An architecture for Artificial General Intelligence: The FrameFormer

Index

1. Reflection on current problems of LLMs for AGI
2. A new computing paradigm: Natural Language Computing Interface
3. Architecture for AGI
4. Next steps: Focus, Team, upcoming meetings

Current LLMs will not get us to AGI

SELF SUPERVISED LEARNING

It allowed the language models to consume the data needed for training according to scaling laws.

- **Pr**os
  Ability to use as training data all text ever produced, without the need for labels.
- **Cons**
  Does not **build hierarchical knowledge** and does not challenge new data before learning it.

THE PROBLEM.

- **We are asking too much from the models**: knowledge base, instructions, interpretive and generative parts.

PROBLEM SOLVING IS A METHOD

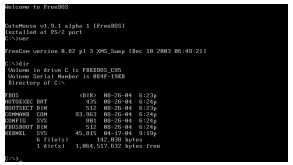
- Models **do not take advantage of human knowledge** in approaching problem solving (scientific method, heuristic techniques...)
- Techniques such as **Chain of Thought** and **Tree of Though** showed net improvements over the basic model in certain tasks[1]

A NEW APPROACH

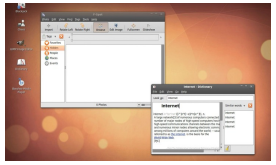
- **Interpretive and generative component ->** LLMs
- **Procedures ->** Classic programs
- **Knowledge Base ->** External Memory

1: Observed 4% to 74% improvement in mini crosswords and creative writing tasks.
Source: Tree of Thoughts: Deliberate Problem Solving with Large Language Models.

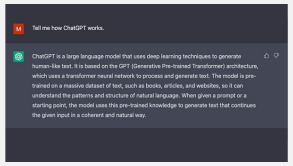
Natural Language Computing Interface - User



Command line: command->action in the file system



Graphics: more intuitive, uses our spatial intuition.
Graphical command -> action in the file system



Natural Language:
Exploits our semantic intuition.
The transmitted command is interpreted, and an action is generated.

NATURAL LANGUAGE COMPUTING INTERFACE

Convenient when the **data**, **instructions**, and **outputs** are all **textual**

Pros:

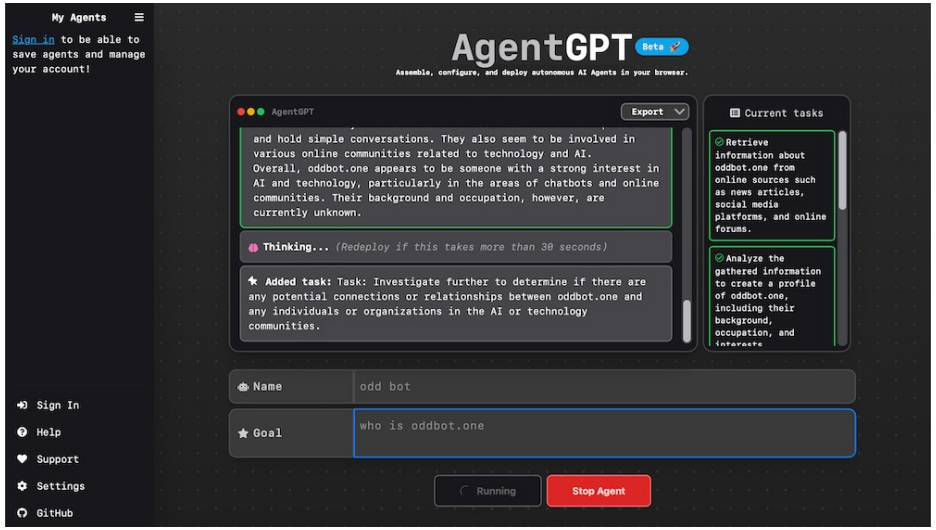
- The user **does not have to learn new languages**.
- **Easy to debug**, since both the instruction set and the data can be in natural language. (Compared to NNs where knowledge is encoded in the weight set).
- It is easy to **import the entire body of knowledge** ever written down by humankind.

Cons:

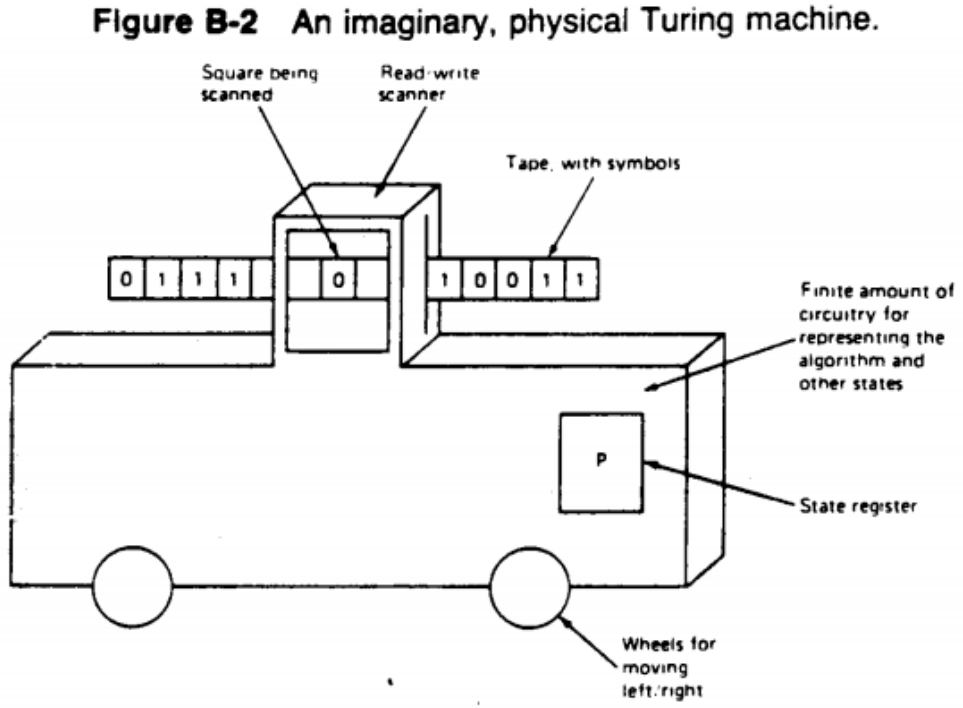
- **Expensive** compared to computation
- **Security**: easy to imagine injections since data and instructions have the same format (language).

Natural Language Computing Interface - System

- The execution of a Chain of Thought can be seen as a classical program, each piece of the inner monologue being an intermediate variable.
- **New computation paradigm**, where you can have the same constructs as a high-level language (if, for loop...)

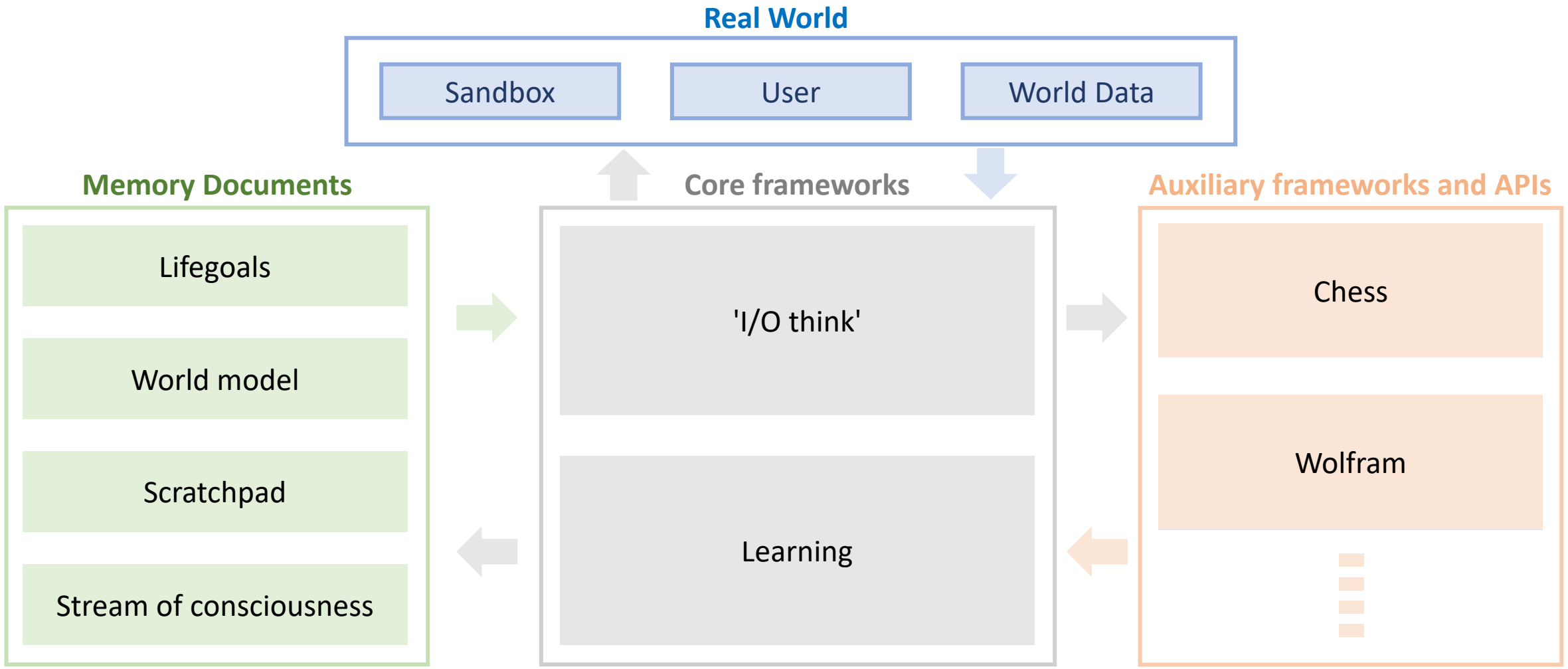


- No need for interpreters and compilers: **everything is in natural language**!
- It is like a **Turing machine** reading text on tape, where the head is an **LLM**!



How do we use NLCI to build AGI?

The architecture for AGI: the FrameFormer



Pros and cons

Pros

- **Expert from day one:** Can import all human knowledge, and receive hardcoded instructions.
- **Easy to debug**: information is stored as natural language.

Cons

- **Convergence:** The system may take too many LLM calls to write a framework.
- **Possible injections:** instructions and data are in the same language-> problem already present in RAM model

Research focus and next steps

POSSIBLE ROUTES

1. Architecture improvement
2. Test the architecture on a task
3. Natural Language Computing Interface

POSSIBLE TASKS

Criteria:

1. Good execution should be easily verifiable
2. The task should be scalable (chessboard 2x2->3x3)

Proposals:

- Let's play Go / Tic Tac Toe
- Let's do mathematical operations
- Let's solve difficult math problems (not self supervised)
- ...

Team

Team Scuola Normale

Marco Eterno

Paolo Tognini
Ph.D. in Quantum Machine Learning

Team Indigo.ai

Backup

Core Framework - 'I/O think'

Mission

- Can make use of auxiliary frameworks to complete tasks (similar to ToolFormers)
- If they do not perform well, it asks the Learning framework to write a new framework for the task.
- It should be the rational consciousness of the sentient being.
- It aims to achieve lifegoals
- It has an internal monologue (Chain of Thought and Tree of Thought)

Properties

Long-term memory: of yet-to-be-defined format, textual or vector.

Dedicated Scratchpad: To save the information it discovers as it goes along

Routines: to carry out sub-tasks of the main problem

Actions

He writes the World Model: with the knowledge he acquires.

Writes stream of consciousness

Handles call and I/O operations with frameworks

Authorizes sandbox executions of frameworks

Authorizes interactions with the real world of frameworks

Interfaces with the user

Core Framework - Learning

Mission

- It is responsible for writing new frameworks when the 'I/O think' detects that none of the current ones are suitable for a certain task
- **Uses the scientific method:** tests Frameworks on the real world, and improves them until they work
- It contains what humanity has learned about how to learn new skills/activities.

Properties	Actions
Long-term memory: of yet-to-be-defined format, text/vector/embedding	**Writes new frameworks**
Dedicated Scratchpad: To save the information it discovers as it builds a framework	
Routines: to carry out sub-tasks of the main problem (e.g. Routines "decompose problem", "find effective representation")	

Auxiliary Framework - illustrative

Mission

- Performs the task for which it was created (playing chess, answering math questions,...)

Properties

Text description: useful for understanding in which cases to use the framework.

Long-term memory: of yet-to-be-defined format, textual or vector.

Dedicated Scratchpad: To save the information it discovers as it goes along

Routines: to carry out sub-tasks of the main problem

Actions

Sandbox execution of code: through 'I/O think' approval

Interface with external data and the user: through 'I/O think' approval