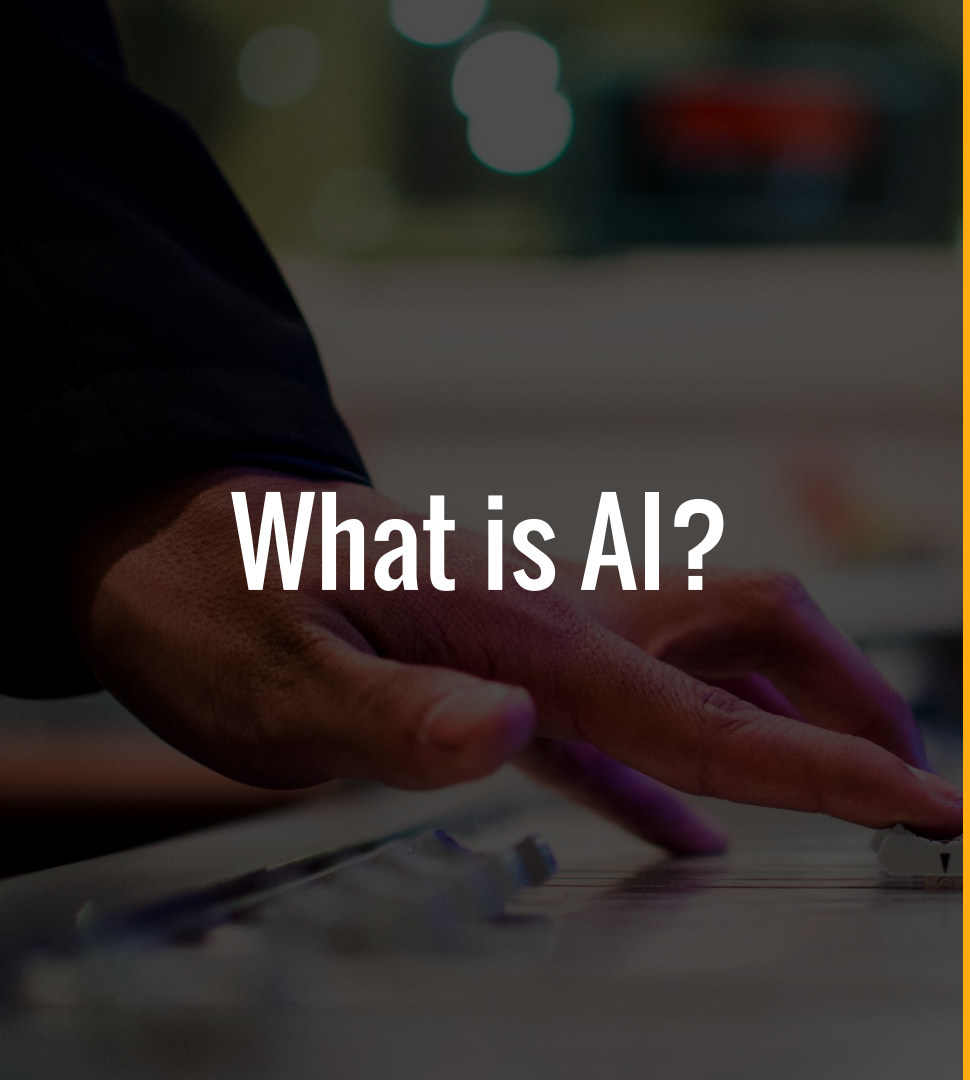
Unleash the Power of AI: Craft Your Magic with ChatGPT API!



What is AI?

"Artificial Intelligence" (AI) covers a wide range of technologies and approaches to create machines capable of performing tasks that would typically require human intelligence.

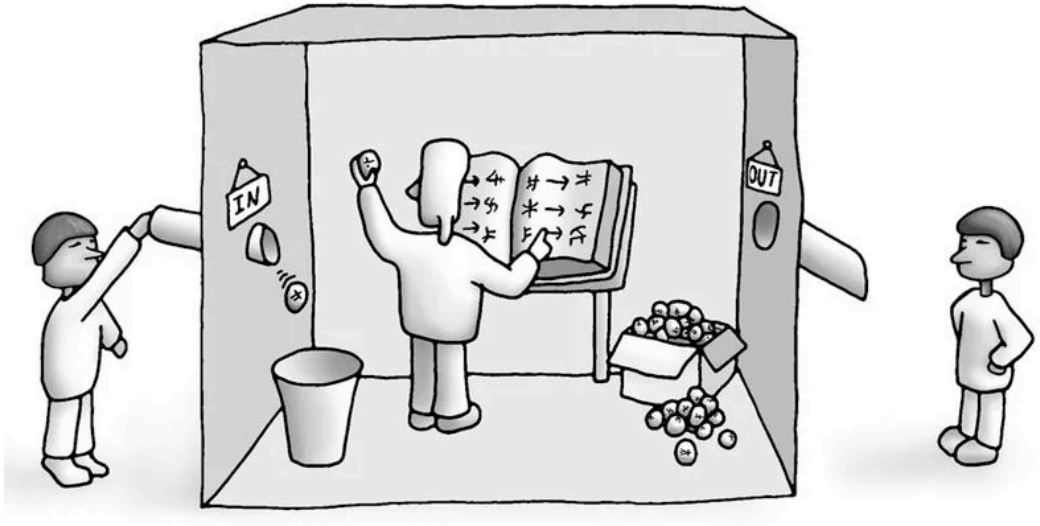
What Is ChatGPT Doing … and Why Does It Work?

By Stephen Wolfram

The Chinese Room Thought Experiment

Implications for Developers

- Syntax vs. Semantics
- Limitations of Strong AI
- Ethical Considerations
- Focus on Narrow AI
- Human-AI Collaboration
- User Experience



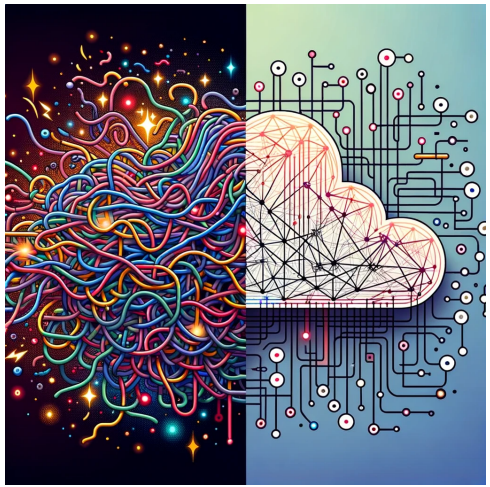
What Sets ChatGPT Apart?

- **As simple as a REST call**: ChatGPT's accessibility is one of its most significant features. Developers can easily integrate it into applications using a standard REST API call, eliminating the need for complex setups.
- **Reduced technical challenges**: Unlike traditional AI setups that require vast computational resources, intricate training processes, and specialized knowledge, ChatGPT, being a pre-trained model, removes these barriers.
- **Conceptual challenges persist**: While the technical barriers are lowered, understanding how to utilize ChatGPT effectively, tailoring prompts, and handling its outputs remains a nuanced task, necessitating a deep understanding of its workings.

What Sets ChatGPT Apart?

As simple as a REST call

Reduced technical challenges



Conceptual challenges persist

Important Concepts

- **Token**: A unit of text that the model reads. In English, a token can be as short as a single character or as long as a word. For instance, "ChatGPT is amazing!" would be split into six tokens: ["ChatGPT", " is", " amazing", "!", " “].
- **Temperature:** A parameter that controls the randomness of the model's output. A higher value (e.g., 1.0) makes the output more random, while a lower value (e.g., 0.2) makes it more deterministic.
- **Query**:The input or prompt given to the model to elicit a response. It serves as a direction or instruction for what kind of answer or output is desired.
    - **Roles - System vs User**: Within a conversation with the model, the "System" role provides instructions on how the model should behave, while the "User" role is typically where questions or prompts are provided.

The Code ...

https://chat.martinrojas.dev/
https://github.com/martinrojas/chatbot-playground

REST Endpoint - Receive and validate request

```
import { OPENAI_KEY } from '$env/static/private';
import type { RequestHandler } from './$types';
import { getTokens } from '$lib/tokenizer';
import { json } from '@sveltejs/kit';
      You, last month • Initial Commit
export const POST: RequestHandler = async ({ request }) => {
  try {
    if (!OPENAI_KEY) throw new Error('No OpenAI key found. Set OPENAI_KEY environment variable.');

    const requestData = await request.json();

    if (!requestData)
      throw new Error('No request data found. Request data: ' + JSON.stringify(request));

    const reqSystemPrompts: ChatCompletionRequestMessage[] = requestData.system;
    const reqMessage: ChatCompletionRequestMessage[] = requestData.messages;

    if (!reqSystemPrompts) throw new Error('No system prompts found in the request data.');
    if (!reqMessage) throw new Error('No request message found in the request data.');
```

REST Endpoint - get token count

```
let tokenCount = 0;

reqSystemPrompts.forEach((prompt) => {
  const tokens = getTokens(prompt.content || '');
  tokenCount += tokens;
});

reqMessage.forEach((message) => {
  const tokens = getTokens(message.content || '');      You, last month • Initial Commit
  tokenCount += tokens;
});

if (tokenCount >= 4000) throw new Error('Query too large. Token count: ' + tokenCount);
```

REST Endpoint - Ensure that there is no dangerous content

```
const moderationRes = await fetch('https://api.openai.com/v1/moderations', {
  headers: {
    'content-type': 'application/json',
    Authorization: `Bearer ${OPENAI_KEY}`
  },
  method: 'POST',
  body: JSON.stringify({
    input: reqMessage[reqMessage.length - 1].content
  })
});
const moderationData = await moderationRes.json();
const [moderationResults] = moderationData.results;

if (moderationResults.flagged) throw new Error('Message flagged by OpenAI');
```

REST Endpoint - create request to the OpenAI API

```
const messages: ChatCompletionRequestMessage[] = [...reqSystemPrompts, ...reqMessage];

const chatRequestOpts: CreateChatCompletionRequest = {
  model: 'gpt-3.5-turbo-16k',
  messages,
  temperature: requestData.temperature / 100 || 0.9,
  stream: true
};      You, last month • Initial Commit

const chatResponse = await fetch('https://api.openai.com/v1/chat/completions', {
  headers: {
    'content-type': 'application/json',
    Authorization: `Bearer ${OPENAI_KEY}`
  },
  method: 'POST',
  body: JSON.stringify(chatRequestOpts)
});
```

REST Endpoint - Handle the response

```
    You, last month • Initial commit
  if (!chatResponse.ok) {
    const err = await chatResponse.json();
    throw new Error('Chat completion request error. Error: ' + JSON.stringify(err));
  }

  return new Response(chatResponse.body, {
    headers: {
      'content-type': 'text/event-stream'
    }
  });
} catch (error) {
  let message = 'Unknown Error';
  if (error instanceof Error) message = error.message;
  console.error(message);
```

The Code ...

https://chat.martinrojas.dev/
https://github.com/martinrojas/chatbot-playground

The Query - Prompt Engineering

Prompt engineering is critical in manipulating ChatGPT's outputs. It's the Wild West out there in terms of exploring the endless possibilities with this tool!

https://learnprompting.org/docs/intro

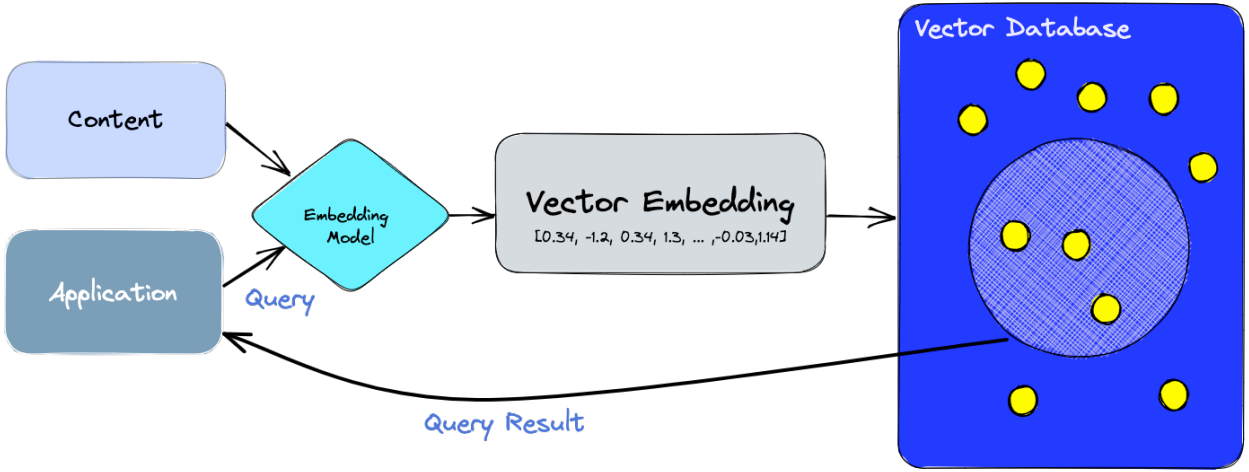
https://www.deeplearning.ai/short-courses/chatgpt-prompt-engineering-for-developers/

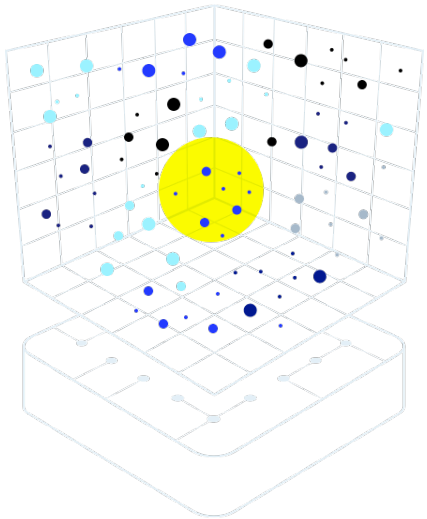
Deep Dive: Fine-tuning & Semantic Searching

1. **Fine-tuning:** The process of training a pre-existing model on a smaller, specific dataset to adapt it for a particular task. Fine-tuning allows developers to leverage a large, generalized model (like ChatGPT) and tailor it to specific applications, ensuring that it performs optimally in specialized scenarios.

2. **Semantic Searching:** Unlike traditional keyword-based searching, semantic searching understands the context and intent behind a query. It looks for the meaning in the search query rather than just matching keywords, enabling more relevant and nuanced search results.

Semantic Search (Vector Search)





Martin Rojas

@martinrojas
Bsky @martinrojas.dev
Mastodon @nextstepsdev@mozilla.social
https://nextsteps.dev/

REACT ATL