

Hewlett Packard
Enterprise

A New Era of Software Development

Embracing Large Language Model Tools like ChatGPT for Iterative Problem Solving

- Andrew David Nieuwsma, Principal Cloud Developer at HPE
- Jeff Fougere, Innovation Strategist at HPE
- Jim Schreckengast, Strategist in HPE's Open Source Program Office

Today's Speakers



Andrew Nieuwsma

Principal Cloud Developer at
HPE



Jeff Fougere

Innovation Strategist at HPE



Jim Schreckengast

Strategist in HPE's Open Source
Program Office

Disclaimer: The perspectives and opinions of today's speakers are their own and may not align with those of Hewlett Packard Enterprise.



Will I lose my job?



Will AI Take my Job?

Probably not, but it will change it!

Replace

- Low skill information workers are at risk
 - Customer Support & Service
 - Call center & receptionist
 - Data Entry & Analysis
 - Accountants + HR

Augment (disrupt)

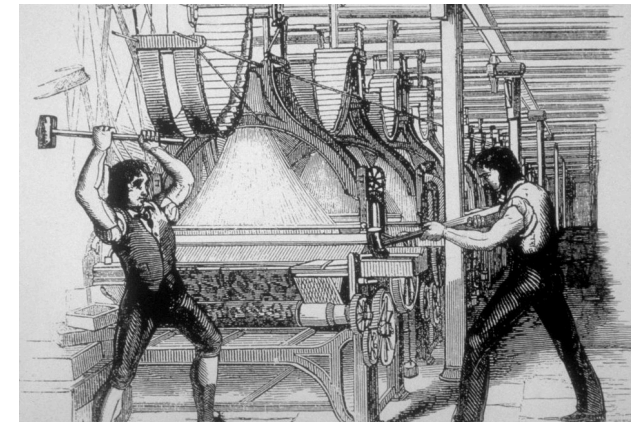
- Creatives (content, “influencers”, Marketing)
- Software
- Law (legal research, contract analysis)
- Medicine (medical imaging interpretation)
- Education & Learning



Andrew Nieuwsma

Principal Cloud Developer

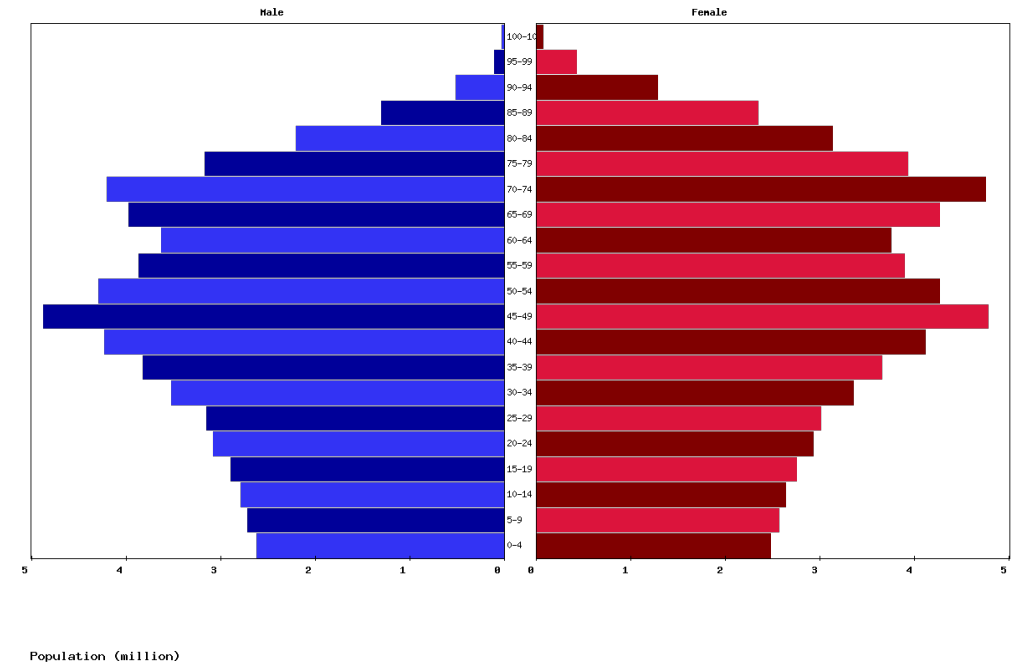
- Software Architect
- Leading Implementation Teams
- Coding as time allows
- Works on open-source project: Cray Systems Management



LUDDITES & THE FIRST CONTEST OF MAN VERSUS MACHINE

This isn't a bad thing

- Worker productivity has stagnated
- Baby Boomers are retiring, and global demographics are not on our side
 - ~Europe, China, Japan numbers don't look good!
 - Not enough workers in the pipeline to fill the jobs of the retiring generation



JAPAN DEMOGRAPHICS "PYRAMID"

[HTTPS://WWW.CATO.ORG/COMMENTARY/AI-COMING-OUR-JOBS-THATS-OKAY](https://www.cato.org/commentary/ai-coming-our-jobs-thats-okay)

[HTTPS://WWW.BLS.GOV/OPUB/MLR/2021/ARTICLE/THE-US-PRODUCTIVITY-SLOWDOWN-THE-ECONOMY-WIDE-AND-INDUSTRY-LEVEL-ANALYSIS.HTM](https://www.bls.gov/opub/mlr/2021/article/the-us-productivity-slowdown-the-economy-wide-and-industry-level-analysis.htm)

[HTTPS://ZEIHAN.COM/?S=DEMOGRAPHICS](https://zeihan.com/?s=DEMOGRAPHICS)



ChatGPT is a productivity booster ... *IF you know how to use it*

ChatGPT is like:

- having an infinite number of interns & assistants
- With deep knowledge about a wide range of topics
- That doesn't know when its wrong
- And doesn't know what you want to do and isn't a mind reader!

Tips for writing good ChatGPT prompts:

1. Ask ChatGPT to adopt a role
2. Ask it to prompt you
3. Be specific and provide as much detail as possible!
4. Give it feedback when its wrong or not what you are looking for
5. Keep the context

Start to think about ChatGPT differently.

- not a person – no moral agency or sentience
- not a mere calculator – its more a complex generative system
- It is an artificially intelligent dialogic interface



[HTTPS://CHAT.OPENAI.COM](https://chat.openai.com)



Use cases

Pro's & Con's



Use Cases - Coding

ChatGPT is GOOD at Coding, ChatGPT is BAD at Coding

- Architecture & Design
 - Doesn't know WHAT to do, only HOW
 - Won't replace good user stories!
 - *Can recommend patterns and approaches
- Code review
 - *Limited token size, so you can't dump huge files*
 - Can catch some code smells – not convinced better than static code analysis
- Code Generation
 - Tedious boiler plate code
 - Get/Set functions
 - Equality Functions
 - Simple stuff that takes high concentration but low creativity
 - Unit testing
 - Requires small function size (GOOD)
 - Effective at boundary case generation
 - Fast!
- Learning how to code
 - API / Library documentation is generally poor 😞
 - Too simplistic, not enough examples
 - Doesn't reveal WHY, WHEN, WHERE to use, only HOW (at best)
 - Break down concepts
 - Less noise than Google
 - More general than Stackoverflow
 - Enhances / augments documentation
 - Ex) Visualization - Matplotlib & Pandas



Use Cases – Coding Continued

- Documentation
 - “Here’s what my project does, help me write a succinct summary”
 - “Review my API specification and propose changes”
- Debugging
 - Repeatedly suggests wrong solution
 - Will propose debugging techniques
 - Break problem into smaller problem sets (decomposition)
 - “I’m getting this stack trace”
 - “what does ___ mean?”
- Pair Programming
 - Very good at:
 - “help me do X”
 - Small automations
 - scripting
 - polyglot
 - Hallucinates API function calls & language features
 - Really bad at converting from Python to GO!
 - It warned me it wasn’t a good idea!
 - Weakly typed -> Strongly Typed == Sorrow
 - Doesn’t write in my voice

Buyer beware!

Coding is building shared mental models
~25% of all software cost is code comprehension
(reading) [1]

Other Use Cases

- ChatGPT is great at writing
 - Critique your writing
 - *“Act like a college writing professor, review this essay”*
 - Edit writing
 - *“Help me edit this letter, what suggestions do you have for clarity or flow?”*
 - Suggest changes
 - *“Help me rewrite this in active voice”*
 - Write in a style (e.g., Shakespeare)
 - Synthesize writing
 - *“Write a 500-word blog post on why Cats are fun pets!”*
 - *“Write a love letter for my wife.”*
 - **DANGER this can backfire! (Southpark Season 26)**
- ChatGPT is useful in the research process
- Good at exploring boundaries and problem definition
 - *“What term describes self referential logical arguments?”*
 - *“What are some examples of ontological arguments?”*
 - *“What are some questionable argumentative methods like ontology?”*
 - *“How are circular reasoning and ontological arguments similar?”*
 - *“What are some books I should read about logical argumentation?”*
- Will not replace Lexus Nexus
- Will not help you confront your own confirmation bias
- Has problems with citations
- Creative meal planning
 - *“I have these ingredients and 20minutes”*



Limitations

- If you **don't know WHAT** to do it will gladly mislead you!
 - You can ask it for help if you have some idea of where to go
 - If you **don't know HOW** to do, it can help (generally)
 - ChatGPT is **WRONG** ... sometimes
 - Needs guidance, correction, and reinforcement
 - *Digital Dunning-Kruger Effect*
 - can get **lost in circles**
 - Hallucinates (**lies**)
 - Not a moral actor
 - Genuinely “believes” it’s telling the truth
 - **Subtle mistakes** are hard to identify (nuance)
 - There is **bias** in its model, don't let it think for you, let it help you see possibilities
 - only as good as its training data
 - Why? Because its based-on HUMAN history and our collective works!
 - read: *Weapons of Math Destruction* by Cathy O’Neil
 - Ex. Basing current mortgage risk calculations on historically redlined neighborhoods
 - discriminatory practice
 - systematically denying loans, insurance, or other financial services to certain neighborhoods or communities
 - based on their racial or ethnic composition
 - Prevalent in U.S. in 1930’s-1970’s
- "*THE TRUTH IS RARELY PURE AND NEVER SIMPLE.*" - OSCAR WILDE



(Some) Legal and Ethical Considerations

Disclaimer: This presentation does not constitute legal advice and does not establish an attorney-client relationship.



(Some) Legal Risks

Your country or company may ban it

Some countries are threatening to ban the use of Generative AI technology, including ChatGPT. Some companies have already banned it.

Confidentiality and data policies

You company may have policies that prevent the use of this technology for confidential data or other types of data (e.g., HIPAA compliance, Export control)

Intellectual Property and Terms of Use

Ambiguity regarding ownership of Intellectual Property and Terms of Use, which may be different across platforms.



(Some) Ethical Considerations: Just because you can . . . should you?

Feels “wrong”

- Controversial technology
- Will I become dependent?
- Shouldn't a human do this?

Risk of hallucinations

You should thoroughly check the results to ensure information is accurate—especially quotes and numbers!

Temptation

It will give you exactly what you ask for. You will be tempted to ignore counter-arguments and you may be overconfident in your work.



Citizen Developer example: The art of the possible

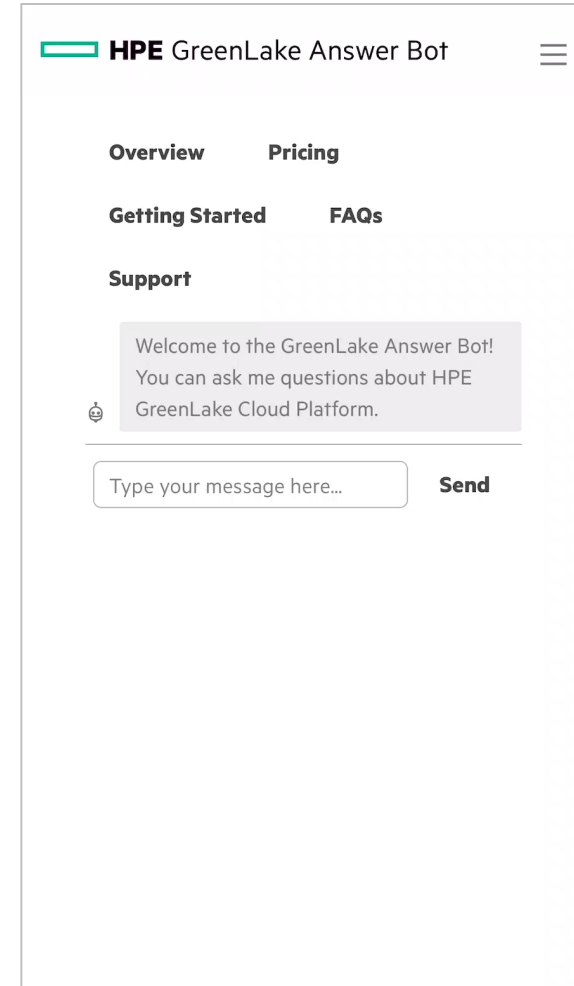
Warning: Be extremely careful with executing Generative AI-created code on your machine. Employ these techniques at your own risk.



Proof-of-Concept: HPE GreenLake Answer Bot

Success Criteria:

1. Created primarily via copy/paste
2. React Webapp
3. Leverages HPE “Design System”
4. Responsive Design
5. Connected to OpenAI’s API
6. Containerized via Docker
7. Deployed on Cloud Platform



Video Demo of Proof-of-Concept

Proof-of-Concept: Starting from Zero



Can you code react in pycharm? How do you begin



Yes, you can work with React in PyCharm, although it's not the primary use case for PyCharm, which is mainly a Python IDE. However, JetBrains (the company that develops PyCharm) also offers WebStorm, which is specifically tailored for web development, including React. But if you prefer to use PyCharm, you can still get the job done.



Here's how you can set up a React project in PyCharm:

1. Install Node.js and npm (Node.js package manager):

Download the latest version of Node.js from <https://nodejs.org/>. The installation will also include npm.

2. Create a new React project:

Open the terminal (command prompt) and navigate to the directory where you want to create your project. Then run the following command:

```
lua
```



Copy code

```
npx create-react-app my-app
```

Replace "my-app" with the name you want for your project. This command will create a new React project and install necessary dependencies.

3. Open the project in PyCharm:



Proof-of-Concept: Theming with a Design System



How do I import components from HPE's design system into react

< 1/2 >



HPE's design system is called Grommet, and you can easily import Grommet components into your React project. To do this, follow these steps:

1. Install Grommet and its peer dependencies:

Open your terminal (command prompt) and navigate to your project directory. Then run the following commands to install Grommet, its peer dependencies, and the required icons package:

```
npm install grommet grommet-icons styled-components
```

Copy code

2. Import Grommet components:

In your React component, you can now import the Grommet components you need. For example, let's say you want to use the ``Box`` and ``Button`` components from Grommet. You would import them like this:

```
javascript
```

Copy code

Proof-of-Concept: Creating chat webapp



write react code to create a web app that following HPE Design System guidelines, including appropriate page layout for a chatbot type experience in the page content area



To create a web app following the HPE Design System guidelines with a chatbot-like experience, you can use Grommet components and HPE-themed styling. Here's an example of a simple chatbot web app with a header and a chat window using React:



1. First, install Grommet and Grommet Icons:

bash

Copy code

```
npm install grommet grommet-icons styled-components
```

2. Create a new React component for the chatbot experience:

javascript

Copy code

```
// src/ChatBot.js
import React, { useState } from 'react';
import { Box, TextArea, Button, Text } from 'grommet';
```



Proof-of-Concept: Copying/Pasting Error Messages



im getting the following error:

Compiled with problems:

×

ERROR

[eslint]

src\App.js

Line 16:9: 'setMessages' is not defined no-undef

Line 17:16: 'messages' is not defined no-undef



Apologies for the confusion. It seems that I missed adding the `useState` import and the state definition in the example. Here's the corrected version:



1. Import `useState` from React in your `App.js` file:

javascript

Copy code

```
import React, { useState } from 'react';
```

2. Add the `messages` and `setMessages` state definition inside the `App` function:

javascript

Copy code

```
function App() {
```



Proof-of-Concept: Refining the Layout



the chat module is too wide. can you make it take up less screen space and have a margin area around it?



Of course! You can adjust the width of the chat module and add a margin around it by updating the `Chatbot.js` file as follows:



javascript

Copy code

```
// Chatbot.js
import React, { useState } from 'react';
import { Box, Heading, TextArea, Button, List, Text } from 'grommet';

const handleUserMessage = (event) => {
  // ...
};

const Chatbot = () => {
  const [messages, setMessages] = useState([]);

  return (
    <Box
      direction="column"
      pad="medium"
    >
```



Proof-of-Concept: Requesting insights



the side panel should have 5 buttons that would make sense on a side panel for this app.



Here's an updated version of the `App.js` file with a side panel containing five buttons:



```
javascript Copy code

import React, { useState } from 'react';
import { Box, Grommet, Header, Main, Sidebar, Text, Button } from 'grommet';
import { hpe } from 'grommet-theme-hpe';
import ChatBot from './ChatBot';
import { Hpe, Menu as HamburgerIcon, User, SettingsOption, Help, ChatOption,

const lightHPETheme = {
  ...hpe,
  global: {
    ...hpe.global,
    colors: {
      ...hpe.global.colors,
      background: '#FFFFFF',
    },
  },
};

function App() {
```

 **Profile**

 **Settings**

 **Help**

 **Chat History**

 **About**



Proof-of-Concept: Connecting to OpenAI's API



how do i connect this chatbot with OpenAI's API?



To connect your chatbot with OpenAI's API and leverage the power of GPT-3, you will need to make API requests to the OpenAI API endpoint. Here's a general outline of the steps involved:

1. Sign up for an API key: Visit the OpenAI website and sign up for an API key. You may need to join the waitlist or subscribe to a specific plan depending on availability.
2. Install the OpenAI Python package: Install the OpenAI Python package using pip or your preferred package manager. You can find the installation instructions on the OpenAI API documentation.
3. Make API requests: In your code, you can use the OpenAI Python package to make requests to the OpenAI API. You will need to provide your API key in the request.

Here's an example code snippet to give you an idea of how to integrate the OpenAI API with your chatbot:

python

 Copy code

```
import openai
```

```
# Set up your OpenAI API credentials
```

Proof-of-Concept: Deploying Containerized Webapp



What's the easiest way to host a docker container so I can run my app on my phone or any internet connected device (not locally or behind a firewall)



The easiest way to host a Docker container and make your app accessible from your phone or any internet-connected device is to use a cloud-based container hosting service. Many providers offer platforms to easily deploy and manage Docker containers. Some popular options include:



1. **Heroku** (<https://www.heroku.com>): Heroku is a user-friendly platform-as-a-service (PaaS) that supports Docker containers. To get started, you'll need a Heroku account, the Heroku CLI, and a Dockerfile in your app repository.
2. **Google Cloud Run** (<https://cloud.google.com/run>): Cloud Run is a managed compute platform that enables you to run stateless containers. To deploy a Docker container on Cloud Run, you'll need a Google Cloud account, the Google Cloud SDK, and a Dockerfile in your app repository.
3. **AWS Fargate** (<https://aws.amazon.com/fargate>): Fargate is a serverless compute engine for containers provided by Amazon Web Services (AWS). To use AWS Fargate, you'll need



Example Chat: Automated Code Review



Pretend you are a senior software architect and provide a critique of this webapp. Provide specific examples from the code



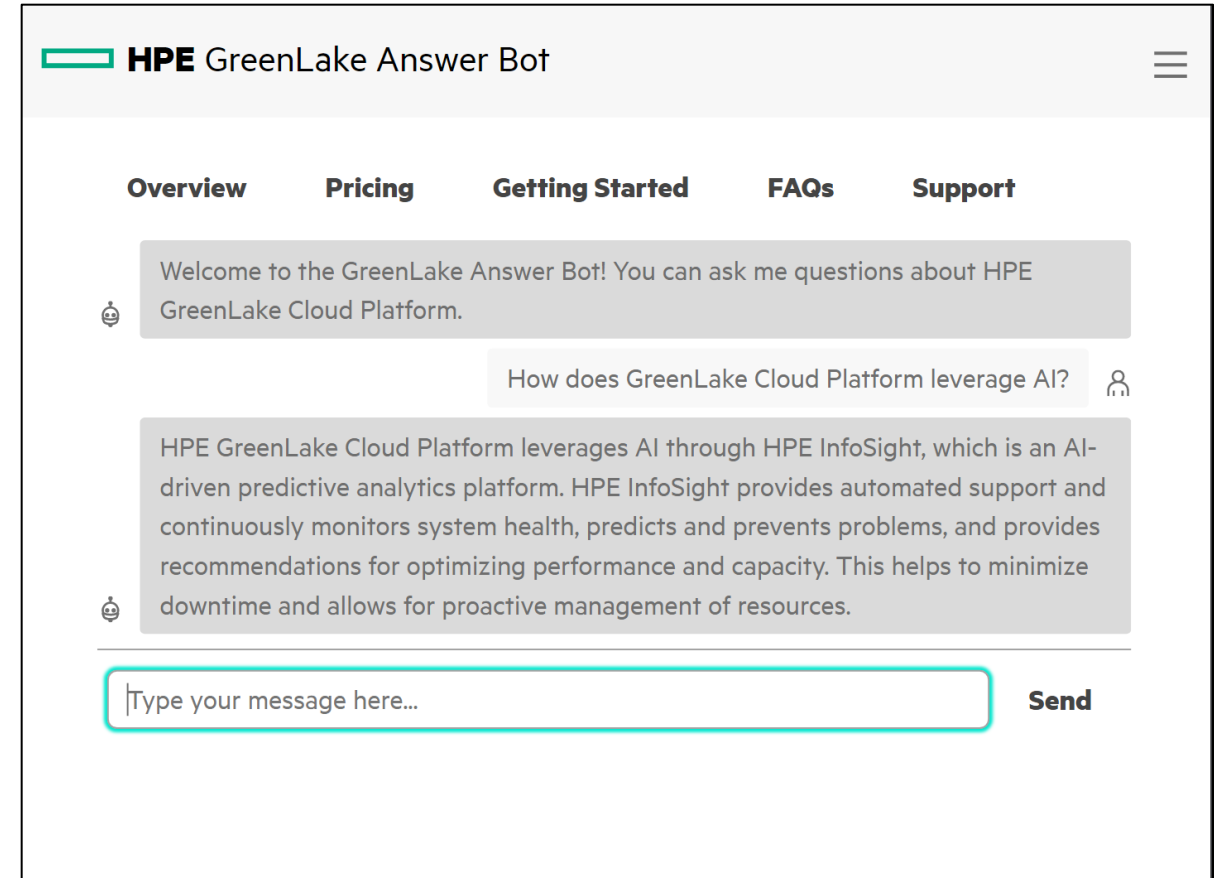
As a senior software architect, here's a critique of the web app, including specific examples from the code:

1. Lack of Code Organization: The code appears to be written in a single file (``App.js``), making it difficult to navigate and maintain. Consider dividing the code into separate modules or components based on functionality. For example, you could have separate components for the header, side panel, chat window, and chat input.
2. Improper Component Composition: The components are tightly coupled and lack proper separation of concerns. For example, the chat functionality and UI components are mixed within the ``App`` component. Consider breaking down the components into smaller, reusable ones to improve maintainability.



Takeaways

1. Feels like “pair programming” with AI
2. There are no dumb questions!
3. Be ambitious!
4. Embrace iteration
5. Modularization is your friend
6. Leverage human review processes



Beyond ChatGPT



Other generative AI tools for code generation

- AI2SQL
- AIXCoder
- AlphaCode*
- AskCodi
- CodeSquire
- CodeT5*
- Codiga
- Cody by SourceGraph*
- Deepcode (now Snyk Code)
- GitHub Copilot
- MutableAI
- OpenAI's Codex
- Polycoder*
- Ponicode* (acquired by CircleCI)
- Replit GhostWriter
- SantaCoder*
- StarCoder*
- TabNine
- WPCode
- ...

* Open Source

- ❑ There are many alternatives to ChatGPT for code generation.
- ❑ Many powerful contenders are open source
- ❑ Some of these models are purpose-built for a specific language or application.
- ❑ This space is evolving rapidly

Many solutions emerging for developers

- Numerous plug-ins for ChatGPT and other frameworks
- AI-driven documentation of code
- Test generation
- Security assessment and vulnerability mitigation generation
- PDFGPT – Ingest and query documents
- Video summarization and automated note-taking
- AI-driven GUI and experience generation
- Automation
 - AutoGPT – Agency and AI autonomy
 - AgentGPT

And these are early days...

The image displays two screenshots of AI-powered educational and development tools.

The top screenshot shows the YouLearn platform interface. It features a video player on the left showing a lecture titled "Lecture 1: Algorithmic Thinking: Peak Finding" by MIT OCW. The video shows a person writing on a chalkboard with the text "Divide & Conquer" and "Look at $n/2$ position". To the right of the video is a "Notes" section with a "Course Overview" and "Problem Sets" list. Below the video are two quiz sections: "Multiple Choice (1/10)" and "Short Answer (1/3)".

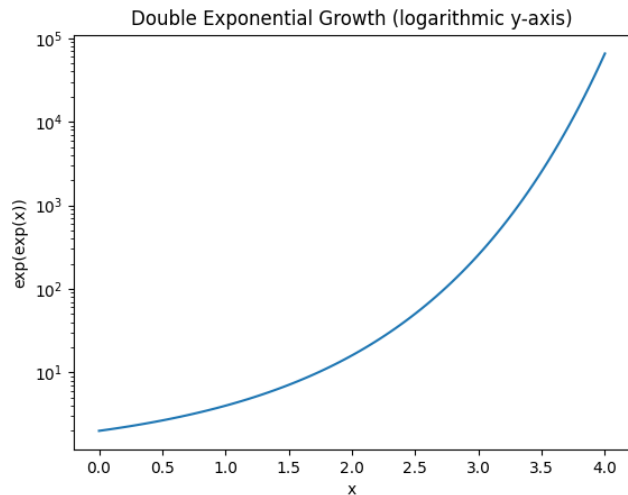
The bottom screenshot shows the AgentGPT interface. It features a central "My Agents" section with a list of agents, including "AgentGPT (1 / 10 runs)". To the right is a "Current tasks" section with a list of tasks, including "Search for open source AI projects on GitHub using relevant keywords related to software development." and "Compile a list of the top-ranked open source AI projects that are aimed at improving software development." The interface also includes a "Web Search" button, a "Pause Mode" button, and a "Custom task" input field.

Where is all this going?





Recent growth in AI is unprecedented.

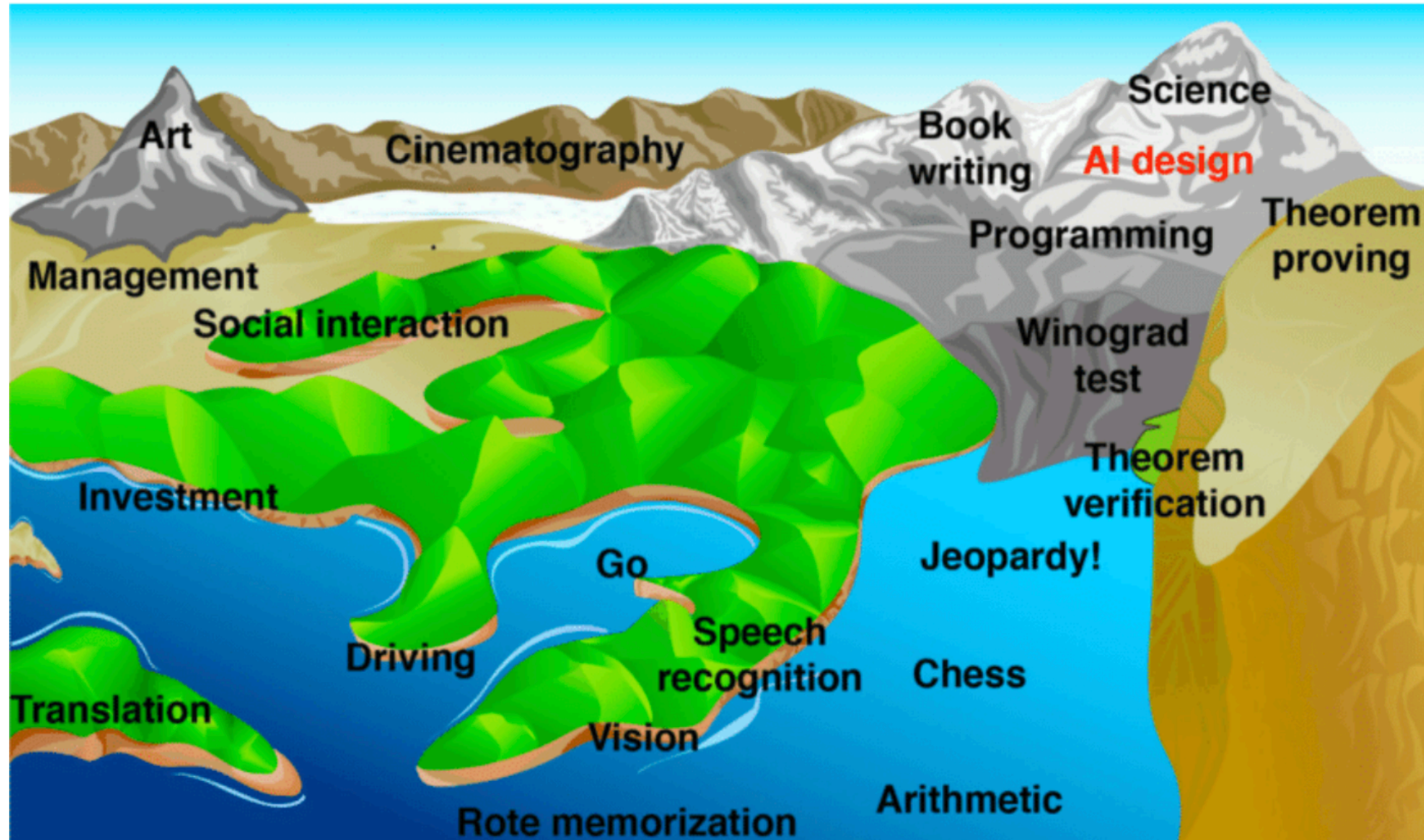


- ChatGPT reached 1M users in five days, 100M users in two months
- In March and April, 17 major LLM models were introduced
- Hugging Face, the open source collaboration space for AI models now has over 200,000 models in place, many of which rival proprietary LLMs.
- Capability growth has been astounding
 - Text-to-image models evolved from crude art generation to photo-realistic image generation in 14 months
 - OpenAI's GPT models' exam performance went from poor performance to top performance across many disciplines in less than a year.
 - Unpredicted emergent abilities of LLMs exceeded 137 by November of 2022

"ChatGPT, a version of OpenAI's GPT-3.5 [family]... gained more than 100m users in its first two months, and is now estimated to produce a volume of text every 14 days that is equivalent to all the printed works of humanity."

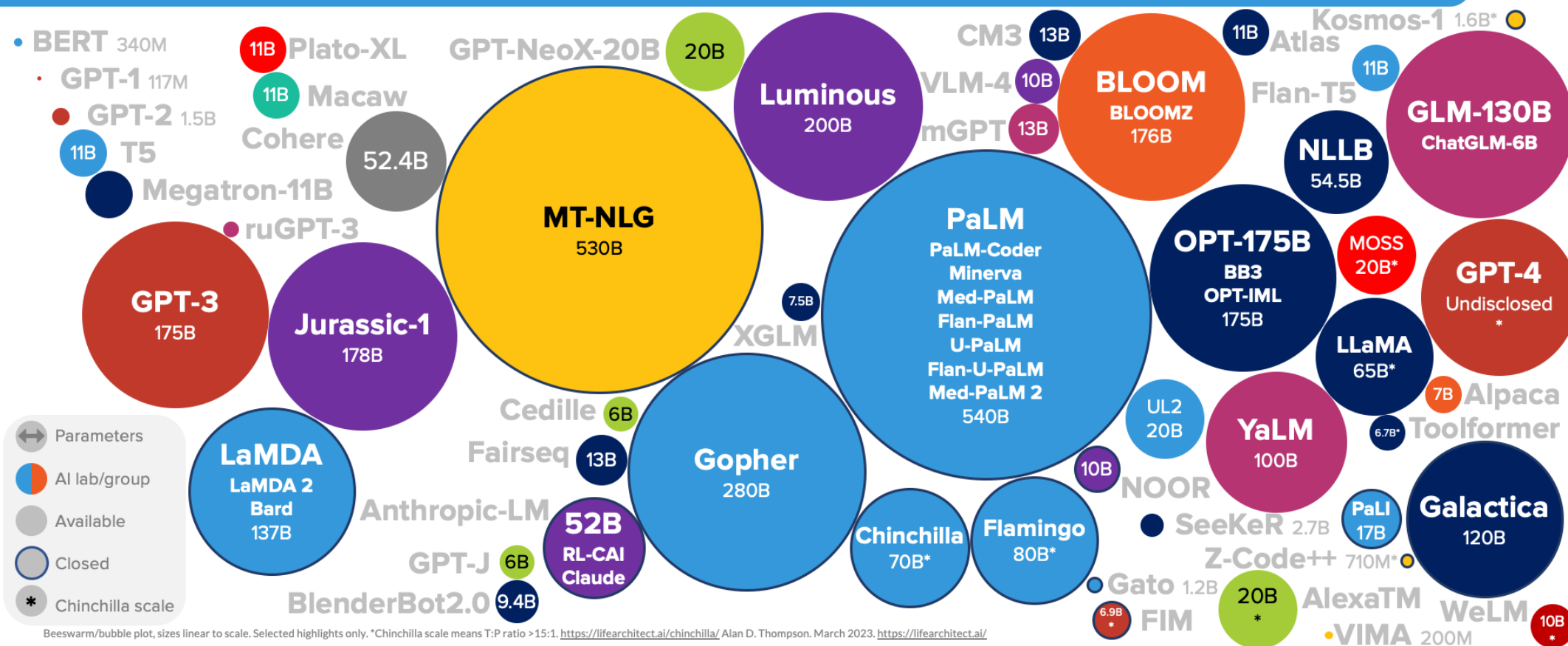
— Dr Thompson, Feb/2023, cited in report by the National Bureau of Economic Research (Scholes, Bernanke, MIT)

Predictions about the situation today were wrong



Hans Moravec's illustration of the rising tide of the AI capacity. From Max Tegmark (2017).

LANGUAGE MODEL SIZES TO MAR/2023



LifeArchitect.ai/models

ACHIEVEMENTS UNLOCKED BY LLMS

EMERGENT ABILITIES OF
LARGE LANGUAGE MODELS (APR/2023)

S

GPT-3 13B,
PaLM 8B



Mod.Arithmetic*



Debugging*



Comprehension*

M

GPT-3 175B,
LaMDA 137B,
PaLM 64B,
Chinchilla 7B



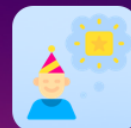
LinguisticsPuzzles*



EmojiMovie*



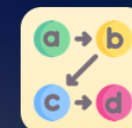
GRE-Comprehension*



MetaphorUnderstanding*



PhysicalIntuition*



LogicalDeduction*

L

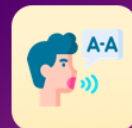
PaLM 540B,
Chinchilla 70B



GeometricShapes*



Proverbs*



PhoneticAlphabet*



ElementaryMath*



CausalJudgment*



CodeLineDescription*

XL

GPT-4,
Gemini (est.)



College-LevelExams



Self-Critique/Reflection



AppBuilding



SpatialReasoning



AdvancedCreativity



EmbodimentOptions

Next...



Grounding



Long-HorizonPlanning



Persuasion



AdvancedEmbodiment



Awareness



More...

Selected highlights only. Sources: Original papers and Dr. Jason Wei's summary: <https://www.jasonwei.net/blog/emergence>. BIG-bench = * Using images from FlatIcon.com. Alan D. Thompson. April 2023. <https://life architect.ai>



LifeArchitect.ai/models

Burning questions

- **Will Open Source be the dominant engine for advancement in this area?**
- **When will AGI be achieved, and by whom?**
- **How far out is superintelligence?**
- **Will AI lead to our extinction, or will it be the greatest achievement for advancing humanity?**



Recommendations for developers

- Try to keep abreast of the latest developments weekly
- Use Generative AI tools to augment your brain and effectiveness
 - Learning
 - Digesting and applying knowledge quickly
 - Synthesizing ideas
 - Accelerating code development
 - Review and improve your work
 - Stimulate creativity and innovation
 - Automating research and work
- Always be the “responsible adult” when applying these technologies
- Use the APIs to personalize your “AI edge” as a developer



DISCUSSION

