mathCrow: Agentic AI for Math Tasks

Ahmed Ali Abbasi

What Is An Agent?

- Agent = “LLM” + “Memory” + “Tools” + “Planning”.
- Autonomous agents are designed to solve complex tasks with minimal human interaction.
- LLM functions as the brain of the agent.
- This project implements agents with two components: “LLM” and “Tools”.

What is an Agent?

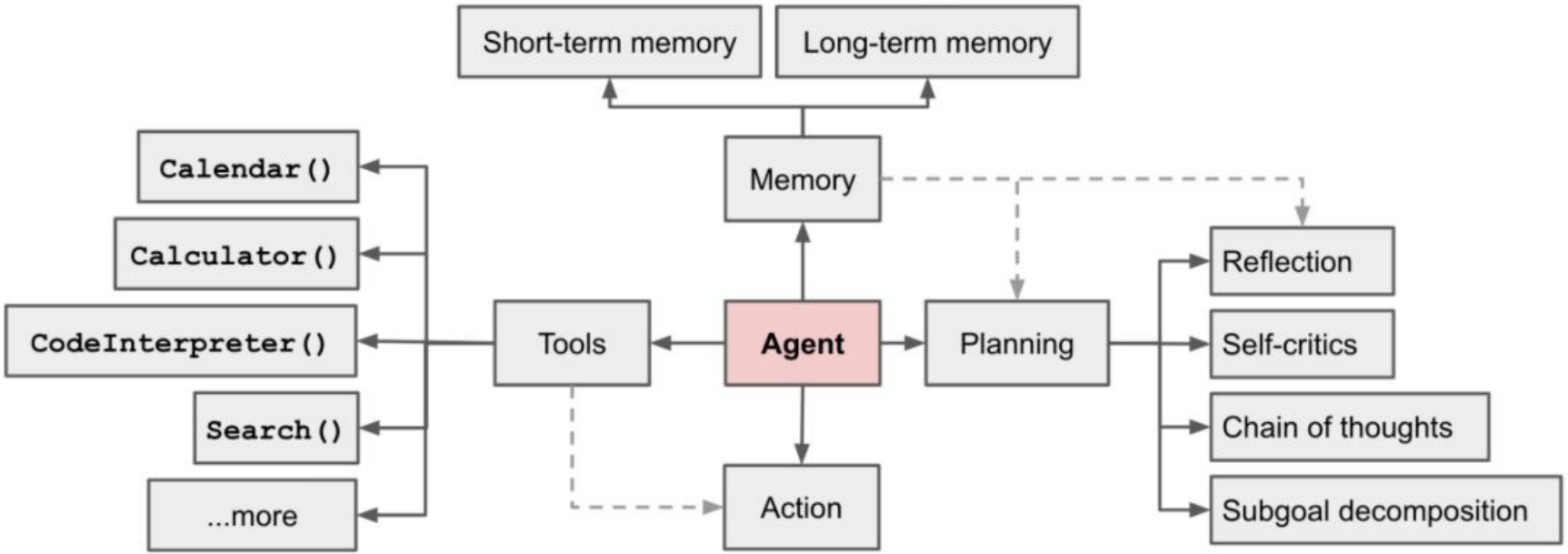
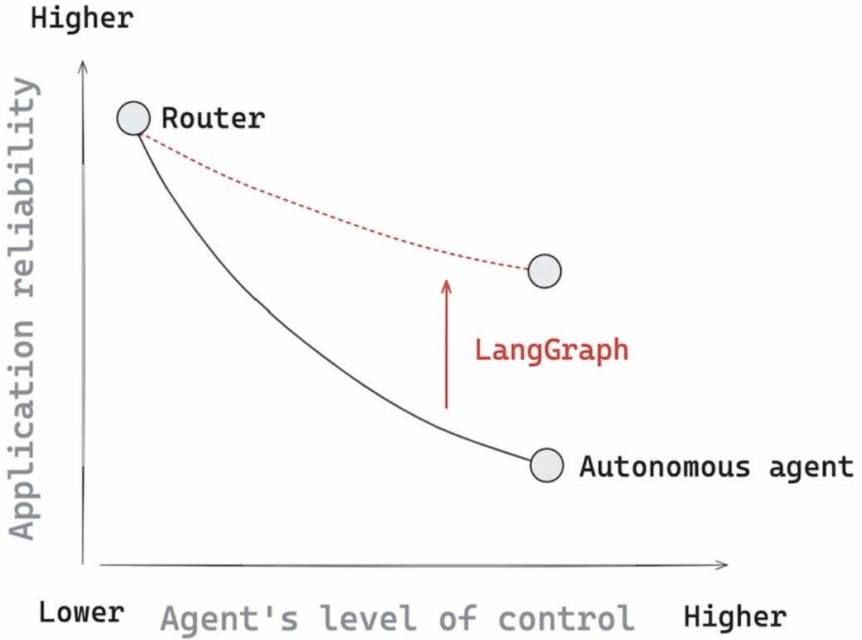


Fig. 1. Overview of a LLM-powered autonomous agent system.

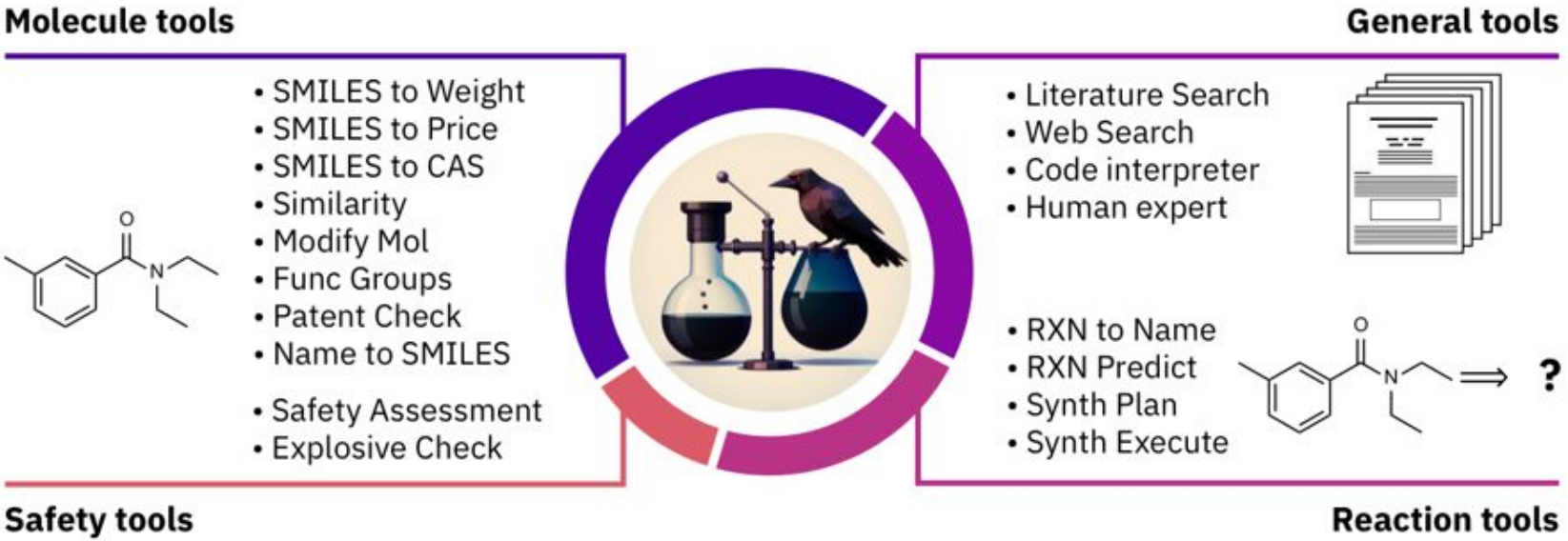
Figure source: https://lilianweng.github.io/posts/2023-06-23-agent/

What Is An Agent?

- Routers have minimum autonomy - implemented by “if-else” conditions.
- Agents leave all decision making and planning to the brain (LLM).
- Langraph is a python package used to implement agentic frameworks.



Recent Academic/Research Use of Agents: ChemCrow

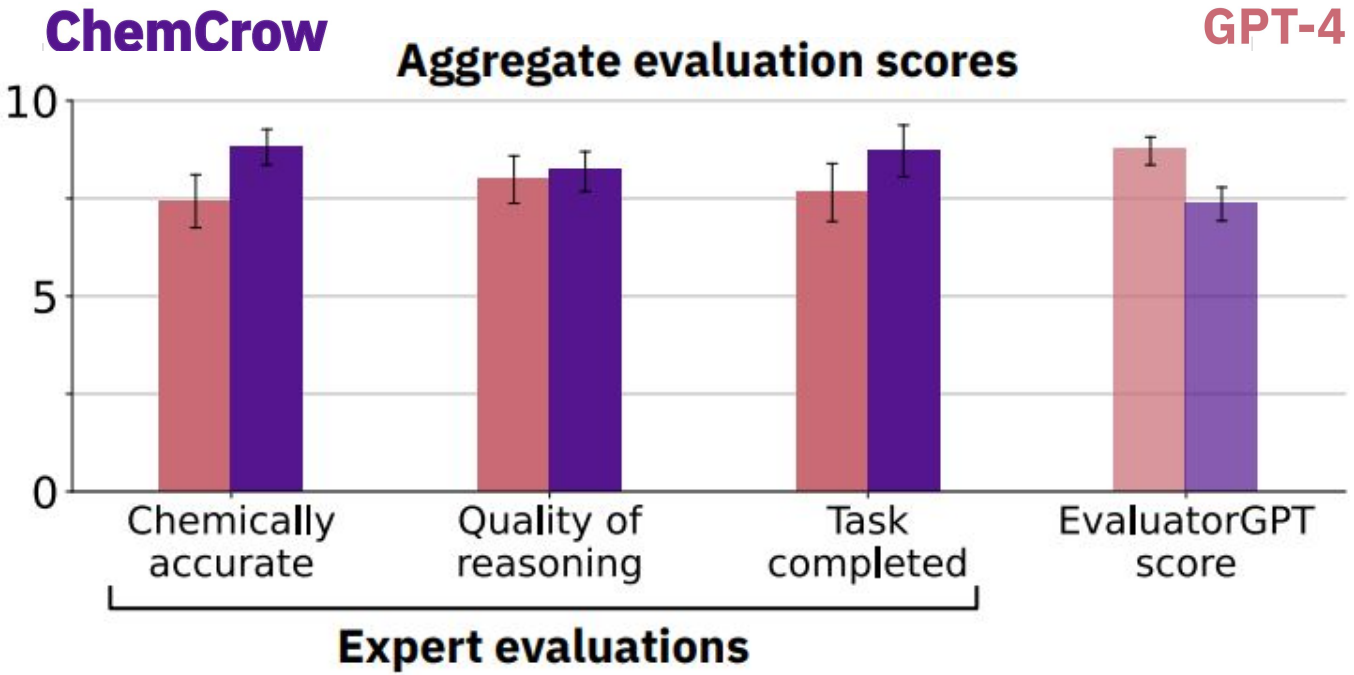


ChemCrow [https://www.nature.com/articles/s42256-024-00832-8] designs an agent specialized to chemistry tasks. Figure shows example of 14 tools for 4 domains {Molecule, General, Safety, Reaction.}

Why Use Expert-Designed Tools instead of chatGPT?

- Specific/Expert-designed tools are more accurate than chatGPT, especially for complex tasks.
- Expert designed tools do not “hallucinate”.
- Tool use by LLMs provides greater autonomy than human-tool-use.
- Tools can be integrated with local LLMs, preserving privacy.

ChemCrow vs ChatGPT4



- Compared on 3 metrics {“accuracy, reasoning, completion”}.
- Human experts ranks ChemCrow higher.
- Evaluator LLM ranks GPT-4 higher.

We Propose MathCrow

- We propose a math agent with two tools.
- Tool 1: Retrieval Augmented Generation (Agentic-RAG) for finely-reading research papers.
- Tool 2: Linear Assignment Problem Solver.
- Agentic use of tools:
  i) improves the generated responses
  ii) eliminates hallucinations
  iii) has greater autonomy than stand-alone tool use.

Tool 1: Retrieval Augmented Generation (RAG)

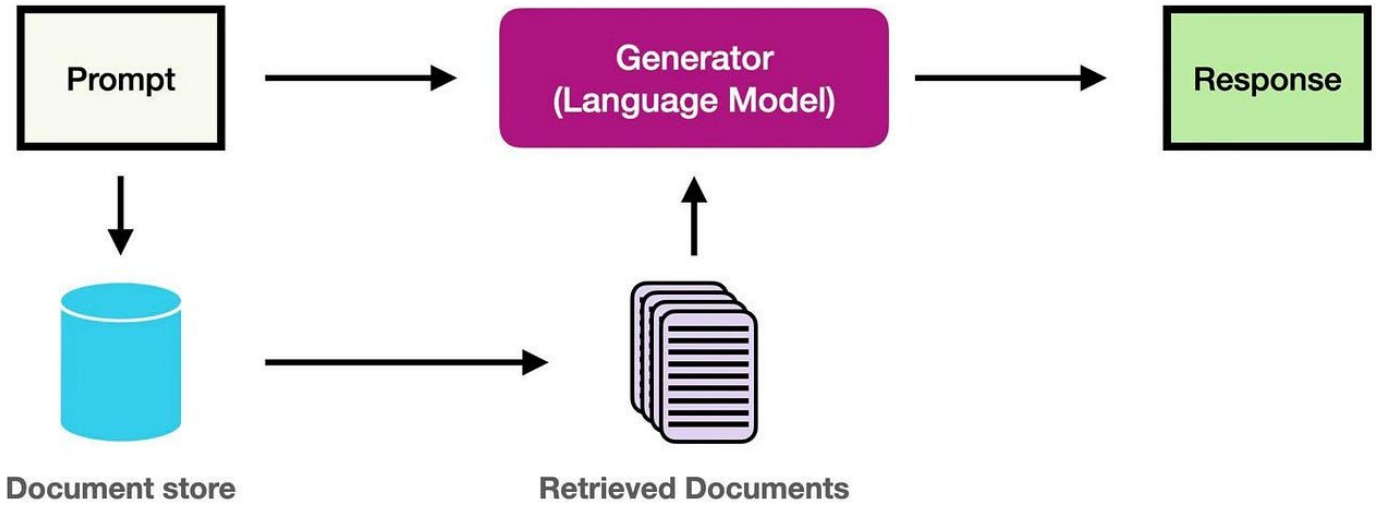
- Key idea is to **retrieve** relevant data (context) before **generating** response.
- Typical data examples are books, research papers or other documents stored in a vector database.
- Advantages of RAG:

  i) analyze documents larger than the context window.

  ii) avoid updating/fine-tuning LLM to incorporate new information.

  iii) fewer hallucinations.

RAG Overview



The retrieved documents from the store are the context included in the LLM query off which the answer is based.

RAG Overview

- Which documents to retrieve from the database?
- Database contains embeddings (dense vectors) of words in all document “chunks”.
- The embeddings of the input query are compared with chunk embeddings.
- Top-k, where k = {3,4,…}, similar chunks are retrieved.
- Agent (LLM / brain) decides if retrieved chunks are relevant.
- Relevant document chunks form the context.

Embedding Models

- Embedding models embed words into vectors.
- Several embedding models including paid versions by Open-AI embeddings and Google Gemini Embeddings.
- Free/Open source embedding models ranked at https://huggingface.co/spaces/mteb/leaderboard
- We use open-source embeddings to have a zero-cost agentic system.

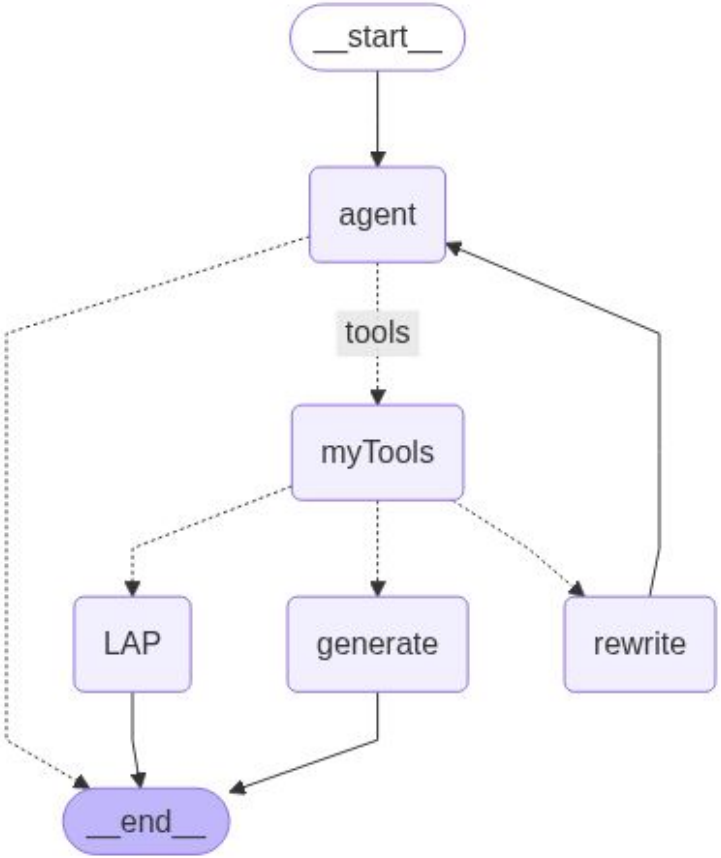
Tool 2: Linear Assignment Problem (LAP) solver

- Wrap“scipy.optimize.linear_sum_assignment” as a “tool” so the agent (LLM) can call it.
- Like ChemCrow, this is an example of a specialized tool for one specific problem.

```python
from langchain.tools import Tool
from scipy.optimize import linear_sum_assignment

class costMatrixClass(BaseModel):
  M: List[List[float]] = Field(description="This is the cost matrix of float numbers."

def solveLAP(M):
  A = np.array(M)
  row_ind, col_ind = linear_sum_assignment(A)
  return {"messages": [row_ind,col_ind]}
  #return M_np

# Create a Tool instance
squareTool = Tool.from_function(
    func=solveLAP,
    name="squareMatrix",
    description="Solves linear assignment problem (LAP).",
    args_schema=costMatrixClass)

TOOLS.append(squareTool)
```

MathCrow Visualization

- Figure shows nodes and edges.
- "__start__" node receives user/input prompt. [solid edge] and passes prompt to node "agent".
- "agent" analyzes prompt to decide which/if any tool to call. [dotted conditional edge]
- if LAP tool was called, routes to LAP node, which displays output.
- if RAG tool was called, evaluate if the retrieved context is relevant. if so, "generate", otherwise "rewrite" query.

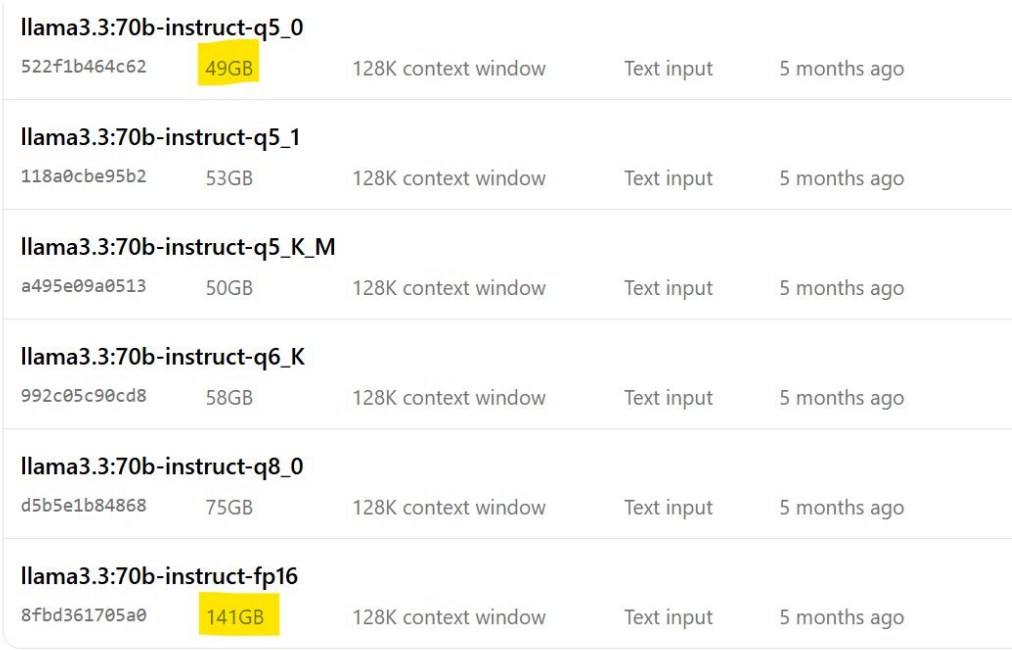


Local LLM deployment

- Trained quantized models have much lower computation (inference) cost and can be deployed locally (e.g. ISU Nova Cluster)
- State-of-the-art quantized models available at ollama.

Local LLM deployment

- https://ollama.com/library/llama3.3/tags
- Notice reduced (49 GB) model size for quantized version.



Conclusion

- Proof of concept extension of ChemCrow to Math Specific Agent, MathCrow.
- Zero-cost local and private agentic framework deployment using quantization and ISU NOVA A-100 GPU.