



Exceptional service in the national interest

UNLOCKING THE POWER OF LARGE LANGUAGE MODELS: PRACTICAL APPLICATIONS IN SCRIPTS AND PROGRAMS

Anthony Garland

SAND2023-06223C

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.



OUTLINE

- What are LLMs
 - Basic LLM 101
 - Recent advances
- LLMs as part of a system
 - Parts
 - Memory
 - Tools
 - Database
 - LLM
 - Common design patterns
 - Chain of thought /Tree of thought
 - QA over docs
 - Agents

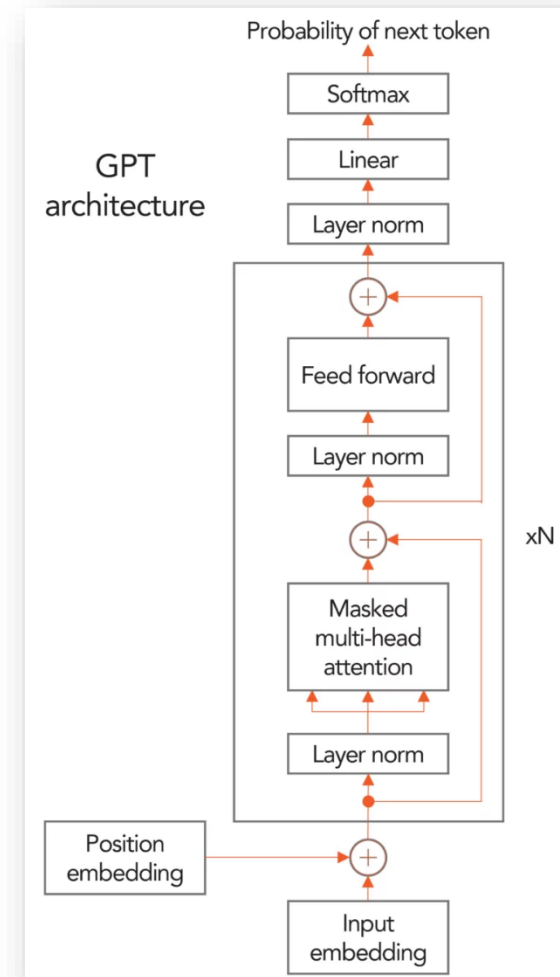
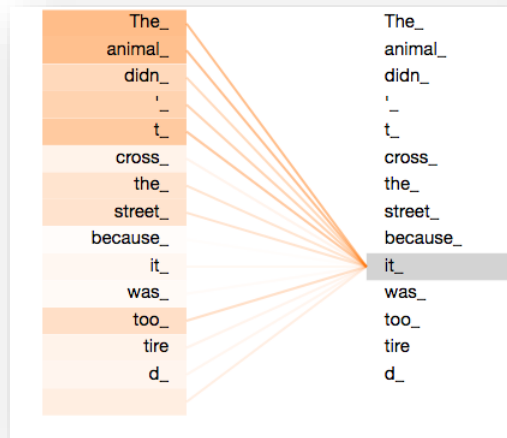




WHAT ARE LLMS?

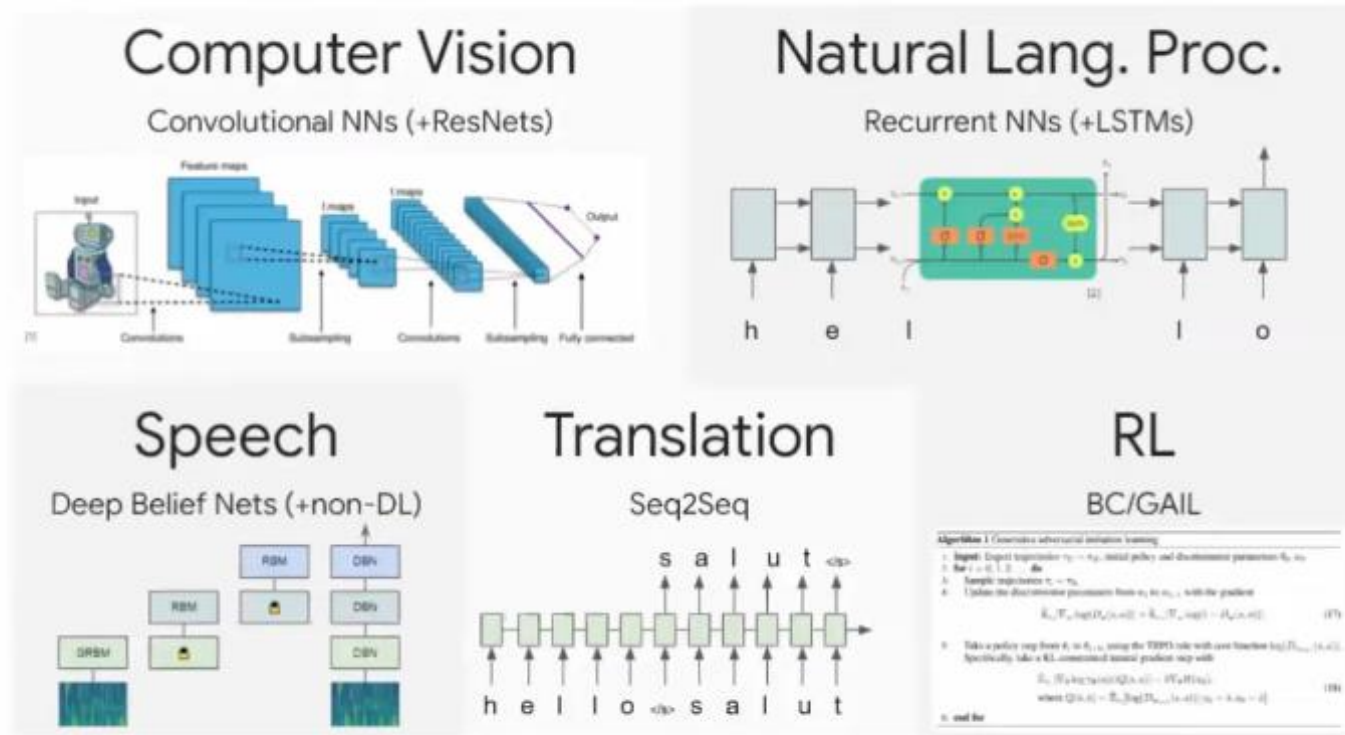
WHAT ARE LLMS

- Modern LLM are stacked layers of transformers
- Transformers
 - Convert input to a vector (embedding)
 - Use self-attention
 - Scaled dot-product
 - Each word can determine how much to pay “attention” to all the other input words
 - Multiple heads. Perform self-attention M times
- Stack transformer layers together N times
- Transformers are a very generic computing paradigm that can use any input



$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

PRE-TRANSFORMER ML LANDSCAPE



POST TRANSFORMER ML LANDSCAPE





WHY INSTRUCT GPT AND CHATGPT WERE INTERESTING

- GPT3 had been around for a while but was hard to use.
- Usability!
 - **Alignment** of the output to be helpful to humans
 - The Chat UI worked well.

Example: Write some code to do X

How to get GPT3 to do what you want

```
You are GPT-3, and you can't do math.

You can do basic math, and your memorization abilities are impressive, but you can't do any complex calculations that a human could not do in their head. You also have an annoying tendency to just make up highly specific, but wrong, answers.

So we hooked you up to a Python 3 kernel, and now you can execute code. If anyone gives you a hard math problem, just use this format and we will take care of the rest:

Question: {{Question with hard calculation.}}

python
{{Code that prints what you need to know}}
output
{{Output of your code}}
Answer: {{Answer}}

Otherwise, use this simpler format:

Question: {{Question without hard calculation}}
Answer: {{Answer}}

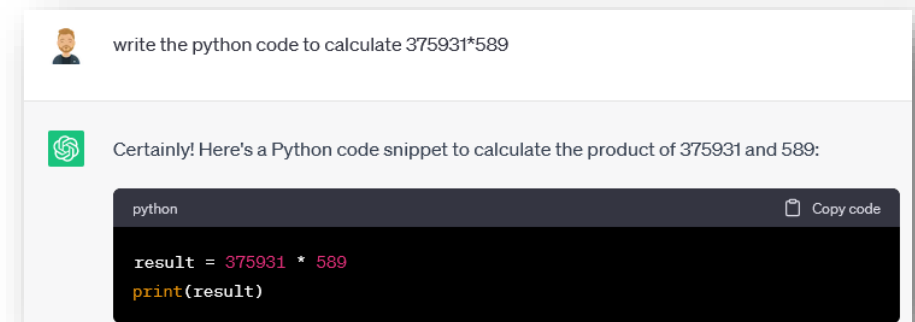
Begin.

Question: What is 37593 * 67?

python
print(37593 * 67)
output
2518731
Answer: 2518731

Question: {question}
```

How to get ChatGPT to do what you want





RECENT ADVANCES IN LLMS

- The “Large” can mean many things
 - Large # parameters ($\geq 7B$ params [Bits and Bytes paper])
 - Large training data
 - Large in quality of data
- Thriving open-source community
- Highly optimized training
- Quantization for running on CPU (or less VRAM on GPU)
 - Use int4 instead of float16 datatypes
- Benchmarks are helpful but struggling to keep up


Open LLM Leaderboard

The Open LLM Leaderboard aims to track, rank and evaluate LLMs and chatbots as they are released.
Anyone from the community can submit a model for automated evaluation on the GPU cluster, as long as it is a Transformers model with weights on the Hub. We also support evaluation of models with delta-weights for non-commercial licensed models, such as LLaMA.

Search your model and press ENTER...

LLM Benchmark (lite) Extended view About

Model	Average	ARC (25-q)	HellaSwag (10-q)	MMLU (5-q)	TruthfulQA (MC) (0-q)
tiiuae/falcon-40b-instruct	63.2	61.6	84.4	54.1	52.5
tindettmrs/guanaco-65b-merged	62.2	60.2	84.6	52.7	51.3
CalderaAI/39B-Lezaxue	60.7	57.6	81.7	45.2	58.3
tiiuae/falcon-40b	60.4	61.9	85.3	52.7	41.7
tindettmrs/guanaco-33b-merged	60	58.2	83.5	48.5	50
auebosa/llama-30b-supercot	59.8	58.5	82.9	44.3	53.6
huddy/llama/llama-65b	58.3	57.8	84.2	48.8	42.3
pinkmanlove/llama-65b-hf	58.3	57.8	84.2	48.8	42.3
llama-65b	58.3	57.8	84.2	48.8	42.3
MetaIX/GPT4-X-Alpaca-30b	57.9	56.7	81.4	43.6	49.7
Anaja/VicUnlocked-alpaca-30b	57.6	55	88.8	44	50.4
digitous/Alpacino30b	57.4	57.1	82.6	46.1	43.8
Anaja/GPT4-x-AlpacaDante2-30b	57.2	56.1	79.8	44	49.1
TheBloke/Grovedary-65b-lora-HF	57	57.8	88.8	50.8	38.8
TheBloke/Wizard-Vicuna-13B-uncensored-HF	57	53.6	79.6	42.7	52
elinas/llama-30b-hf-transformers-4.29	56.9	57.1	82.6	45.7	42.3
auebosa/llama30b-SuperHOT	56.9	57.1	82.6	45.7	42.3
llama-30b	56.9	57.1	82.6	45.7	42.3
cyl/awesome-llama	56.8	54.4	79.7	41.8	51.3
NousResearch/Nous-Hermes-13b	56.4	52.5	88	41.8	51.1
elinas/chronos-33b	56.4	54.9	81.3	43.8	45.8
HuggingFaceM4/starchat-beta	56.2	52	88.6	44.7	47.3
openaccess-ai-collective/wizard-mega-13b	55.7	52.5	78.6	41	50.6



**LLMS AS PART
OF A SYSTEM**



FOUNDATION MODELS

[2012-2022]

- **Apply DL to task X**
- End-to-End (Given inputs, predict outputs)
- Specific to one task
- Adaptation via fine-tuning only
- Examples
 - BERT
 - CNNs
 - ...

[2022- now]

- Combine foundation/base models to X,Y,Z to do tasks Q,R,S
- Can do many tasks
- Adaption via prompting (no parameter updating)
- Powerful building blocks
- Examples
 - LLMs
 - CLIP
 - ...



LLMS AS PART OF A SYSTEM

- ChatGPT and Chat UIs are fun, but basic
- The real power of LLMs comes as part of a system
- What are the building blocks?

Building Blocks

- Memory
 - Long term
 - Short term
- Tools
 - Code interpreters
 - Shells
 - File system
 - API
- Database
- LLM
 - Tree of thought
 - Reflection

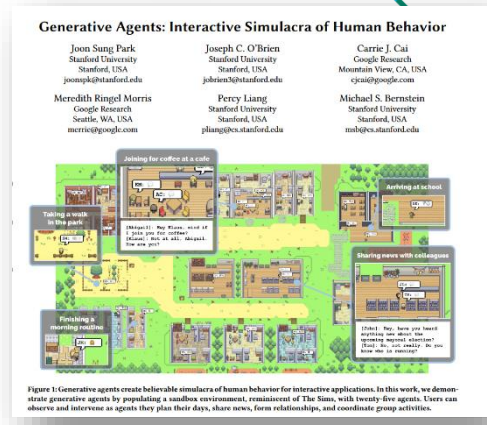
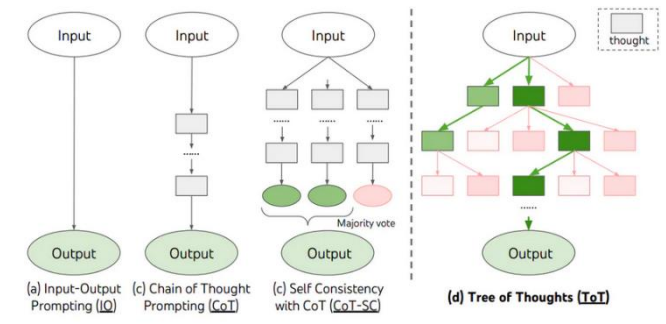
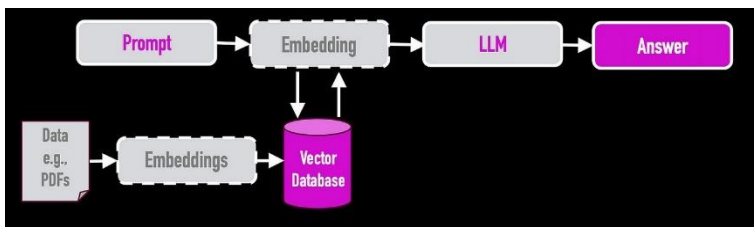
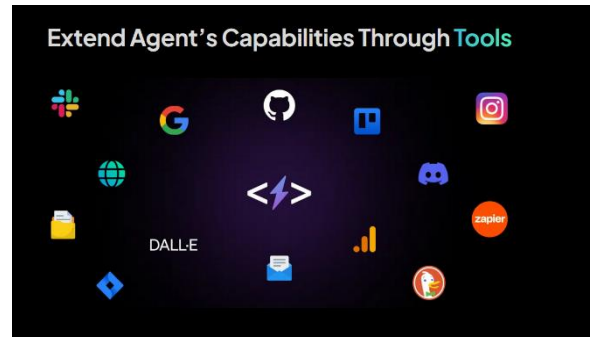
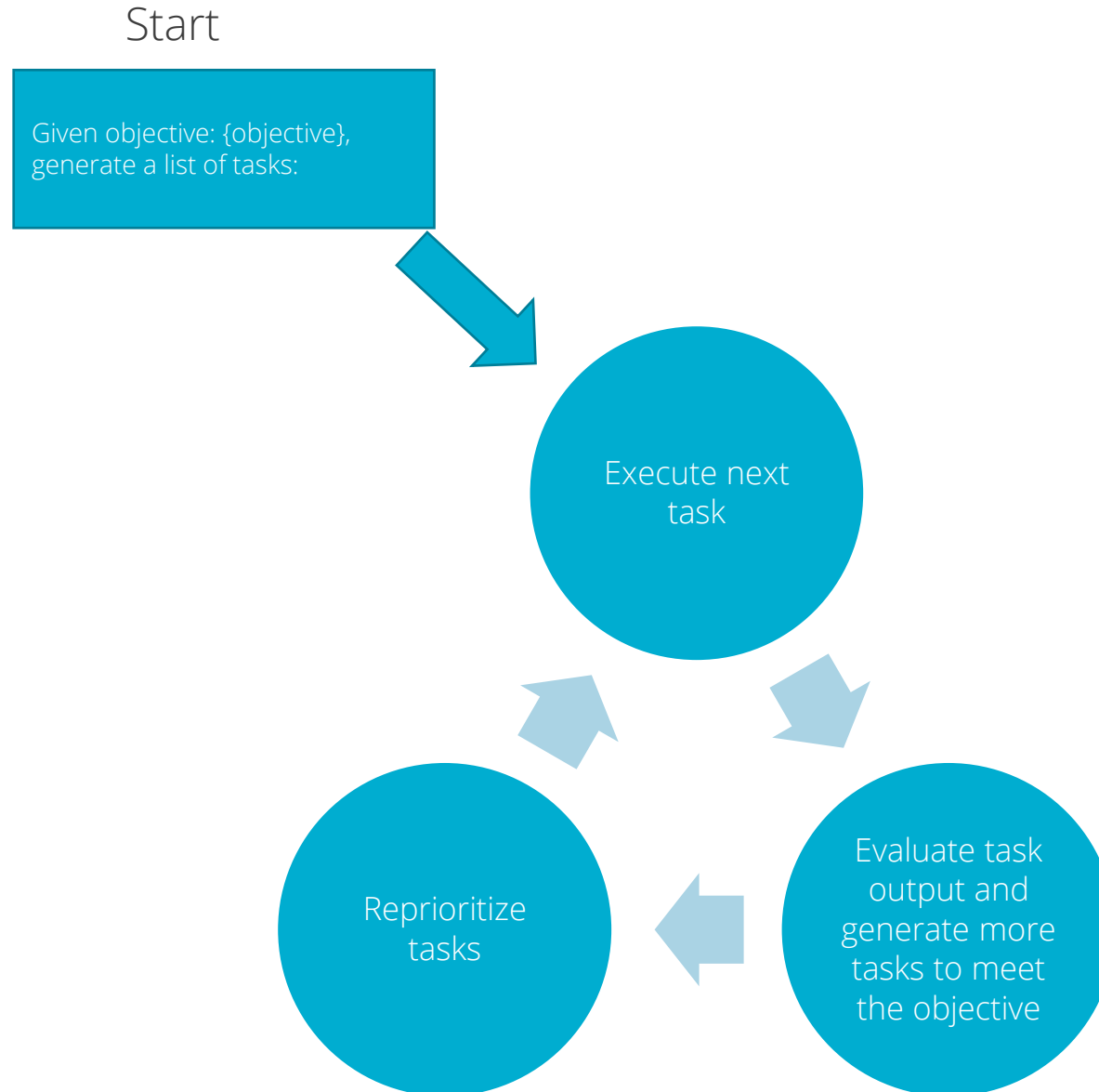


Figure 1: Generative agents create believable simulacra of human behavior for interactive applications. In this work, we demonstrate generative agents by populating a sandbox environment, reminiscent of The Sims, with twenty-five agents. Users can observe and intervene as agents they plan their days, share news, form relationships, and coordinate group activities.



AGENTS

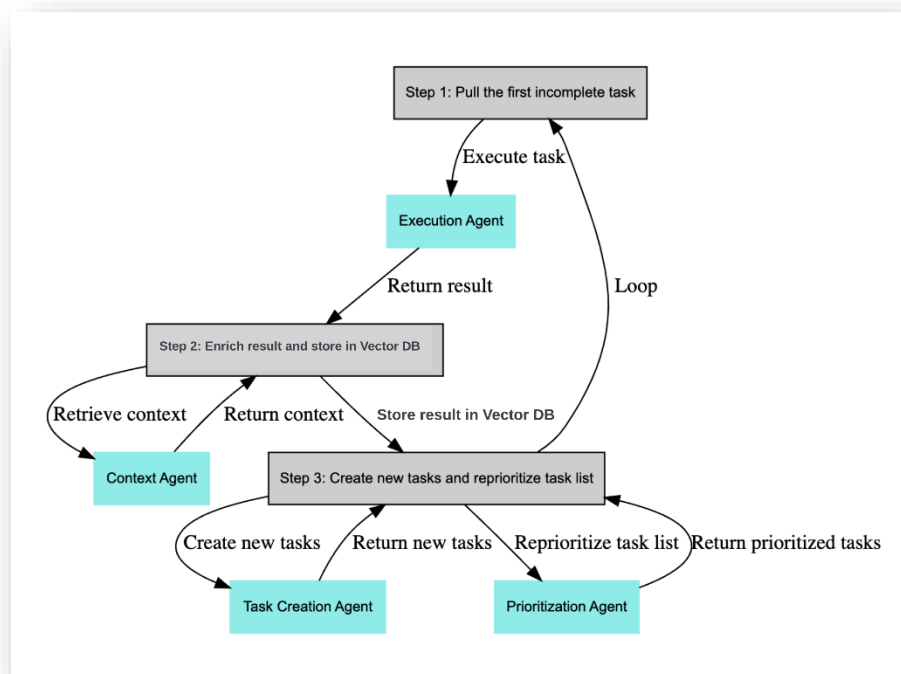




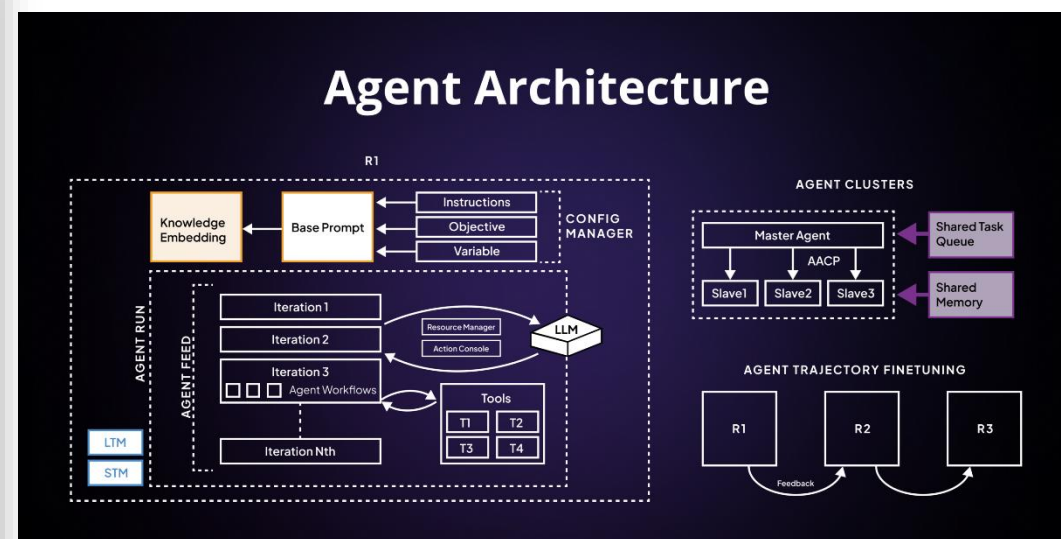
AGENTS

A few open source projects

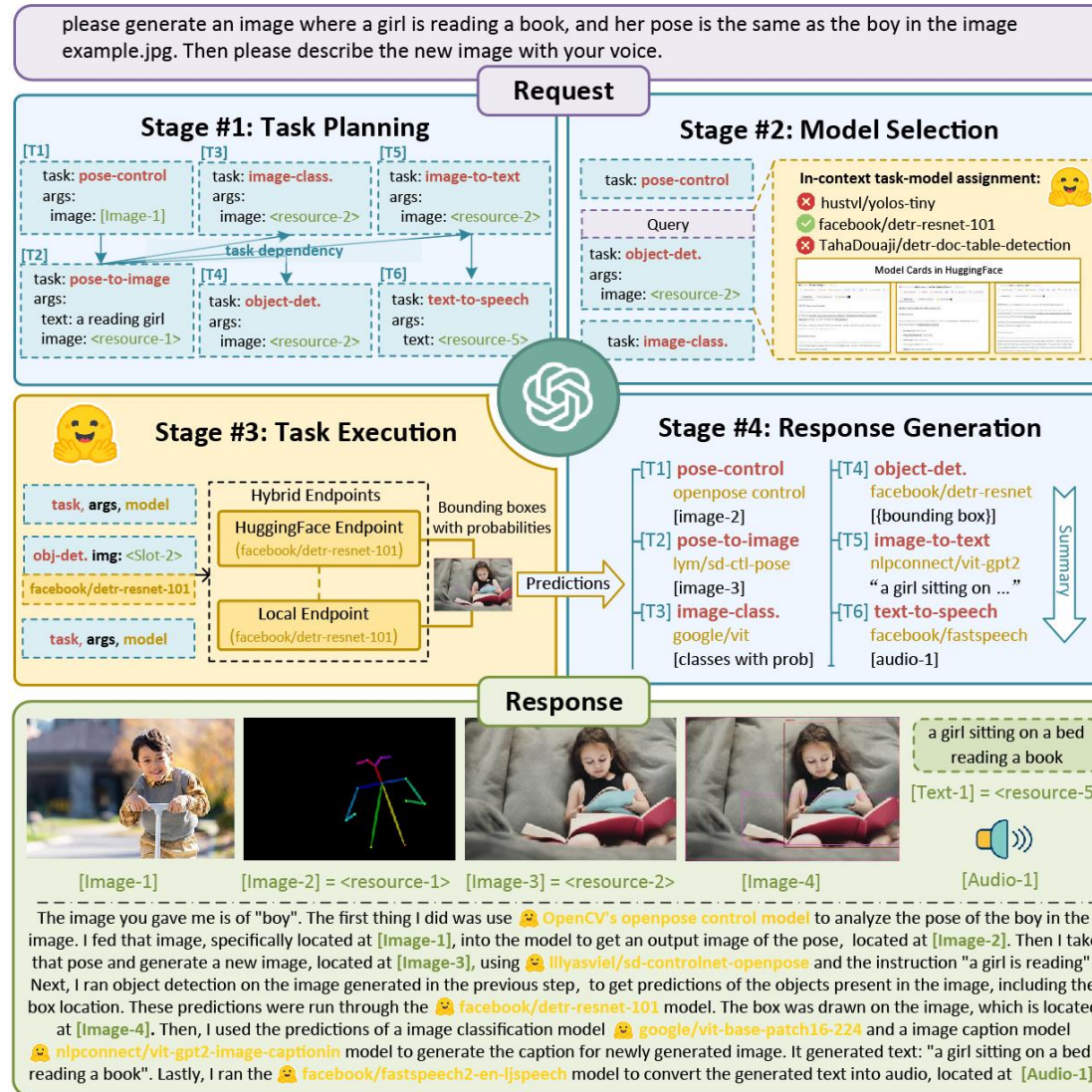
1. BabyAGI
2. AutoGPT
3. AgentGPT
4. SuperAGI
5. GPT-engineer
6. ...



Yoheinakajima BabyAGI



<https://github.com/TransformerOptimus/SuperAGI>





Write some code to make a multiplayer game in python

<https://agentgpt.reworkd.ai/>

<https://github.com/reworkd/AgentGPT>

The screenshot displays the AgentGPT interface with the following components:

- Task List:** A vertical list of tasks on the left side, including "Task Added: Define the gameplay mechanics and rules of the platformer game.", "Task Added: Design the levels and create the game assets (such as sprites and backgrounds).", "Task Added: Write the code for player movement, collision detection, and other gameplay features.", "Task Added: Test the game for bugs and make any necessary adjustments to ensure smooth gameplay.", "Starting task: Define the gameplay mechanics and rules of the platformer game.", and "Writing code...".
- Current tasks:** A panel on the right side showing the current tasks, including "Define the gameplay mechanics and rules of the platformer game.", "Design the levels and create the game assets (such as sprites and backgrounds).", "Write the code for player movement, collision detection, and other gameplay features.", and "Test the game for bugs and make any necessary adjustments to ensure smooth gameplay.".
- Code Editor:** A central window titled "Executing: Define the gameplay mechanics and rules of the platformer game." containing the following text:

Gameplay Mechanics and Rules
Overview
A platformer game is a type of video game where the player controls a character who can jump and navigate through a series of platforms and obstacles. The goal is usually to reach the end of the level or collect certain items. To define the gameplay mechanics and rules of the platformer game, we need to consider the following aspects:

 1. Character Movement
 2. Collision Detection
 3. Platform Generation
 4. Obstacle Generation
 5. Scoring System
 6. Level Progression

Let's define each of these aspects in detail.

1. Character Movement
The character in the platformer game should be able to move horizontally and jump. Here's the code for character movement:

```
Python Copy Code  
class Character:  
    def __init__(self):  
        self.x = 0 # character's x position  
        self.y = 0 # character's y position  
        self.velocity_x = 0 # horizontal velocity  
        self.velocity_y = 0 # vertical velocity  
  
    def move(self, delta_time):  
        self.x += self.velocity_x * delta_time
```
- Chat and Controls:** At the bottom, there is a "Chat with your agent..." input field, a "Custom task" button, an "Add" button, and three control buttons (stop, pause, play).



AgentGPT templates available now

ResearchGPT Academics and Professional Generate a thorough report on a specific subject	BrandGPT Academics and Professional Evaluate a brand's performance, market position, and future prospects	TravelGPT Other Plan a detailed journey to a selected destination	PlatformerGPT Creative and Social Code a platformer game featuring a popular character or theme
IndustryGPT Academics and Professional Present a comprehensive review of an industry, covering key trends, players, and future predictions	ScraperGPT Other Extract and summarize data from a selected website	PostGPT Creative and Social Create engaging captions and hashtags for your social media posts	EmailGPT Academics and Professional Compose a concise and detailed email
ResumeGPT Academics and Professional Design a professional resume based on your career history and skills	NovelGPT Creative and Social Begin writing a novel in a selected genre	DietGPT Health and Fitness Create a customized diet plan based on dietary preferences and goals	FitnessGPT Health and Fitness Design a workout regimen tailored to your fitness goals
MarketingGPT Academics and Professional Design a comprehensive marketing strategy for your business	BudgetGPT Academics and Professional Prepare a personal or family budget	StudyGPT Academics and Professional Design a study schedule to achieve your academic objectives	NewsGPT Other Author a detailed news article on a selected topic
EventPlannerGPT Other Organize a detailed schedule for your forthcoming event	BlogGPT Creative and Social Write a blog post on a selected topic	AstroGPT Science and Technology Discuss astronomical phenomena, discoveries, and related technology	ArtReviewGPT Creative and Social Critique a piece of art, discussing its style, context, and influence

SNL can make their own

EngineerGPT
Assist an engineer in designing a 3D object

ScienceGPT
Analyze this data and write a report

SierraGPT
Design a sierra input deck based on these requirements and 3d models

Sierra2GPT
Interpret the output of sierra to see if it meets these requirements

ContractGPT
Design a WPA with another agent to get job X done.

SummaryGPT
Given these historical documents and these new concerns, write a report



```
kovar_lpbf_search_results.txt - Auto-GPT [SSH: GPUdesktop] - Visual Studio Code
EXPLORER
  AUTO-GPT [SSH: GPUDESKTOP]
    .devcontainer
    .github
    auto_gpt_workspace
      <file_name>
      file_logger.txt
      kovar_lpbf_search_results.txt
    autogpt
    benchmark
    docs
    logs
    redis-data
    scripts
    tests
    .env
    .env.copy.template
    .env.template
    .envrc
    .flake8
    .gitignore
    .pre-commit-config.yaml
    .sourcery.yaml
    al_settings.yaml
    auto-gpt.json
    azure.yaml.template
    BULLETIN.md
    CODE_OF_CONDUCT.md
    CONTRIBUTING.md
    CURRENT_BULLETIN.md
    data_ingestion.py
    docker-compose.yml
    Dockerfile
    LICENSE
    main.py
    myreadme.md
    nvrproject.toml
    OUTLINE
    TIMELINE

  README.md 9+
  kovar_lpbf_search_results.txt
  Preview README.md

auto_gpt_workspace > kovar_lpbf_search_results.txt
1 LPBF printing of Kovar:
ls

PROBLEMS 345 OUTPUT PORTS DEBUG CONSOLE TERMINAL
bash - Auto-GPT + - - - - ^ X

mand to save important information to files to avoid exceeding my short term memory limit.\n- Analyze the information to determine the best approach for printing Kovar with LPBF.
CRITICISM: I need to make sure I am not relying too heavily on a single source of information and that I am analyzing the information critically to ensure its accuracy and relevance.
NEXT ACTION: COMMAND = do_nothing ARGUMENTS = {}
SYSTEM: Command do_nothing returned: No action performed.
THOUGHTS: I should start by reading through the search results to gather information about Kovar and LPBF printing. I can use the 'write_to_file' command to save important information to files and avoid exceeding my short term memory limit.
REASONING: Reading through the search results will help me gather information about Kovar and LPBF printing. Saving important information to files will help me avoid exceeding my short term memory limit and allow me to easily access the information later.
PLAN:
- Read through the search results to gather information about Kovar and LPBF printing.\n- Use the 'write_to_file' command to save important information to files to avoid exceeding my short term memory limit.\n- Analyze the information to determine the best approach for printing Kovar with LPBF.
CRITICISM: I need to make sure I am not relying too heavily on a single source of information and that I am analyzing the information critically to ensure its accuracy and relevance.
NEXT ACTION: COMMAND = write_to_file ARGUMENTS = {'file': 'kovar_lpbf_search_results.txt', 'text': 'Search Results for Kovar LPBF Printing:\n\n- The printing process of LPBF starts with the machine filling its chamber with inert gas then heating it to the required printing temperature. Then, you apply a layer of metal powder to the build platform. The software determines the thickness of the layer. LPBF printers have supports to reduce warping and distorting observed at high temperatures.\n\n- Much like the Texas A&M team, a collaborative of Chinese and U.S researchers has also found speed to be a vital variable in metal 3D printing. Through X-ray imaging, the scientists assessed the...\n\n- As you will have understood, LPBF uses metallic powders to design more or less complex parts. Metallic and alloy materials such as stainless steel, cobalt-chromium, aluminium, titanium and inconel are most often used. Some precious metals (gold, platinum, silver) are sometimes used, but almost exclusively in the jewellery sector.\n\n- The anniversary of the basic patent for LPBF marks an occasion for a video clip: 25 years ago, only one Fraunhofer ILT researcher was actively working on the project and the patent for metallic 3D laser printing was filed in 1996. This is an auspicious occasion for a retrospective and outlook on our technology," Jasmin Saewe says, pleased.\n\n- In-situ characterization of pore dynamics during LPBF process. a Schematic illustration of the in-situ high-speed X-ray imaging experiment. b Representative cuboid (300 μm × 200 μm × 200 μm ...)\n\n- Laser Powder Bed Fusion (LPBF) is one of the most well-known Additive Manufacturing processes. In LPBF, a laser beam selectively melts powder in a powder bed, thereby generating a cross-sectional area. A component is then built as several melting tracks are strung together in a micro-welding process.\n\n- Our LPBF machine prints steel parts with material properties comparable to wrought steel. The process uses inert gas, metallic powder, and a laser to melt layers into a solid, single metal part. Post-processing like annealing or heat treating is typically required. Advantages Production-grade metal components Complex shapes\n\n- Laser powder bed fusion (LPBF) is one of the most commonly used metal 3D printing processes. The process can deliver complex features, optimized geometries, lightweight parts and intricate designs. But, it's not as simple as just pressing print.'}
SYSTEM: Command write_to_file returned: Error: File has already been updated.
Thinking...
```

AutoGPT

Goal: Find process parameters for Kovar LPBF

GPT-ENGINEER

Prompt

Make a wrapper that can go around a gradio app to enable login with msal. Msal is a microsoft authentication python library.

If user isn't logged in, then redirect to login page.

Record the user in the session.

Modify the gradio ui to show the user's name and a logout button.

The gradio app should be just a simple interface that copies an image from the input to the output when a button is clicked.

Gradio uses fastapi

Don't use flask.

The user should be redirected to a login page which is separate from the main page.

Back and forth dialog
about specifications



```
projects > gradiowrapper > memory > logs > {} clarify > ...
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
{
  "role": "system",
  "content": "You will read instructions and not carry them out, only seek to clarify them.\nSpecifically you will first summarise a list of super short bullets of areas that need clarification.\nThen you will pick one clarifying question, and wait for an answer from the user.\n"
},
{
  "role": "user",
  "content": "Make a wrapper that can go around a gradio app to enable login with msal. Msal is a microsoft authentication python library. \nIf user isn't logged in, then redirect to login page.\nRecord the user in the session.\nModify the gradio ui to show the user's name and a logout button.\nThe gradio app should be just a simple interface that copies an image from the input to the output when a button is clicked. \nGradio uses fastapi \nDon't use flask. \nThe user should be redirected to a login page which is separate from the main page. "
},
{
  "role": "assistant",
  "content": "Areas that need clarification:\n1. How should the wrapper be implemented? Should it be a separate class or function?\n2. How should the redirect to the login page be handled? Should it be a separate route or a built-in feature of the wrapper?\n3. How should the user be recorded in the session? Should it be done automatically by the wrapper or does it require additional code?\n4. How should the user's name and logout button be displayed in the modified Gradio UI? Should it be a separate component or integrated into the existing UI?\n5. How should the login page be implemented? Should it be a separate route or a separate HTML file?\n6. How should the user be redirected back to the main page after successful login? Should it be automatic or require additional code?\n7. How should the image copying functionality be implemented in the Gradio app? Should it be a separate function or integrated into the existing code?\n8. How should the integration between Gradio and MSAL be done? Are there any specific methods or functions to use?\n9. Are there any specific requirements or limitations for using MSAL with Gradio and FastAPI?\n10. Are there any specific security considerations to keep in mind when implementing the login functionality with MSAL?"
},
{
  "role": "user",
  "content": "the wrapper should be a class. The login page should be a feature of the wrapper. The wrapper should record the session. The user's name and log out should be integrating into the existing UI. \n\nIs anything else unclear? If yes, only answer in the form:\n{remaining unclear areas} remaining questions.\n{Next question}\nIf everything is sufficiently clear, only answer \"Nothing more to clarify.\"."
},
{
  "role": "assistant",
  "content": "It is still unclear how the integration between Gradio and MSAL should be done. \n\nIntegration between Gradio and MSAL remaining question."
},
{
  "role": "user",
  "content": "MSAL can help with authentication. It will require a separate page for login, then redirect to microsoft for the auth, then back to a redirect url. \n\nIs anything else unclear? If yes, only answer in the form:\n{remaining unclear areas} remaining questions.\n{Next question}\nIf everything is sufficiently clear, only answer \"Nothing more to clarify.\"."
},
{
  "role": "assistant",
```


GPT-Engineer result = Multiple python files in OO programming.



The screenshot shows a Visual Studio Code workspace with the following structure:

- EXPLORER
 - GPT-ENGINEER
 - old
 - projects > gradiowrapper
 - __pycache__
 - memory
 - logs
 - clarify
 - execute_entrypoint
 - gen_clarified_code
 - gen_entrypoint
 - human_review
 - review
 - workspace
 - __pycache__
 - .gitignore
 - all_output.txt
 - config.py
 - gradio_app.py
 - main.py
 - msal_wrapper.py
 - README.md
 - run.sh
 - settings.toml
 - .secrets.toml
 - prompt
 - README.md

The main editor displays two Python files:

```
msal_wrapper.py
1 from typing import List
2 from fastapi import Request
3 from msal import ConfidentialClientApplication
4
5 class MSALWrapper:
6     def __init__(self, client_id: str, redirect_uri: str, scopes:
7         self.client_id = client_id
8         self.redirect_uri = redirect_uri
9         self.scopes = scopes
10        self.login_route = login_route
11        self.redirect_route = redirect_route
12        self.session = {}
13
14    def login(self) -> str:
15        authority = f"https://login.microsoftonline.com/{self.client_id}"
16        app = ConfidentialClientApplication(
17            self.client_id,
18            authority=authority,
19            redirect_uri=self.redirect_uri
20        )
21        login_url = app.get_authorization_request_url(
22            scopes=self.scopes,
23            redirect_uri=self.redirect_route
24        )
25        return login_url
26
27    def handle_redirect(self, request: Request) -> str:
28        code = request.query_params.get("code")
29        authority = f"https://login.microsoftonline.com/{self.client_id}"
30        app = ConfidentialClientApplication(
31            self.client_id,
32            authority=authority,
33            redirect_uri=self.redirect_uri
34        )
35        result = app.acquire_token_by_authorization_code(
36            code,
37            scopes=self.scopes,
38            redirect_uri=self.redirect_route
39        )
40        access_token = result.get("access_token")
41        self.session["access_token"] = access_token
42        return access_token
43
44    def get_user_name(self, access_token: str) -> str:
45        # Implement the logic to retrieve the user's name using the access token
46        # This can be done by making a request to the Microsoft Graph API
47        # and extracting the user's name from the response
48        pass
49
50    def logout(self) -> None:
51        self.session.clear()
52
```

```
main.py
1 from fastapi import FastAPI, Request
2 from msal_wrapper import MSALWrapper
3 from gradio_app import GradioApp
4
5 app = FastAPI()
6 msal_wrapper = MSALWrapper(
7     client_id="YOUR_CLIENT_ID",
8     redirect_uri="YOUR_REDIRECT_URI",
9     scopes=["openid", "profile", "email"],
10    login_route="/login",
11    redirect_route="/redirect"
12)
13 gradio_app = GradioApp(msal_wrapper)
14
15 @app.get("/")
16 def home(request: Request):
17     if "access_token" not in msal_wrapper.session:
18         return {"message": "Please login", "login_url": msal_wrapper.login()}
19     else:
20         access_token = msal_wrapper.session["access_token"]
21         user_name = msal_wrapper.get_user_name(access_token)
22         return {"message": f"Welcome, {user_name}!"}
23
24 @app.get("/login")
25 def login():
26     return {"login_url": msal_wrapper.login()}
27
28 @app.get("/redirect")
29 def redirect(request: Request):
30     access_token = msal_wrapper.handle_redirect(request)
31     return {"access_token": access_token}
32
33 @app.get("/logout")
34 def logout():
35     msal_wrapper.logout()
36     return {"message": "Logged out"}
37
38 if __name__ == "__main__":
39     gradio_app.run()
40
```

CONCLUSION

- Chatbots are cool, but aren't why LLMs are so powerful
- LLMs as a system are powerful
 - LLM
 - Controller
 - Synthesizer
 - Tools provide the system with more capabilities
 - Good systems are Self-correcting
- The exact setup of how to piece together different building blocks is an open question
- All the tools exist and open source right now!
 - Open source LLMs
 - Open source tool kits
- How to get started?
 - LangChain
 - LLamaIndex
 - AgentGPT
- Agents will only get better with time and research





**THANKS FOR
LISTENING**