An architecture for Artificial General Intelligence: The FrameFormer

Indice

1. Riflessione sui problemi attuali di LLMs per AGI
2. Un nuovo paradigma di computazione: Natural Language Computing Interface
3. L’architettura per AGI
4. Prossimi passi: Focus, Team, prossimi incontri

I LLMs attuali non arriveranno ad AGI

IL SELF SUPERVISED LEARNING

Ha consentito ai modelli di linguaggio di consumare I dati necessari al training secondo le scaling laws.

- **Pro**
  Capacità di utilizzare come training data tutto il testo mai prodotto, senza bisogno di labels.
- **Contro**
  non costruisce una conoscenza gerarchica e non fa challenge sui nuovi dati prima di impararli.

IL PROBLEMA

- **Stiamo chiedendo troppo ai modelli**: Knowledge base, istruzioni, parte interpretativa e generativa.

IL PROBLEM SOLVING È METODO

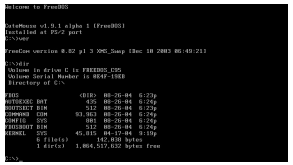
- I modelli **non sfruttano la conoscenza umana** nell'approccio alla risoluzione dei problemi (metodo scientifico, tecniche euristiche…)
- Tecniche come **Chain of Thought** e **Tree of Though** hanno mostrato netti miglioramenti rispetto al modello base in determinati task[1]

UN NUOVO APPROCCIO

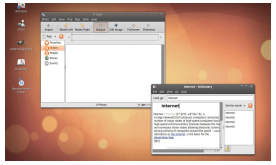
- **Componente interpretativa e generativa ->** LLMs
- **Procedure ->** Programmi classici
- **Knowledge Base ->** Memoria Esterna

1: Osservato un miglioramento dal 4 al 74% in task di mini crosswords e creative writing.
Fonte: Tree of Thoughts: Deliberate Problem Solving with Large Language Models

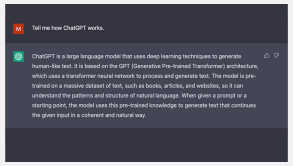
Natural Language Computing Interface – User



Linea di comando: comando->azione nel file system



Grafica: più intuitiva, sfrutta la nostra intuizione spaziale.
Comando grafico -> azione nel file system



Linguaggio Naturale:
Sfrutta la nostra intuizione semantica.
Il comando trasmesso è interpretato, e si fa corrispondere una azione.

NATURAL LANGUAGE COMPUTING INTERFACE

Conveniente quando sia I **dati**, che le **istruzioni**, che gli **output** sono **testuali**

Pro:

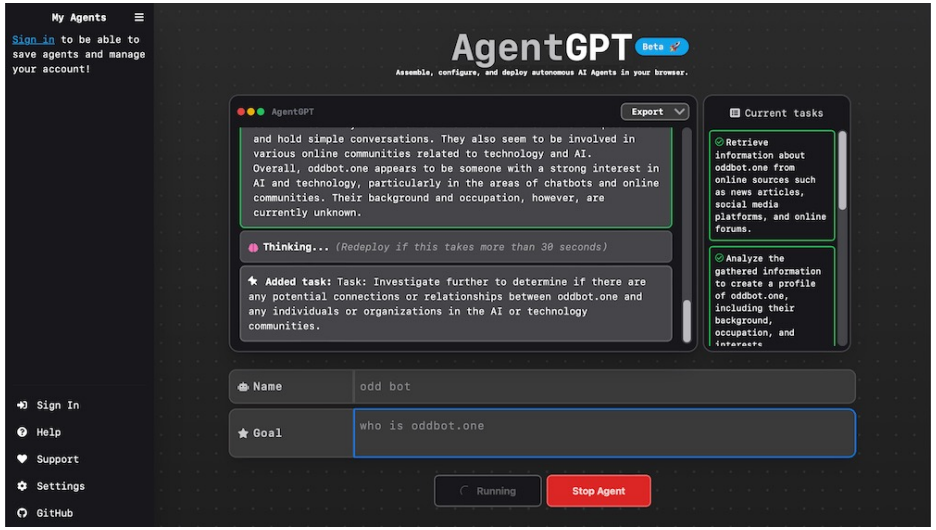
- L'utente **non deve imparare nuovi linguaggi**.
- **Facile da debuggare**, in quanto sia il set di istruzioni che I dati possono essere in linguaggio naturale. (rispetto a NN in cui la conoscenza è codificata nel set di pesi)
- È facile **importare l'intero corpus di conoscenza** mai scritto.

Contro:

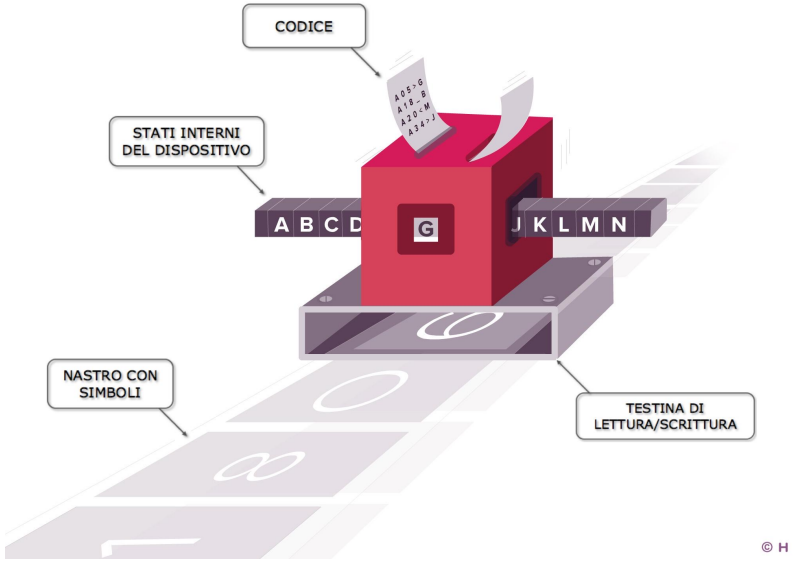
- **Costoso** rispetto alla computazione
- **Sicurezza**: facile immaginare delle injection visto che dati ed istruzioni hanno la stessa forma.

Natural Language Computing Interface - System

- l'esecuzione di una Chain of Thought può essere visto come un programma classico, ogni pezzo del monologo interiore è una variabile intermedia.
- **Nuovo paradigma di computazione**, in cui è possibile avere gli stessi costrutti di un linguaggio ad alto livello (if, for…)

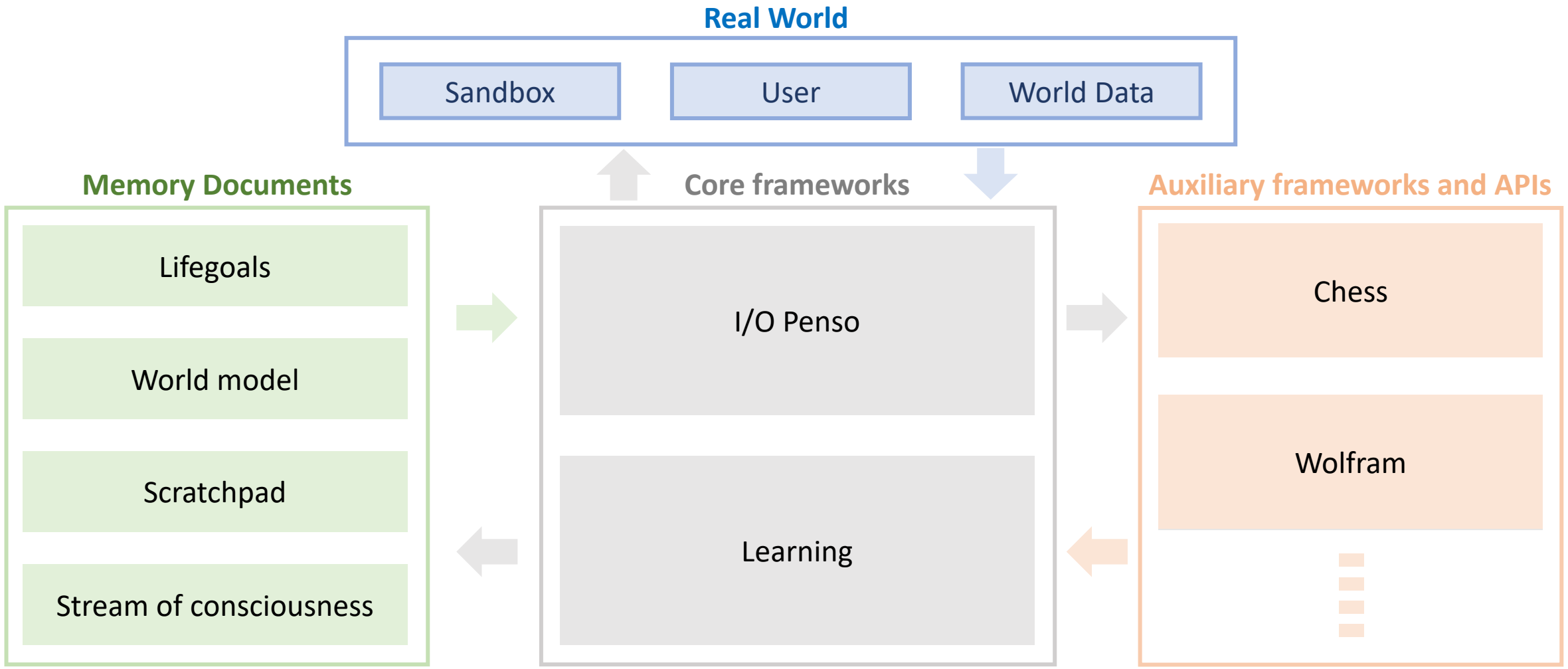


- Non servono interpreti e compilatori: **tutto è in linguaggio macchina**!
- È come una **macchina di Turing** che legge testo sul nastro, dove la testina è un **LLM**!



Come usiamo NLCI per costruire AGI?

L’architettura per AGI: il FrameFormer



Pro e contro

Pro

- **Esperto fin dal primo giorno:** Può importare la conoscenza umana, e ricevere istruzioni hardcoded.
- **Facile da debuggare**: le informazioni sono immagazzinate in forma di linguaggio naturale.

Contro

- **Convergenza:** Il sistema potrebbe impiegare troppe chiamate del LLM per scrivere un framework.
- **Possibili injections:** istruzioni e dati sono nello stesso linguaggio-> problema già presente nel modello RAM

Focus della ricerca e prossimi passi

POSSIBILI FILONI

1. Miglioramento dell'architettura
2. Messa in pratica dell'architettura su un task
3. Natural Language Computing Interface

POSSIBILI TASK

Criteri:

1. La buona esecuzione deve essere facilmente verificabile
2. Il task dovrebbe essere estendibile (scacchiera 2x2->3x3)

Proposte:

- Giochiamo a Go/Tris
- Facciamo operazioni matematiche
- Risolviamo problemi di matematica difficili (non è self supervised)
- …

Team

Team Scuola Normale

Marco Eterno

Paolo Tognini
PhD in Quantum Machine Learning

Team Indigo.ai

Backup

Core Framework – I/O Penso

Missione

- Può avvalersi dei framework ausiliari per portare a termine le attività (analogamente ai toolformer)
- Nel caso in cui questi non performassero bene, chiede al framework Learning di scrivere un nuovo framework per l'attività.
- Dovrebbe essere la coscienza razionale dell'essere senziente.
- Ha come obiettivo raggiungere i lifegoals
- Ha un monologo interno (Chain of Thought e Tree of Thought)

Proprietà

Memoria a lungo termine: di formato ancora da definire, testuale o vettoriale.

Scratchpad dedicato: Per salvare le informazioni che scopre man mano

Routines: per svolgere sotto-task del problema principale

Azioni

Scrive il World Model: con le conoscenze che acquisisce

Scrive il flusso di coscienza

Gestisce la chiamata e le operazioni I/O con i frameworks

Autorizza le esecuzioni in sandbox dei frameworks

Autorizza le interazioni con il mondo reale dei frameworks

Si interfaccia con l'utente

Core Framework – Learning

Missione

- Si occupa di scrivere nuovi framework quando l'I/O penso rileva che nessuno degli attuali è adatto ad un certo task
- **Utilizza il metodo scientifico:** testa I framework sul mondo reale, e li migliora fino a quando funzionano
- Raccoglie ciò che l'umanità ha imparato sul come imparare nuove skills/attività.

Proprietà	Azioni
Memoria a lungo termine: di formato ancora da definire, testuale/vettoriale/embedding **Scratchpad dedicato:** Per salvarsi le informazioni che scopre man mano che costruisce un framework **Routines:** per svolgere sotto-task del problema principale (es. Routine "decomponi problema", "trova rappresentazione efficace")	**Scrive nuovi frameworks**

Auxiliary Framework - Esemplificativo

Missione

- Esegue il task per cui è stato creato (giocare a scacchi, rispondere a domande di matematica,…)

Proprietà

Descrizione testuale: utile per capire in quali casi utilizzare il framework.

Memoria a lungo termine: di formato ancora da definire, testuale o vettoriale.

Scratchpad dedicato: Per salvarsi le informazioni che scopre man mano

Routines: per svolgere sotto-task del problema principale

Azioni

Esecuzione in sandbox del codice: tramite approvazione dell' I/O Penso

Interfaccia con I dati esterni e con l'utente: tramite approvazione dell' I/O Penso