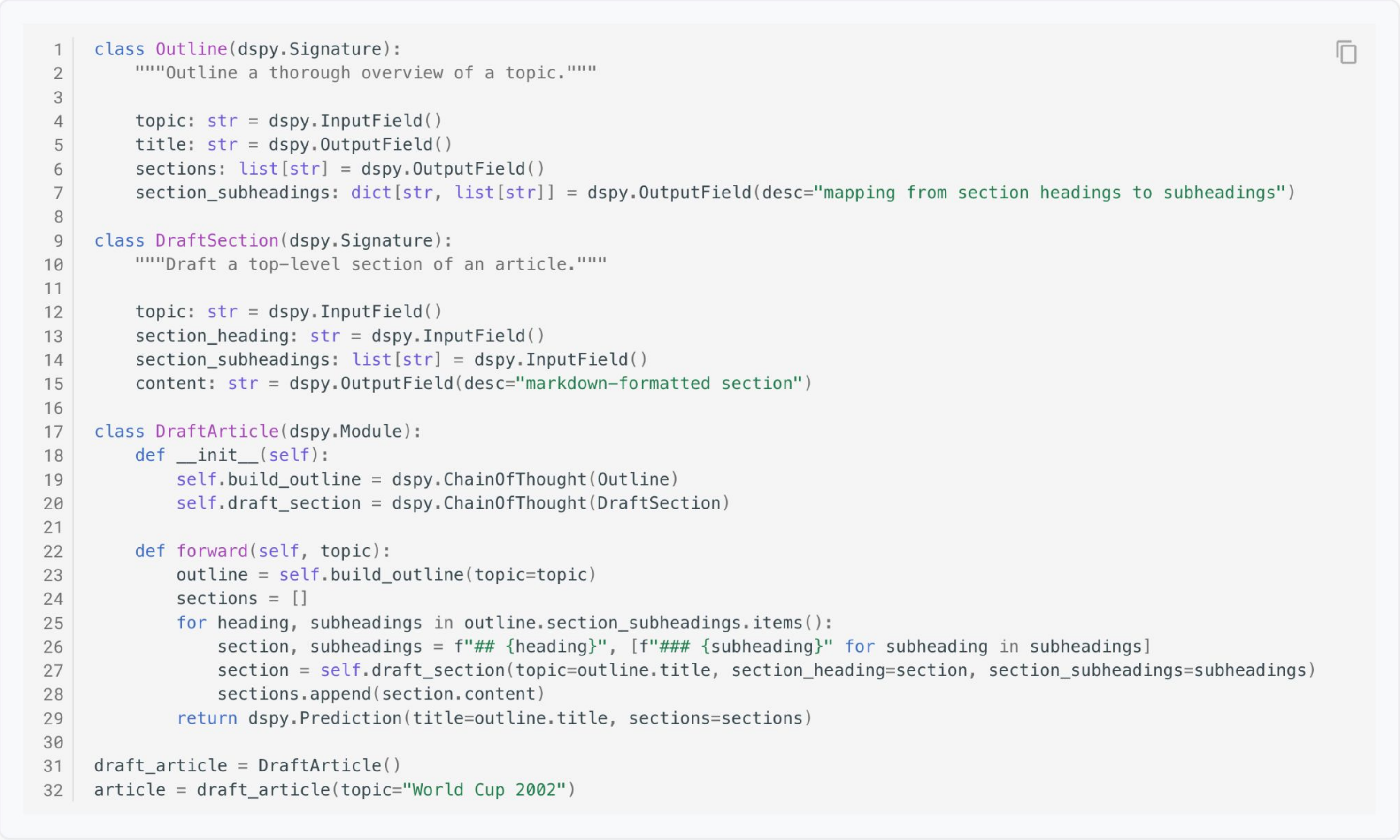
```
<section id="pokmeon">
I like pikachu because its cute and I like how it electrecutes people
</section>
<section id="brock">
Brock stinks and I kicked his ass back on step 3932
</section>
...
```

https://www.twitch.tv/claudeplayspokemon

Frameworks for building LM agents / AI systems

- DsPy
- Model Context Protocol (MCP)
- LangChain
- LlamaIndex
- SWE-agent
- OpenHands

DsPy (dspy.ai)



```python
class Outline(dspy.Signature):
    """Outline a thorough overview of a topic."""

    topic: str = dspy.InputField()
    title: str = dspy.OutputField()
    sections: list[str] = dspy.OutputField()
    section_subheadings: dict[str, list[str]] = dspy.OutputField(desc="mapping from section headings to subheadings")

class DraftSection(dspy.Signature):
    """Draft a top-level section of an article."""

    topic: str = dspy.InputField()
    section_heading: str = dspy.InputField()
    section_subheadings: list[str] = dspy.InputField()
    content: str = dspy.OutputField(desc="markdown-formatted section")

class DraftArticle(dspy.Module):
    def __init__(self):
        self.build_outline = dspy.ChainOfThought(Outline)
        self.draft_section = dspy.ChainOfThought(DraftSection)

    def forward(self, topic):
        outline = self.build_outline(topic=topic)
        sections = []
        for heading, subheadings in outline.section_subheadings.items():
            section, subheadings = f"## {heading}", [f"### {subheading}" for subheading in subheadings]
            section = self.draft_section(topic=outline.title, section_heading=section, section_subheadings=subheadings)
            sections.append(section.content)
        return dspy.Prediction(title=outline.title, sections=sections)

draft_article = DraftArticle()
article = draft_article(topic="World Cup 2002")
```