

Dall'AI agli Agenti LLM

Francesco La Rosa

13 Dicembre 2025

Università di Messina



Obiettivi del Talk

- I fondamenti dell'AI e dei Large Language Models (LLM)
- Le principali soluzioni LLM *closed source* e *open source*
- LLM tramite API (OpenAI)
- Capire cosa sono agenti, tool e orchestrazione.
- Un sistema multi-agente complesso.
- Progetto finale: `deep_research.py`.

Modulo 1: Fondamenti di AI e LLM

AI, Machine Learning e Generative AI

- **AI**: sistemi che svolgono compiti che richiederebbero intelligenza umana.
- **Machine Learning**: modelli che imparano dai dati.
- **Deep Learning**: reti neurali profonde per compiti complessi.
- **Generative AI**: modelli che *generano* contenuti (testo, immagini, audio, codice).
- I LLM sono modelli di Generative AI specializzati sul testo.

Cos'è un Large Language Model (LLM)

- Modello statistico del linguaggio addestrato su enormi quantità di testo.
- Predice il **prossimo token** dato il contesto precedente.
- Basato sull'architettura **Transformer**.
- Può eseguire:
 - generazione di testo
 - traduzione
 - riepilogo
 - Q&A
 - generazione/analisi di codice

Token, contesto e parametri

- **Token:** unità elementare di testo (parola, sottoparola, simbolo).
- **Contesto:** numero massimo di token che il modello considera in input.
- **Parametri:** pesi del modello (milioni o miliardi).
- Impatti pratici:
 - tempo di risposta
 - costo computazionale
 - capacità di ragionamento su contesti lunghi

Limiti dei LLM (1/2)

- **Allucinazioni**: risposte plausibili ma sbagliate.
- **Mancanza di aggiornamento in tempo reale** (a meno di collegamenti esterni).
- **Bias** nei dati di addestramento.
- **Finestra di contesto** limitata.
- Necessità di **prompting** e **controlli** adeguati.

Limiti dei LLM (2/2) - Allucinazioni

The window contains the following text:

Generi: Novella letteraria, romanzo
Soggetto principale: L'amore, la libertà, la lotta per l'autonomia e la fedeltà.

Trama e principali eventi

I Promessi Sposi è una novella letteraria che segue la vita di due individui, Alessandro e Giuditta, che si sposano in un matrimonio promesso. La trama si sviluppa in due fasi:

1. **La promessa:** Alessandro, un giovane uomo di 19 anni, è promesso a Giuditta, una giovane donna di 18 anni, per un matrimonio che le è stato promesso da sua madre. L'uomo, però, non è pronto per la vita di sposo, e la sua famiglia lo rifiuta.
2. **La vita in coppia:** Alessandro e Giuditta si sposano, ma la loro relazione è marciata da disaccordi, disprezzo e disillusione. La donna, però, non si rassegna al suo destino e cerca di

Send a message

+ ⊕ qwen3:1.7b □

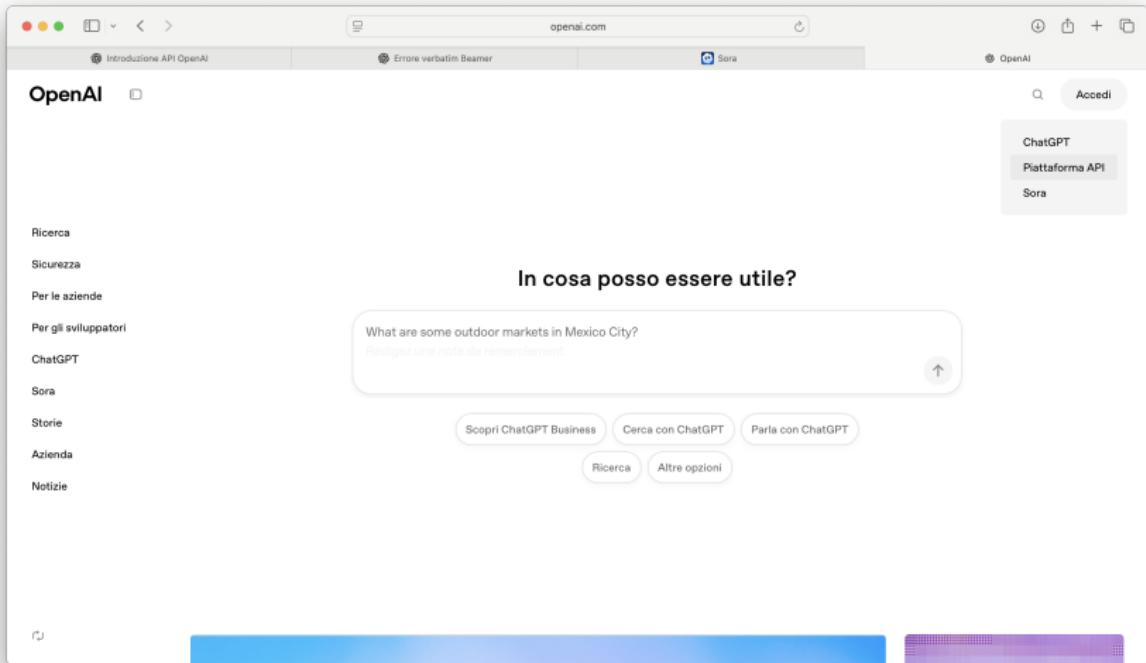
Allucinazioni in un modello LLM open source.

Modulo 2: Ecosistema LLM Open & Closed

LLM Closed Source

- API di provider commerciali:
 - OpenAI (serie GPT)
 - Anthropic (Claude)
 - Google (Gemini)
- Pro:
 - Modelli molto performanti
 - Infrastruttura gestita
 - Integrazioni (tool, file, immagini, ecc.)
- Contro:
 - Costo a consumo
 - Dipendenza dal fornitore
 - Meno controllo interno

LLM Closed Source - OpenAI



OpenAI GUI

- Modelli aperti (pesi disponibili) es.:
 - LLaMA
 - Mistral / Mixtral
 - Qwen
 - Falcon, OLMo, ecc.
- Pro:
 - Controllo totale (on-prem, privacy)
 - Costo variabile (CPU/GPU proprie)
 - Possibilità di fine-tuning
- Contro:
 - Richiedono risorse hardware
 - DevOps più complesso

LLM Open Source - Ollama

The screenshot shows a window titled "Scrivi un programma in C per leggere il contenuto di un file di testo e stamparlo a schermo". A status message indicates "Thought for 66.7 seconds". Below the instruction, a code completion result is displayed:

```
c
#include <stdio.h>

int main() {
    FILE *file;
    char buffer[1024]; // Buffer per il contenuto del file

    // Richiede all'utente di inserire il nome del file
    printf("Inserisci il nome del file da leggere: ");
    scanf("%s", buffer);
```

A "Send a message" button is at the bottom left, and a "Copy" button is at the top right of the code block. The bottom right corner of the window contains a model selection bar with "qwen3:1.7b" and a dropdown arrow.

Criteri di scelta del modello

- Licenza (commerciale, open, restrizioni)
- Costo (API vs infrastruttura propria)
- Performance su task specifico
- Latency e throughput
- Esigenze di privacy e compliance
- Possibilità di esecuzione on-prem o edge

Modulo 3: Laboratorio

Introduzione alle API OpenAI

GPT - Generative Pre-trained Transformer

- Modello generativo addestrato su grandi dataset.
- Produce testo, codice, spiegazioni, piani.
- I modelli OpenAI (GPT-5, GPT-4o, ecc.) supportano:
 - Ragionamento
 - Tool calling
 - Conversazioni multi-turno

Completions vs Responses API

API legacy (storiche):

- Chat completions: modello produce una risposta e basta.

API moderne:

- `client.responses.create()`
- Accesso migliorato a strumenti, memoria, agenti.

II Client Python

Installazione e configurazione del client Python

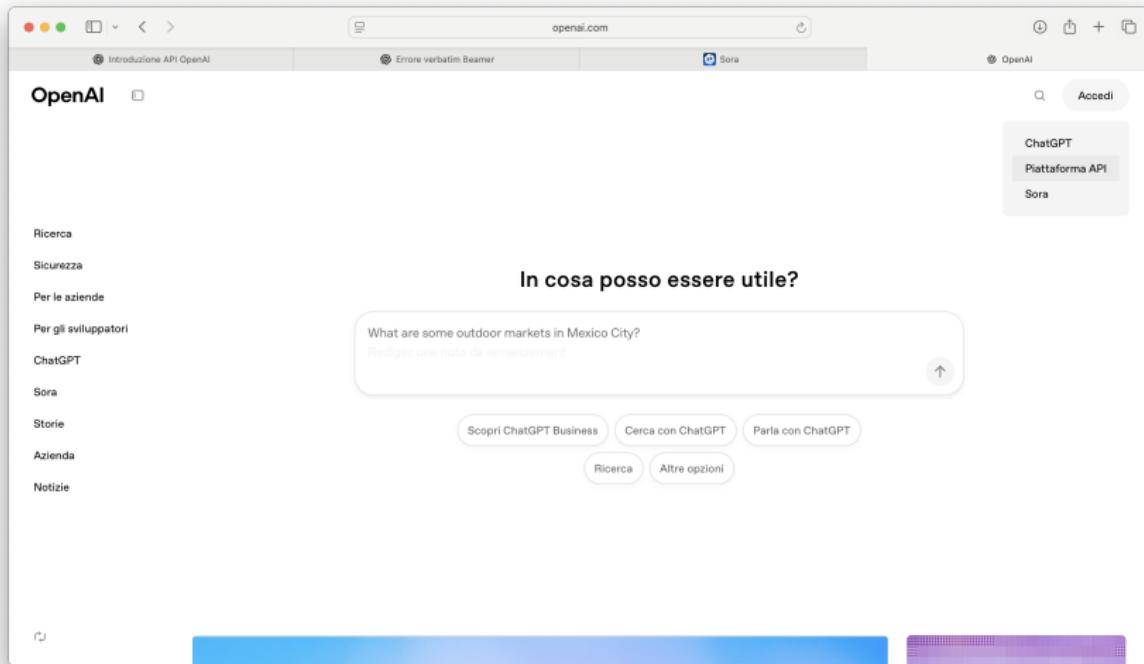
Installazione

```
pip install openai python-dotenv
```

Variabili d'ambiente

```
export OPENAI_API_KEY="sk-..."
```

Come creare un API-KEY (1/3)



OpenAI GUI

Come creare un API-KEY (2/3)

The screenshot shows the OpenAI Platform interface on a Mac OS X system. The browser window has tabs for "Introduzione API OpenAI", "Errone verbatim Beamer", "Sora", and "OpenAI". The main content area is titled "OpenAI Platform".

Developer quickstart: A section with the heading "Developer quickstart" and sub-headings "Make your first API request in minutes." and "Learn the basics of the OpenAI platform.". It includes a "Get started" button and a code snippet in JavaScript:

```
import OpenAI from "openai";
const client = new OpenAI();
const response = await client.responses.create({
  model: "gpt-5.1",
  input: "Write a short bedtime story about a unicorn."
});
console.log(response.output.text);
```

Models: A section showing three models: "gpt-5.1", "gpt-5-mini", and "gpt-5-nano". Each model has a "View all" link.

Model	Status	Description
gpt-5.1	New	The best model for coding and agentic tasks with configurable reasoning effort.
gpt-5-mini		A faster, cost-efficient version of GPT-5 for well-defined tasks.
gpt-5-nano		Fastest, most cost-efficient version of GPT-5.

OpenAI GUI

Come creare un API-KEY (3/3)

The screenshot shows the OpenAI platform interface for managing API keys. The left sidebar contains navigation links such as Create, Chat, Agent Builder, Audio, Images, Videos, Assistants, Manage, Usage, API key*, Logs, Storage, Batches, Optimize, Evaluation, and Fine-tuning. The main content area is titled "API keys" and displays two entries:

NAME	STATUS	SECRET KEY	CREATED	LAST USED	CREATED BY	PERMISSIONS
consapevolmente-key	Active	sk-...,fdka	4 dic 2025	8 dic 2025	Francesco La Rosa	All
unipf-key	Active	sk-...,ZnIA	24 ott 2025	27 ott 2025	Francesco La Rosa	All

On the right side, there is a button labeled "+ Create new secret key".

OpenAI GUI

Prima chiamata API

Codice Python minimo

```
from openai import OpenAI

client = OpenAI()

response = client.responses.create(
    model="gpt-4.1-mini",
    input="Spiegami cos'è un LLM in 3 righe"
)

print(response.output_text)
```

Agenti e Tool

Cos'è un Agente

Un agente è costituito da:

- Un LLM
- Uno scopo (system prompt)
- Uno o più strumenti (tool)
- Una memoria locale

Gli agenti permettono:

- Ragionamento con strumenti esterni
- Pianificazione
- Orchestrazione di processi complessi

Esempio di Tool (astratto)

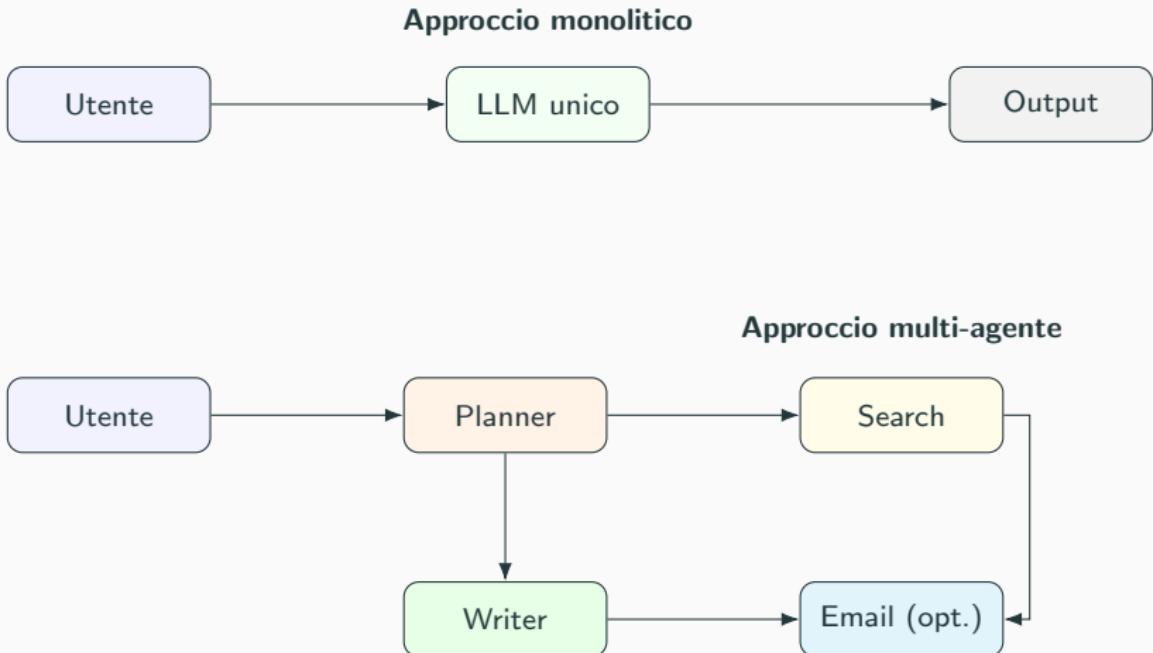
Tool Python: somma

```
def somma(a, b):
    return {"result": a + b}

response = client.responses.create(
    model="gpt-4.1-mini",
    input="Calcola 5+7",
    tools=[
        {
            "type": "function",
            "function": {
                "name": "somma",
                "arguments": {"a": 5, "b": 7},
                "output": {"type": "object"},
            }
        }
    ]
)
```

Diagrammi concettuali

LLM singolo vs sistema multi-agente (diagramma)



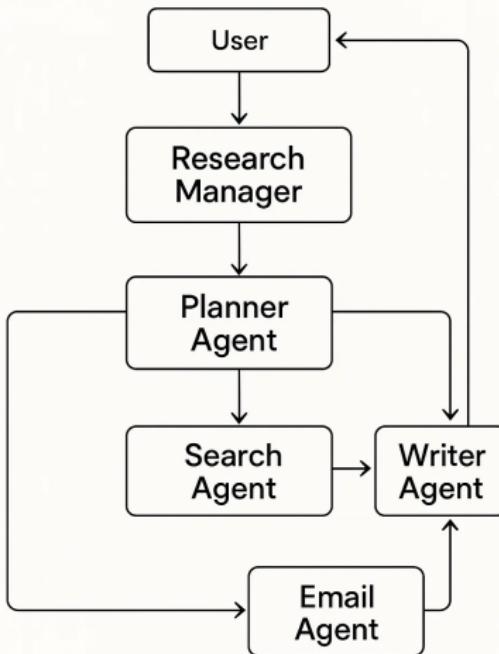
Architettura Multi-Agente

Perché usare più agenti?

- Un singolo LLM può sbagliare o perdere contesto.
- Specializzazione per ruolo = maggiore qualità.
- Cooperazione strutturata = risultati complessi.
- Modularità = estensibilità (aggiungo/rimuovo agenti).

Progetto: un sistema multi-agente di Deep Research

DEEP RESEARCH



Gli agenti del progetto

Il sistema multi-agente contiene:

- **Planner Agent**: definisce il piano di ricerca.
- **Search Agent**: esegue ricerche web/API.
- **Writer Agent**: produce testi strutturati.
- **Email Agent**: invia report (facoltativo).
- **Research Manager**: coordina il flusso.

Flusso della ricerca (testuale)

1. Input dell'utente (tema di ricerca).
2. Pianificazione del lavoro (Planner).
3. Raccolta informazioni (Search).
4. Elaborazione contenuti (Writer).
5. Output finale (Manager / Email).

Analisi del Codice del Progetto

Progetto: Deep Research

- deep_research.py
- research_manager.py
- planner_agent.py
- search_agent.py
- writer_agent.py
- email_agent.py

deep_research.py: panoramica

- Punto d'ingresso dell'app.
- Carica e istanzia gli agenti.
- Passa l'obiettivo al Research Manager.
- Stampa e salva l'output.

deep_research.py

The screenshot shows a terminal window titled "deep_research" with the following content:

```
deep_research.py 2
deep_research.py > ..
1 import gradio as gr
2 from dotenv import load_dotenv
3 from research_manager import ResearchManager
4
5 load_dotenv	override=True
6
7
8 async def run(query: str):
9     async for chunk in ResearchManager().run(query):
10         yield chunk
11
12
13 with gr.Blocks(themes.Default(primary_bg="#sky")) as ui:
14     gr.Markdown("# Deep Research")
15     query_textbox = gr.Textbox(label="What topic would you like to research?")
16     run_button = gr.Button("Run", variant="primary")
17     report = gr.Markdown(label="Report")
18
19     run_button.click(fnrue, inputs=query_textbox, outputs=report)
20     query_textbox.submit(fnrue, inputs=query_textbox, outputs=report)
21
22 ui.launch(inbrowser=True)
23 | No output, run or generate
```

The terminal window includes a sidebar with "OUTLINE", "TIMELINE", and a file tree for "DEEP_RESEARCH" containing "deep_research.py", "email_agent.py", "planner_agent.py", "research_manager.py", "search_agent.py", and "writer_agent.py". The status bar at the bottom shows "Cursor Tab" and "Ln 23, Col 1" along with other system information.

research_manager.py - panoramica

- Coordina il flusso globale.
- Gestisce errori, ripetizioni, strutture dati.

research_manager.py

```
research_manager.py : X
research_manager.py > ...
1  from agents import Runner, trace, gen_trace_id
2  from search_agent import search_agent
3  from planner_agent import planner_agent, WebSearchItem, WebSearchPlan
4  from writer_agent import writer_agent, ReportData
5  from email_agent import writer_email_agent
6  import asyncio
7
8  class ResearchManager:
9
10     async def run(self, query: str):
11         """ Run the deep research process, yielding the status updates and the final report"""
12         trace_id = await gen_trace_id()
13         with trace("Deep Research", trace_id=trace_id):
14             print(f"View trace: https://platform.openai.com/traces?trace_id={trace_id}")
15             yield f"View trace: https://platform.openai.com/traces?trace_id={trace_id}"
16             print("Starting research...")
17             search_plan = await self.plan_searches(query)
18             yield "Searches planned, starting to search..."
19             search_results = await self.perform_searches(search_plan)
20             yield "Searches complete, writing report..."
21             report = await self.write_report(query, search_results)
22             yield "Report written, sending email..."
23             await self.send_email_report()
24             yield "Email sent, research complete"
25             yield report.markdown_report
26
27
28     async def plan_searches(self, query: str) -> WebSearchPlan:
29         """ Plan the searches to perform for the query """
30         print("Planning searches...")
31         result = await Runner.run(
32             planner_agent,
33             f"Query: {query}",
34         )
35         print(f"Will perform {len(result.final_output.searches)} searches")
36         return result.final_output_as[WebSearchPlan]
37
38     async def perform_searches(self, search_plan: WebSearchPlan) -> list[str]:
39         """ Perform the searches to perform for the query """
40         print("Searching...")
```

planner_agent.py - panoramica

- Scopo: decomporre l'obiettivo in sotto-task.
- Input: richiesta dell'utente.
- Output: elenco strutturato delle attività.

planner_agent.py

The screenshot shows a terminal window titled "deep_research" with two tabs open: "deep_research.py" and "planner_agent.py". The "planner_agent.py" tab contains the following Python code:

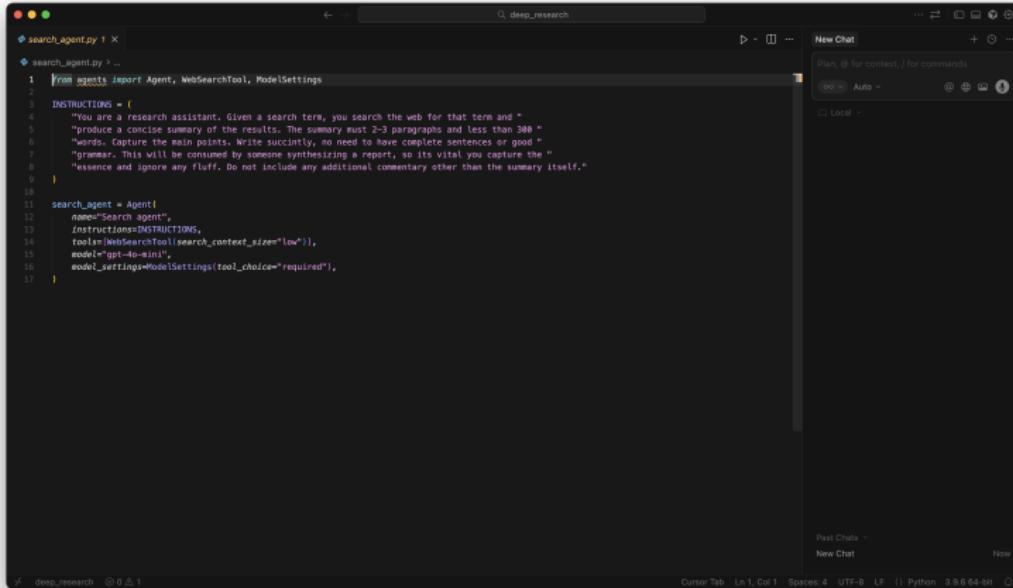
```
❶ deep_research.py 2 | ❷ planner_agent.py 2 |
❸ planner_agent.py > ...
1  from pydantic import BaseModel, Field
2  from agents import Agent
3
4  HOW_MANY_SEARCHES = 5
5
6  INSTRUCTIONS = f"You are a helpful research assistant. Given a query, come up with a set of web searches \
7  to perform to best answer the query. Output {HOW_MANY_SEARCHES} terms to query for."
8
9
10 class WebSearchItem(BaseModel):
11     reason: str = Field(description="Your reasoning for why this search is important to the query.")
12     query: str = Field(description="The search term to use for the web search.")
13
14
15 class WebSearchPlan(BaseModel):
16     searches: List[WebSearchItem] = Field(description="A list of web searches to perform to best answer the query.")
17
18 planner_agent = Agent(
19     name="PlannerAgent",
20     instructions=INSTRUCTIONS,
21     model="gpt-4o-min",
22     output_type=WebSearchPlan,
23 )
```

The terminal interface includes a sidebar with "New Chat" and "Past Chats" sections, and a status bar at the bottom showing "Cursor Tab: Lin 1, Col 1" and "Spaces: 4 - UTF-8 - LF - () Python 3.9.6 64-bit".

search_agent.py - panoramica

- Scopo: raccogliere dati da fonti esterne.
- Può usare API reali (Bing, PubMed...) o strumenti simulati.

search_agent.py



```
search_agent.py 1
search_agent.py > ..
1 | from agents import Agent, WebSearchTool, ModelSettings
2 |
3 INSTRUCTIONS = [
4     "You are a research assistant. Given a search term, you search the web for that term and",
5     "produce a concise summary of the results. The summary must 2-3 paragraphs and less than 300",
6     "words. Capture the main points. Write succinctly, no need to have complete sentences or good",
7     "grammar. This will be consumed by someone synthesizing a report, so its vital you capture the",
8     "essence and ignore any fluff. Do not include any additional commentary other than the summary itself."
9 ]
10
11 search_agent = Agent(
12     name="Search agent",
13     instructions=INSTRUCTIONS,
14     tools=[WebSearchTool(search_context_size="low")],
15     model="gpt-4o-mini",
16     model_settings=ModelSettings(tool_choice="required"),
17 )
```

writer_agent.py - panoramica

- Unisce tutte le informazioni.
- Produce report formattati o articoli.

writer_agent.py

```
writer_agent.py 2 X
writer_agent.py > ...
1 | from pydantic import BaseModel, Field
2 | from agents import Agent
3 |
4 | INSTRUCTIONS = [
5 |     "You are a senior researcher tasked with writing a cohesive report for a research query.",
6 |     "You will be provided with the original query, and some initial research done by a research assistant.\n",
7 |     "You should first come up with an outline for the report that describes the structure and\n",
8 |     "flow of the report. Then, generate the report and return that as your final output.\n",
9 |     "The final output should be in markdown format, and it should be lengthy and detailed. Aim\n",
10 |     "for 5-10 pages of content, at least 1000 words."
11 | ]
12 |
13 |
14 | class ReportData(BaseModel):
15 |     short_summary: str = Field(description="A short 2-3 sentence summary of the findings.")
16 |
17 |     markdown_report: str = Field(description="The final report")
18 |
19 |     follow_up_questions: list[str] = Field(description="Suggested topics to research further")
20 |
21 |
22 | writer_agent = Agent(
23 |     name="WriterAgent",
24 |     instructions=INSTRUCTIONS,
25 |     model="gpt-4o-min",
26 |     output_type=ReportData,
27 | )
```

email_agent.py - panoramica

- Scopo: inviare report via email.
- Può usare API reali (SMTP) o strumenti simulati.

email_agent.py

```
deep_research ~ 0 △ 3
↳ deep_research
email_agent.py 3 | ⌂ New Chat
email_agent.py > ⌂ send_email
1 import os
2 from typing import Dict
3
4 import sendgrid
5 from sendgrid.helpers.mail import Email, Mail, Content, To
6 from agents import Agent, function_tool
7
8
9 #function_tool
10 def send_email(subject: str, html_body: str) -> Dict[str, str]:
11     """Send an email with the given subject and HTML body"""
12     sg = sendgrid.SendGridAPIClient(api_key=os.environ.get("SENDGRID_API_KEY"))
13     from_email = Email("from@example.com") # put your verified sender here
14     to_email = To("to@example.com") # put your recipient here
15     content = Content("text/html", html_body)
16     mail = Mail(from_email, to_email, subject, content).get()
17     response = sg.client.mail.send.post(request_body=mail)
18     print("Email response", response.status_code)
19     return "success"
20
21
22 INSTRUCTIONS = """You are able to send a nicely formatted HTML email based on a detailed report.
23 You will be provided with a detailed report. You should use your tool to send one email, providing the
24 report converted into clean, well presented HTML with an appropriate subject line."""
25
26 email_agent = Agent(
27     name="Email agent",
28     instructions=INSTRUCTIONS,
29     tools=[send_email],
30     model="gpt-4o-min1",
31 )
32
```

Past Chats - Now

Cursor Tab Ln 17, Col 59 Spaces: 4 UTF-8 LF () Python 3.9.6 64-bit

Esecuzione finale

Eseguire l'applicazione

Comando tipico

```
python deep_research.py "Ricerca call for paper  
relativa ai Big Data nel 2026"
```

- Mostra i passaggi degli agenti.
- Genera un report finale.
- Possibili estensioni: salvataggio JSON/HTML/PDF.

Deep Research - Big Data

The screenshot shows a web browser window with the following details:

- Address Bar:** 127.0.0.1
- Tab Bar:** Installation | uv, agents/requirements.txt at main · e..., SensGrid, Seleziona Python 3.12, 127.0.0.1:7980, Traces - trace_bfff1f1bec2940198...
- Content Area:**
 - Section Header:** Deep Research
 - Text Input:** What topic would you like to research?
Per favore ricerca call for paper per conferenze internazionali sul BigData previste per il 2026.
 - Button:** Run
 - Section Header:** Comprehensive Report: Call for Papers for International Conferences on Big Data in 2026
 - Section Header:** Table of Contents
 - Table of Contents List:**
 1. [Introduction](#)
 2. [Overview of Big Data Conferences in 2026](#)
 3. [Detailed List of Conferences](#)
 1. [January 2026](#)
 2. [February 2026](#)
 3. [March 2026](#)
 4. [April 2026](#)
 5. [May 2026](#)
 6. [June 2026](#)
 7. [July 2026](#)
 4. [Focus Areas for Submission](#)
 5. [Conclusion](#)
 6. [References](#)
 - Section Header:** Introduction

Deep Research - Big Data

The screenshot shows a web browser window with the following details:

- Address Bar:** 127.0.0.1
- Tab Bar:** Installation | v4, agents/requirements.txt at main · e..., SensGrid, Selectionare Python 3.12, 127.0.0.1:7980, Traces - trace_b1fffb1bec2940198...

The main content area displays a report on upcoming Big Data conferences:

- 1. Big Data and Artificial Intelligence Conference (BDAI 2026)**
 - Date: July 3-5, 2026
 - Location: Chengdu, China
 - Focus: Interdisciplinary advances in AI and big data.
 - Link: [BDAI 2026](#)
- 2. Data-Intensive Research Conference**
 - Date: July 20-23, 2026
 - Location: Minneapolis, MN, USA
 - Focus: Innovative research in data-intensive applications.
 - Link: [DIRC 2026](#)

Focus Areas for Submission

The conferences listed offer platforms that span various focus areas within the realm of Big Data including but not limited to:

- Data Analytics: Methods for analyzing complex data sets to extract meaningful insights.
- Machine Learning: Algorithms and techniques for enabling computers to learn from and make predictions based on data.
- Big Data Applications: Practical applications of big data analytics in fields such as health care, economics, and education.
- Data Security and Privacy: Addressing concerns regarding the protection of sensitive information in the era of big data.
- Interdisciplinary Research: Bridging efforts between different domains to utilize big data effectively.

Researchers are encouraged to visit the respective conference websites for detailed information on submission standards, deadlines, and specific thematic areas of interest.

Conclusion

The year 2026 represents a significant opportunity for researchers in the Big Data domain to contribute to the body of knowledge and engage with a global community. This report consolidates key information on upcoming conferences, their focus areas, and deadlines to facilitate effective participation. Researchers should remain attentive to deadlines and specific submission requirements laid out by each conference to maximize their contribution.

References

- International Conference on Big Data Analytics: [icbda.org](#)
- IEEE BigComp 2026: [icrcmne.org](#)
- International Conference on Big Data and Economic Development: [icbded.org](#)
- All other relevant conference links as provided in the detailed list.

Deep Research - Traces

The screenshot shows the platform.openai.com interface with the URL `platform.openai.com` in the address bar. The page title is "Traces / Research trace". The left sidebar includes sections for Create, Chat, Agent Builder, Audio, Images, Videos, Assistants, Manage, Usage, API keys, Logs (selected), Storage, Batches, Optimize, Evaluation, and Fine-tuning. The main content area displays a list of traces with their execution times:

Agent	Method	Time
PlannerAgent	POST /v1/responses	5,870 ms
Search agent	POST /v1/responses	5,840 ms
Search agent	POST /v1/responses	7,382 ms
Search agent	POST /v1/responses	7,380 ms
Search agent	POST /v1/responses	7,200 ms
Search agent	POST /v1/responses	7,197 ms
Search agent	POST /v1/responses	6,033 ms
Search agent	POST /v1/responses	6,030 ms
Search agent	POST /v1/responses	10.41 s
Search agent	POST /v1/responses	10.40 s
Search agent	POST /v1/responses	5,907 ms
Search agent	POST /v1/responses	6,903 ms
WriterAgent	POST /v1/responses	46.79 s
Email agent	POST /v1/responses	46.79 s
Email agent	POST /v1/responses	65.18 s
Email agent	POST /v1/responses	65.05 s

The right side of the screen shows the "Output" section with a large amount of text content. The text discusses the "short_summary" of several international conferences focused on Big Data, including the International Conference on Big Data Analytics in Tokyo, the IEEE Big Data and Smart Computing in Guangzhou, and the ALIDATA conference in Venice. It also mentions specific calls for papers and submission deadlines. The "workshop_report" section provides a comprehensive report on Big Data conferences in 2026, detailing topics like "Introduction", "Overview of Big Data Conference in 2026", "Detailed Overview of Conference Submission Deadlines", "January 2026", "February 2026", "March 2026", "April 2026", "May 2026", "June 2026", "July 2026", "Focus Areas for Submissions", "Submission", "Conclusion", "Inference", and "References". The text concludes by stating that the world increasingly turns to Big Data for insights across diverse fields.

Deep Research - Email

The screenshot shows a web browser window for platform.openai.com. The left sidebar contains a navigation menu with items like 'Create', 'Chat', 'Agent Builder', 'Audio', 'Images', 'Videos', 'Assistants', 'Manage', 'Usage', 'API keys', 'Logs' (which is selected), 'Storage', 'Batches', 'Optimize', 'Evaluation', and 'Fine-tuning'. The main content area is titled 'Traces / Research trace' and shows a trace for 'trace_64ed95e657e4fb..'. The trace details are as follows:

Action	Duration
POST /v1/agents	6,272 ms
POST /v1/agents	5,890 ms
POST /v1/agents	5,889 ms
POST /v1/agents	5,806 ms
POST /v1/agents	5,803 ms
POST /v1/agents	4,817 ms
POST /v1/agents	4,894 ms
POST /v1/agents	8,188 ms
POST /v1/agents	8,182 ms
POST /v1/agents	49.63 s
POST /v1/agents	49.63 s
POST /v1/agents	59.80 s
send_email	71 ms
POST /v1/agents	2,006 ms

The right side of the interface displays the generated email content:

```
requirements to avoid disqualification.</li><li><strong>Prepare for Peer Review</strong> Submissions typically undergo a peer-review process, and providing clear, concise, and well-structured papers is essential for acceptance. </p><p>Registration Process:</p><ul><li>Online registration is required. Early bird discounts are generally available. Be sure to check the respective websites for specific registration details and potential deadlines.</li><li>Conclusion:</li><p>The landscape of Big Data conferences in 2026 presents ample opportunities for researchers and practitioners to engage and exchange knowledge on the latest trends and technologies in the field. This report has collated significant events, their themes, and key speakers for data professionals and participants. All researchers are encouraged to visit individual conference websites to gather additional information and prepare for their submissions accordingly.</p><div style="text-align: right;"><small>Note: This concludes the report on international big data conferences scheduled for 2026. Stay connected with research developments and embrace the opportunities to share your work with the global big data community.</small></div></div>
```

Below the content, there is an 'Output' section with a 'success' status.

Deep Research - Testo Email

The screenshot shows an Outlook inbox with the following details:

- Toolbar:** Includes standard icons for Ignora (Ignore), Blocca (Block), Elimina Archiva (Delete/Archive), Segnala (Report), Rispondi (Reply), Rispondi a Invitta Riunione (Reply All), Condividi in Teams (Share in Teams), Zoom, Sposta (Move), Pulsante Regole (Rules), Azioni rapide (Quick Actions), Gli Jetto/Da Categorizza (To/Cc/Subject), Contrassegna (Flag), Criterio Posponi (Delay Criteria), Stampa (Print), Altre app (Other apps), Viva Insights (Viva Insights), and Strumenti (Tools).
- Email Preview:** The subject is "Overview of International Conferences on Big Data in 2026". A warning message at the top states: "Non è possibile verificare che questo messaggio di posta elettronica provenga dal mittente, quindi potrebbe non essere sicuro rispondere. Scopri di più" (It is not possible to verify that this email message comes from the sender, so it may not be safe to respond. Learn more) with a link to "Scopri di più". Below this, a button says "Elimina il messaggio di posta elettronica" (Delete the email message).
- Email Content:**

International Conferences on Big Data in 2026

The landscape of Big Data is continuously evolving, and conferences are excellent venues for researchers and practitioners to share insights, advancements, and innovations. This report provides an overview of international conferences focused on Big Data scheduled for 2026, detailing the themes, dates, locations, submission deadlines, and logistical information relevant for prospective participants.

1. Overview of Big Data Conferences in 2026

Several notable conferences will take place throughout the year, each emphasizing different aspects of Big Data, including analytics, privacy, applications, and artificial intelligence. Below is a comprehensive list of significant conferences organized by month:

January 2026

International Conference on Big Data Economy and Sustainable Development (BDESD 2026)
Date: January 4, 2026
Location: Online
Submission Deadline: December 21, 2025
Website: [BDESD 2026](#)
Topics: Big Data mining, Artificial Intelligence, Sustainable Development.

5th International Conference on Big Data, Information and Computer Network (BDICN 2026)

Estensioni

Come estendere il sistema

- Aggiungere un agente “Validator”.
- Inserire un Memory Agent persistente.
- Integrare ricerche web reali.
- Deploy come API con FastAPI.
- Creare un’interfaccia Streamlit.

Conclusioni

- Le API OpenAI permettono molto più che generare testo.
- Gli agenti abilitano sistemi complessi e modulari.
- Il progetto multi-agente è una base potente per deep research.

Domande?