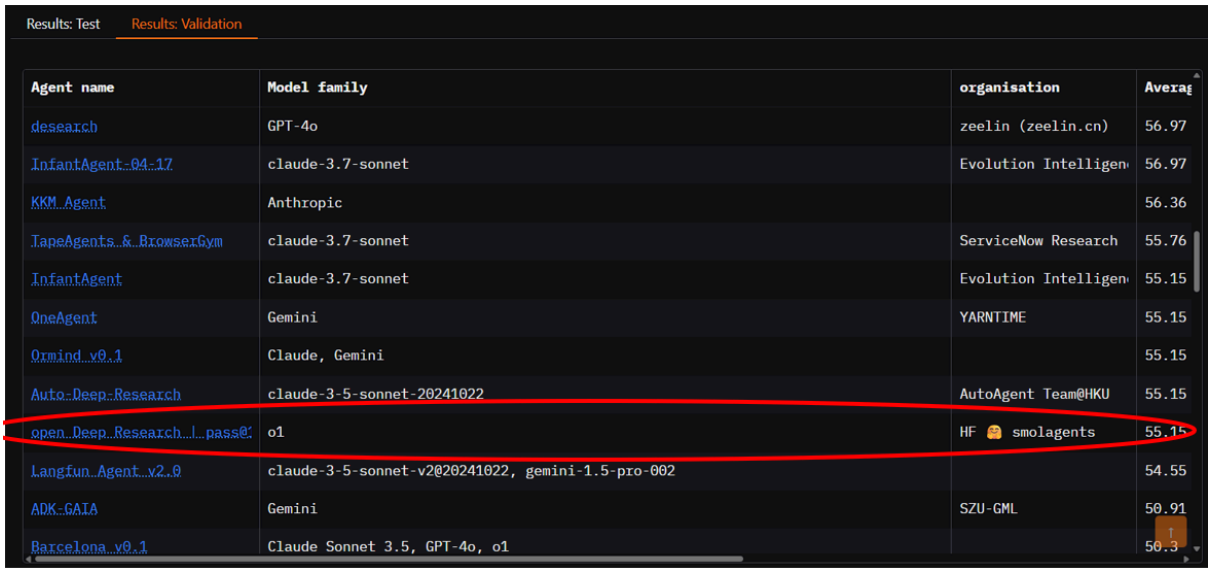
What are Smol Agents?

- It's a lightweight, minimalistic library from Hugging Face.
- In order to solve a problem Agents typically return the next action (including tool calls) in Text/JSON format. However, writing actions directly in Code produces more efficient agents with lesser steps on average.
  Core idea described in: Executable Code Actions Elicit Better LLM Agents
- Their argument is that writing actions (tool calls) in code allows a more structured way of dealing with the problem and allows coding operations like for-loops that reduce the number of steps.
- Motivation to reduce the number of steps is pretty obvious since it reduces: latency, number of tokens, and the risk of failure.

Currently the only open source 'deep research' implementation that does well on the GAIA benchmark (leaderboard)



Results: Test | Results: Validation

Agent name	Model family	organisation	Averag
desearch	GPT-4o	zeelin (zeelin.cn)	56.97
InfantAgent-04-17	claude-3.7-sonnet	Evolution Intelligen	56.97
KKM Agent	Anthropic		56.36
TapeAgents & BrowserGym	claude-3.7-sonnet	ServiceNow Research	55.76
InfantAgent	claude-3.7-sonnet	Evolution Intelligen	55.15
OneAgent	Gemini	YARNTIME	55.15
Ormind v0.1	Claude, Gemini		55.15
Auto Deep Research	claude-3-5-sonnet-20241022	AutoAgent Team@HKU	55.15
open Deep Research	pass@	o1	HF 🤗 smolagents	55.15
Langfun Agent v2.0	claude-3-5-sonnet-v2@20241022, gemini-1.5-pro-002		54.55
ADK-GAIA	Gemini	SZU-GML	50.91
Barcelona v0.1	Claude Sonnet 3.5, GPT-4o, o1		50.3

I ran an extremely naive version of smol agents and improved my own naive submission by around 15% point on average on the GAIA. Still have some way to go.
My naive submission with no library:

Agent name	Model family	organisation	Average score (%)
AutoGPT4	AutoGPT + GPT4	AutoGPT	4.85
fhswfgaia	llama3.8:8b	fhswfgaia	3.03
gpt_3.5	gpt_3.5	test	2.42

Submission with smolagents (without any tools - even lacks file reading for xlsx, audio files etc.):

Agent name	Model family	organisation	Average score (%)
HuggingFace Agents + Llama3-70B	Meta-Llama-3-70B-Instruct	Hugging Face	16.97
fhswf-gaia-smolagents	qwen-coder	fhswf-gaia	16.97

What differentiates it from other agentic libraries:

- Smol Agents use a code-first approach for tool-calling with the `CodeAgent` class. Unlike traditional tool-calling agents that operate with JSON or text-based action definitions, Smol Agents writes and executes actions as Python code snippets.

On comparisons they report: approximately 30% fewer steps.

Tools:

Smol Agents comes with several pre-built tools accessible through the **`add_base_tools=True`** parameter when initializing an agent:

1. **DuckDuckGoSearchTool**: Performs web searches using the DuckDuckGo search engine
2. **VisitWebpageTool**: Retrieves and processes web page content
3. **PythonInterpreterTool**: Executes Python code in a controlled environment
4. **FinalAnswerTool**: Provides the agent's final response to a query

Example code for Smol Agents:

```
 # Create an agent with tools
agent = CodeAgent(
    tools=[
        DuckDuckGoSearchTool(),
        VisitWebpageTool(),
        FinalAnswerTool()
    ],
    model=model,
    additional_authorized_imports=["wikipedia", "requests", "json", "re",
"datetime", "os"]
)

 # Run the agent on a task
result = agent.run("Research the population trends of major European cities over
the last decade.")
```

Model Agnostic:

- Local models via Transformers or Ollama or Proprietary models (OpenAI, Anthropic, etc.) via **LiteLLM integration**
- Hugging Face Hub models via Inference API

```
from smolagents import CodeAgent, LiteLLMModel, DuckDuckGoSearchTool,
VisitWebpageTool, FinalAnswerTool

 # Initialize a model using Ollama
model = LiteLLMModel(
    model_id="ollama_chat/qwen2.5-coder:32b",  # Format:
"ollama_chat/[model-name]"
    api_base="http://localhost:11434",         # Default Ollama API endpoint
    api_key="ollama",                          # Placeholder, not actually
required
    num_ctx=30000                              # Expand context window if needed
)
```

Will continue to add more to this document about:

- Improved submissions on GAIA + Insights
- Experiments with various open source LLMs (currently using the QWEN coder family of models)
- Evaluation and Telemetry (tracing tokes, actions, performance, debugging)