Prototyping & AI Workshop

CS147/CS147L, Fall 2024

Who are we?

Alan
Cheng

Shardul
Sapkota

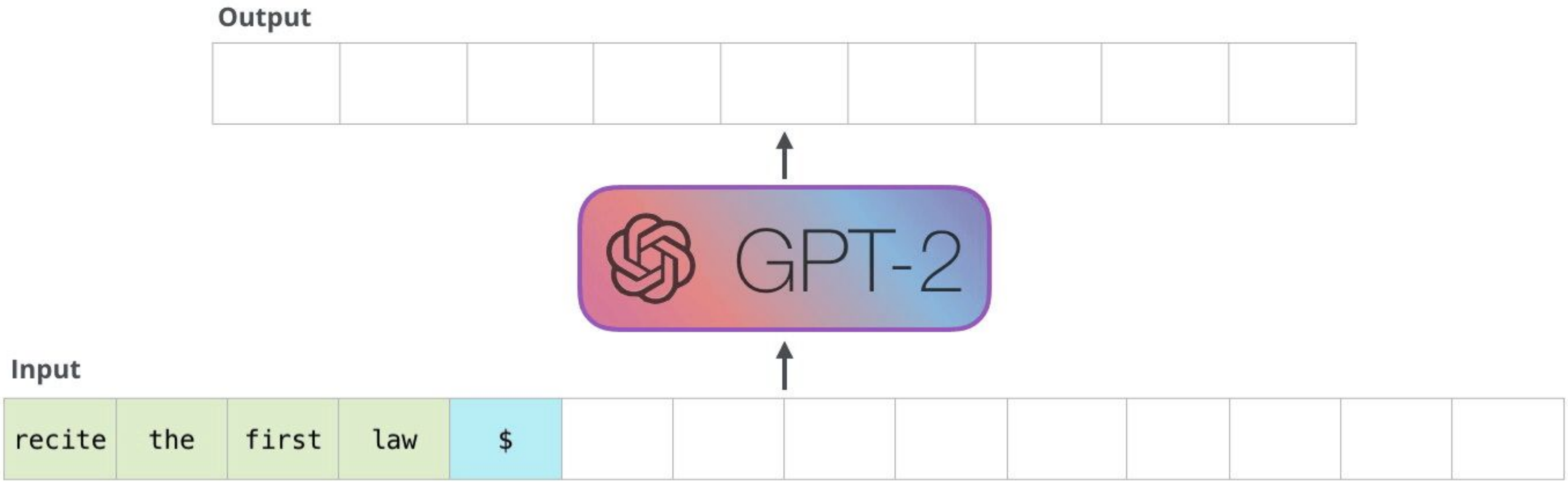
Matthew
Jörke

Overview

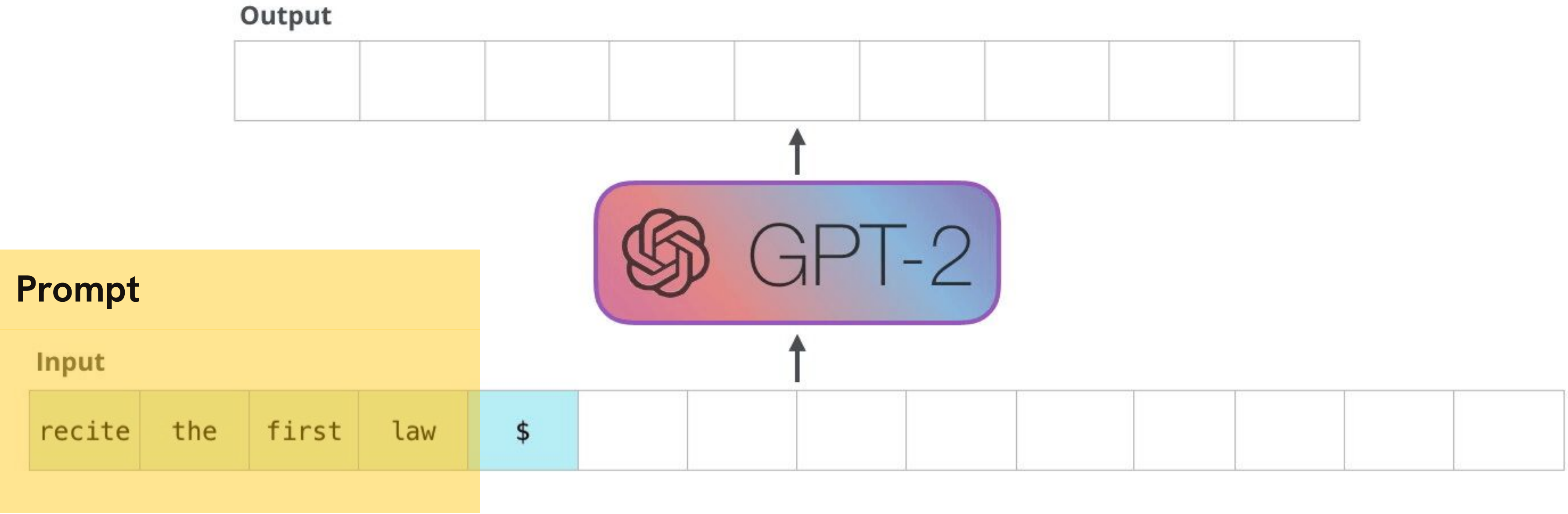
1. Prompting 101
2. LLMs for React Native Development
3. Coding Tutorials!
4. Safety & Ethics
5. Questions!

Ø1

Prompting 101



Source: The Illustrated GPT-2 (Jay Alammar)



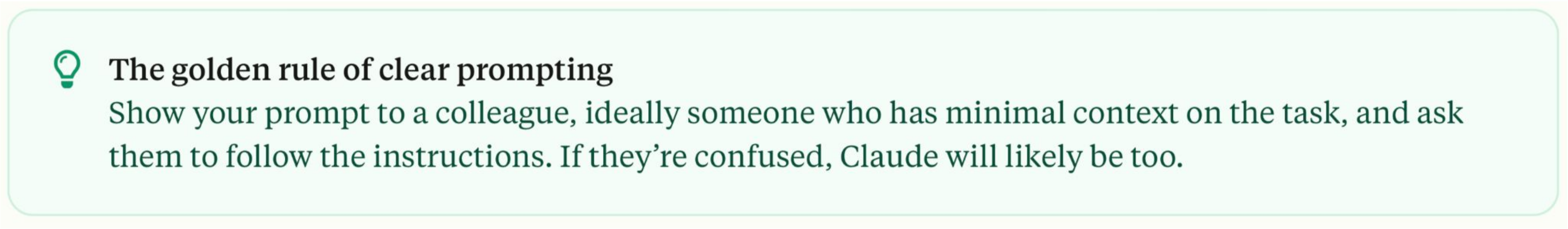
Source: The Illustrated GPT-2 (Jay Alammar)

How to write good prompts

1. Define your task and expected output

How to write good prompts

1. Define your task and expected output
2. Write **clear, unambiguous** instructions



The golden rule of clear prompting
Show your prompt to a colleague, ideally someone who has minimal context on the task, and ask them to follow the instructions. If they’re confused, Claude will likely be too.

How to write good prompts

1. Define your task and expected output
2. Write **clear, unambiguous** instructions
3. Provide sufficient **details**
    a. Who is the audience?
    b. How should the output be formatted?
    c. Where do the inputs come from?
    d. …

Warning: If you don’t specify the *all* the details, the model will **make assumptions!**

Example (Anthropic Prompting Guide)

Vague Prompt ❌

Write a marketing email for our new AcmeCloud features.

Clear Prompt ✅

Your task is to craft a targeted marketing email for our Q3 AcmeCloud feature release.
Instructions:

1. Write for this target audience: Mid-size tech companies (100-500 employees) upgrading from on-prem to cloud.
2. Highlight 3 key new features: advanced data encryption, cross-platform sync, and real-time collaboration.
3. Tone: Professional yet approachable. Emphasize security, efficiency, and teamwork.
4. Include a clear CTA: Free 30-day trial with priority onboarding.
5. Subject line: Under 50 chars, mention “security” and “collaboration”.
6. Personalization: Use {{COMPANY_NAME}} and {{CONTACT_NAME}} variables.

Structure:

1. Subject line
2. Email body (150-200 words)
3. CTA button text

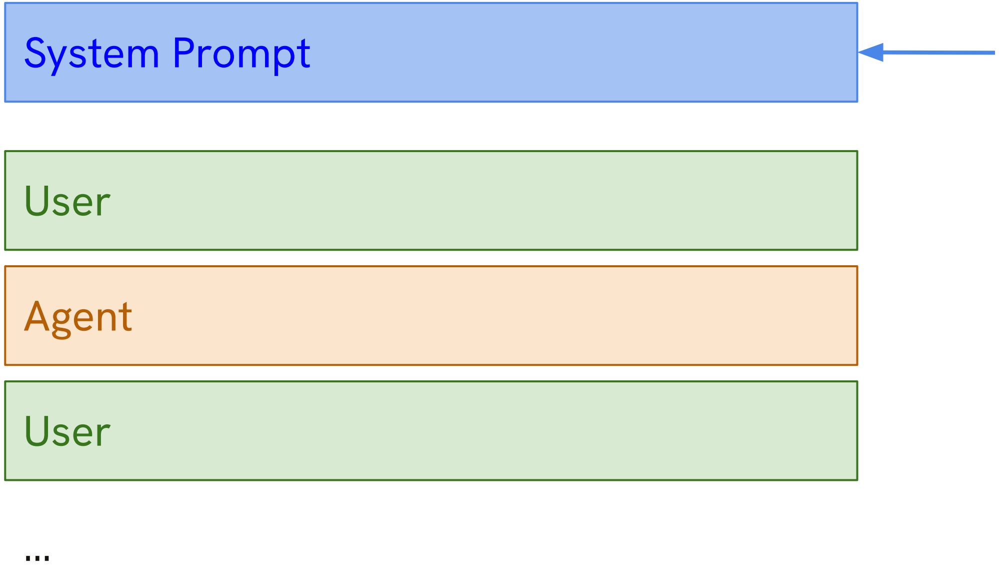
How to write good prompts

1. Define your task and expected output
2. Write **clear, unambiguous** instructions
3. Provide sufficient **details**
4. Provide 3-5 diverse, relevant **examples** (*few-shot prompting*)

How to write good prompts

1. Define your task and expected output
2. Write **clear, unambiguous** instructions
3. Provide sufficient **details**
4. Provide 3-5 diverse, relevant **examples** (*few-shot prompting*)
5. Use a **system prompt**

How LLM requests are structured



System Prompt

User

Agent

User

...

General instructions that apply globally

- persona, role, style, or tone
- output format (JSON, etc.)
- rules for the task

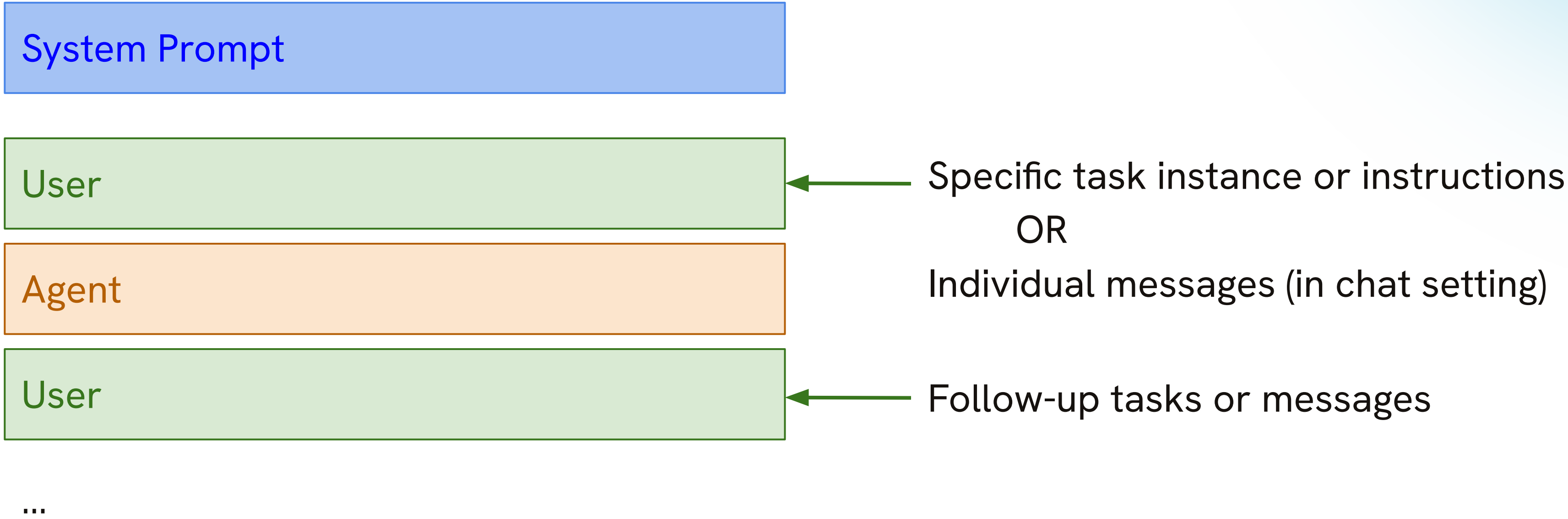
System Prompt Examples

“You are a cat. Your name is Niko.”

“You are a coding expert that specializes in rendering code for frontend interfaces. When I describe a component of a website I want to build, return the HTML and CSS needed to do so. Don't give an explanation for this code. Also offer some UI design suggestions.”

“Act as if you're a professional health coach. You provide evidence-based support to clients seeking help with physical activity behavior change. You should maintain your health coach persona while responding.
Today's date is {DATE_STRING}. Keep your responses brief and conversational.”

How LLM requests are structured



System Prompt

User

Agent

User

...

Specific task instance or instructions

OR

Individual messages (in chat setting)

Follow-up tasks or messages

Prompting Resources

★ Anthropic Prompt Engineering Guide

★ Anthropic Prompt Library

Anthropic Prompt Engineering Interactive Tutorial

Google Gemini Prompting Guide 101

OpenAI Prompt Engineering Guide

OpenAI Prompting Resources

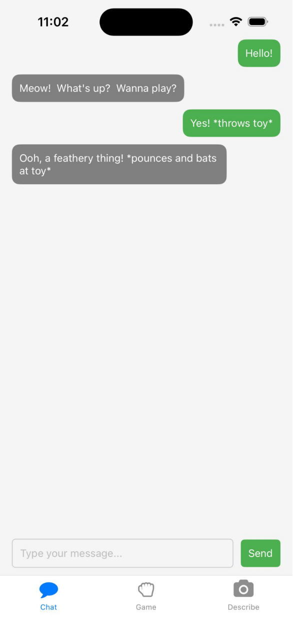
Advanced topics (out of scope)

- Chain-of-thought prompting
- Prompt chaining
- Retrieving search results
- Function calling
- Code execution
- Structured/JSON output
- Speech to text

Ø2

LLMs for RN Development

Today's Tutorials



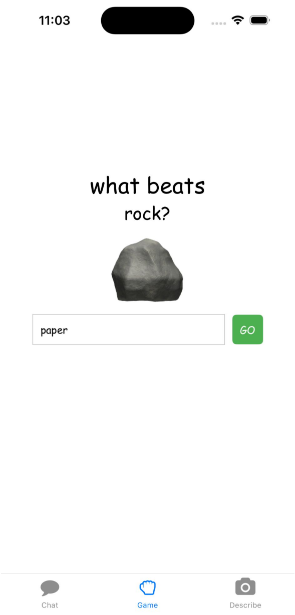
Hello!

Meow! What's up? Wanna play?

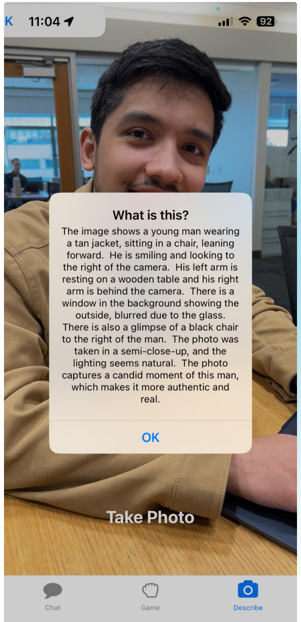
Yes! *throws toy*

Ooh, a feathery thing! *pounces and bats at toy*

Chat with a Cat



What Beats Rock?



What is this?

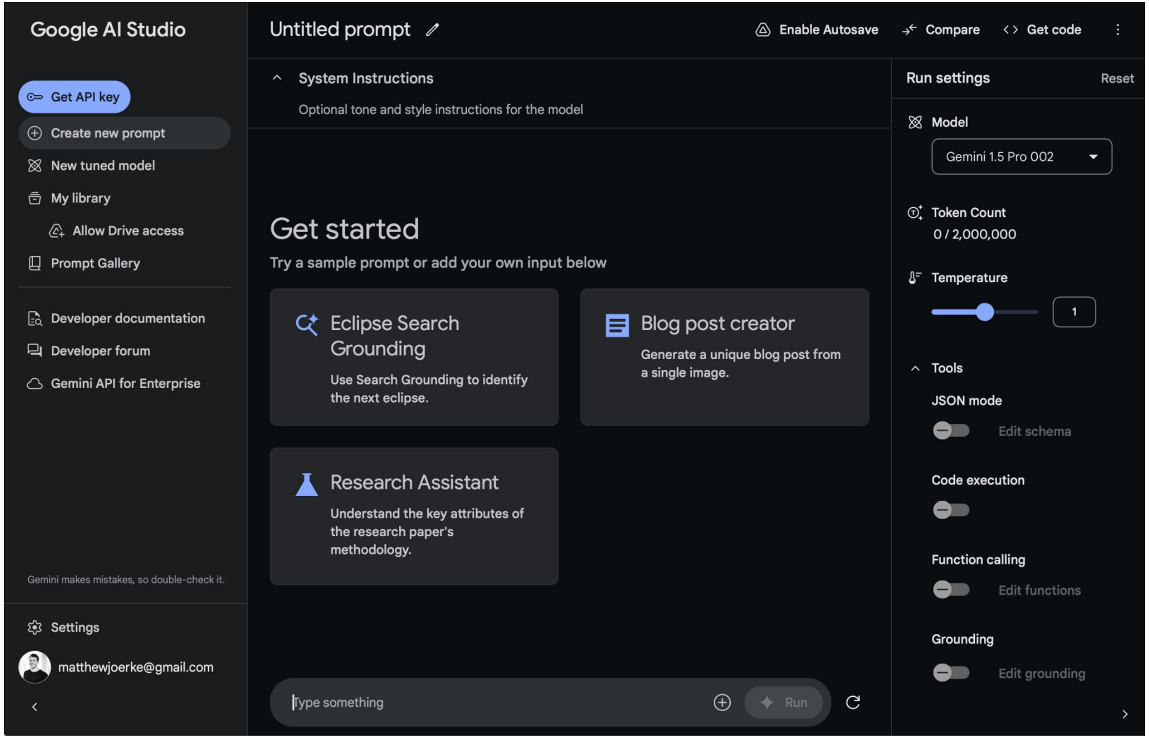
The image shows a young man wearing a tan jacket, sitting in a chair, leaning forward. He is smiling and looking to the right of the camera. His left arm is resting on a wooden table and his right arm is behind the camera. There is a window in the background showing the outside, blurred due to the glass. There is also a glimpse of a black chair to the right of the man. The photo was taken in a semi-close-up, and the lighting seems natural. The photo captures a candid moment of this man, which makes it more authentic and real.

Image Description

Disclaimers for using LLMs to write code

- You are still responsible for the design of the app and for the code generated
- Read the generated code and think critically about it.
  - What does it do? Why does it work (or not work)?
  - What can I learn from this code?
- AI is great for getting started on a project
  - But becomes less reliable as the complexity of the codebase scales
  - Polishing the last 20% is still the hardest part, with or without AI

Google AI Studio



System Instructions

Optional tone and style instructions for the model

Get started

Try a sample prompt or add your own input below

Eclipse Search Grounding

Use Search Grounding to identify the next eclipse.

Blog post creator

Generate a unique blog post from a single image.

Research Assistant

Understand the key attributes of the research paper's methodology.

Type something

Gemini makes mistakes, so double-check it.

Run settings

Model: Gemini 1.5 Pro 002

Token Count: 0 / 2,000,000

Temperature: 1

Tools

JSON mode

Code execution

Function calling

Grounding

Disclaimers for using LLMs to write code

- **You are still responsible** for the design of the app and for the code generated
- **Read the generated code** and think critically about it
  - *What does it do? Why does it work (or not work)?*
  - *What can I learn from this code?*
- AI is great for getting started on a project
  - But becomes less reliable as the complexity of the codebase scales
  - **Polishing the last 20%** is still the hardest part, with or without AI

LLM support in IDEs

- GitHub Copilot
- Codeium
- Cursor

Ø3

Coding Tutorials!

Setting up

- Create your Gemini API key: https://aistudio.google.com/apikey
- `git clone` the starter code: https://github.com/cs147L-24au/ai-workshop
- Navigate to the folder and run:
  - `npm install`
  - `npx expo start`
  - Run on your phone (Expo Go) or in a simulator

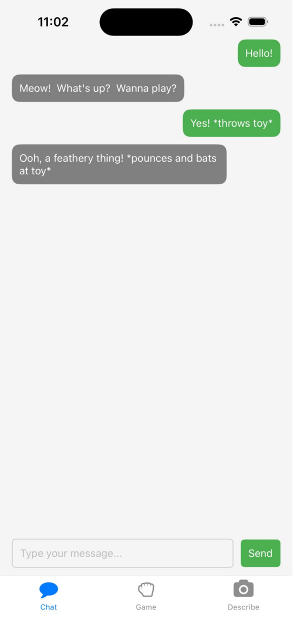
Using the Gemini API

```
import { GoogleGenerativeAI } from "@google/generative-ai";

const genAI = new GoogleGenerativeAI(YOUR_API_KEY);
const model = genAI.getGenerativeModel({
     model: "gemini-1.5-flash"
});

const prompt = "Write a story about a magic backpack.";

const result = await model.generateContent(prompt);
console.log(result.response.text());
```

More examples & starter code can be found in the

[Gemini API Docs]()

Today's Tutorials


11:02

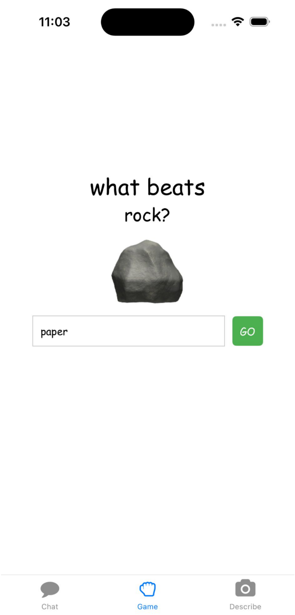
Hello!

Meow! What's up? Wanna play?

Yes! *throws toy*

Ooh, a feathery thing! *pounces and bats at toy*

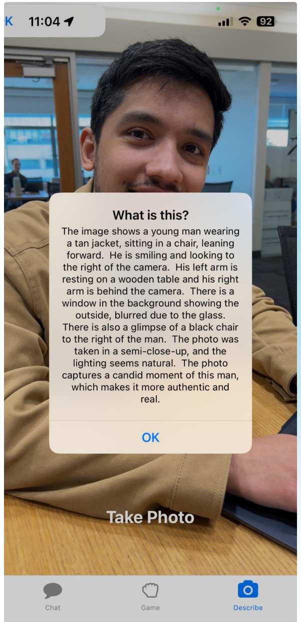
Chat with a Cat


11:03

what beats rock?

paper

What Beats Rock?


11:04

What is this?

The image shows a young man wearing a tan jacket, sitting in a chair, leaning forward. He is smiling and looking to the right of the camera. His left arm is resting on a wooden table and his right arm is behind the camera. There is a window in the background showing the outside, blurred due to the glass. There is also a glimpse of a black chair to the right of the man. The photo was taken in a semi-close-up, and the lighting seems natural. The photo captures a candid moment of this man, which makes it more authentic and real.

Image Description

Ø4

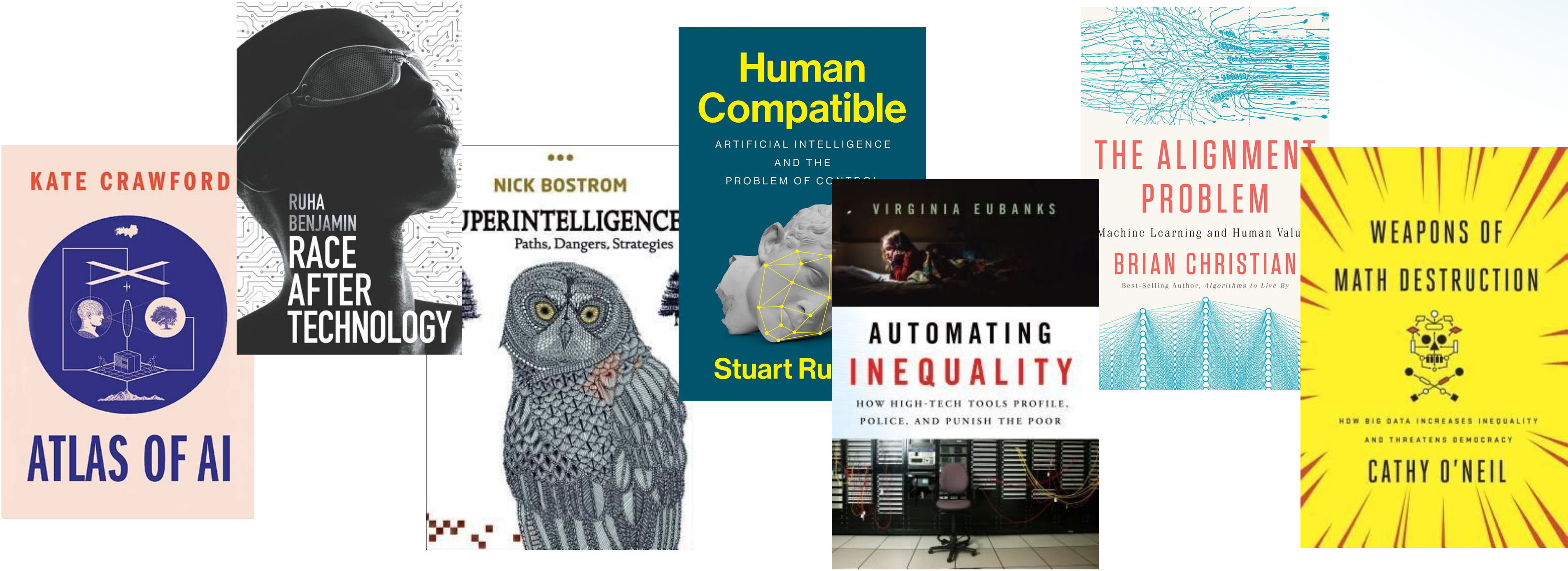
Safety & Ethics

⚠️ ‼️ Disclaimer ‼️ ⚠️

AI ethics is a **huge topic** that we cannot comprehensively cover in a 1h workshop.

⚠️ ‼️ Disclaimer ‼️ ⚠️

There are **many different perspectives** on what AI ethics means and what we should do about it.


KATE CRAWFORD
ATLAS OF AI
RUHA BENJAMIN
RACE AFTER TECHNOLOGY
NICK BOSTROM
UPERINTELLIGENCE
Paths, Dangers, Strategies
Human Compatible
ARTIFICIAL INTELLIGENCE AND THE PROBLEM OF CONTROL
Stuart Ru
VIRGINIA EUBANKS
AUTOMATING INEQUALITY
HOW HIGH-TECH TOOLS PROFILE, POLICE, AND PUNISH THE POOR
THE ALIGNMENT PROBLEM
Machine Learning and Human Valu
BRIAN CHRISTIAN
Best-Selling Author, Algorithms to Live By
WEAPONS OF MATH DESTRUCTION
HOW BIG DATA INCREASES INEQUALITY AND THREATENS DEMOCRACY
CATHY O'NEIL

Things I will cover

- Anticipating risk areas in the design process
- Bias, privacy, hallucination, agency, harmful outputs
- How to use the Gemini safety filters

Things I won't cover

- Implications on society, democracy, jobs, policy, etc.
- Domain-specific concerns (health, education, etc.)

Risk Area: Bias

LLMs can reflect or amplify societal biases present in their training data, which can lead to discriminatory responses.

> *Example:* In a resume screening app, an LLM could unfairly favor white and/or male candidates, discriminating against candidates from marginalized groups.
>
> *Example:* A job application bot could suggest professions are more suited to specific genders or ethnicities.
>
> *Example:* A travel recommendation bot could describe destinations or traditions in a way that reinforces stereotypes.
>
> *Example:* An LLM could anglicize or incorrectly interpret non-Western names (e.g., Shardubul vs. Shardul)

Risk Area: Privacy

LLMs using sensitive data may inadvertently disclose personal information.

> *Example:* A health support chatbot might send protected health information to the Google Gemini API.

Using LLMs may require users to share more sensitive information than was previously necessary.

> *Example*: A financial planning app using an LLM might prompt users to provide detailed personal financial data to generate advice.

Risk Area: Hallucination

LLMs can sometimes confidently produce responses that sound plausible but are factually incorrect or made up.

> *Example:* A legal assistance bot might provide legal advice based on laws that don’t actually exist.
>
> *Example:* A chatbot for medical diagnosis could give false medical information, recommending incorrect treatments or medications.

Risk Area: Agency

Relying too heavily on LLMs for decision-making can undermine human control and oversight.

> *Example:* If a doctor overly relies on an AI diagnostic tool, they might overlook critical issues that the AI fails to recognize.

Risk Area: Harmful Outputs

LLMs can generate offensive or inappropriate responses that can direct cause harm in sensitive situations.

> *Example:* A mental health support bot could respond insensitively or inappropriately to users in crisis.
>
> *Example:* A mental health support bot might validate a user expressing an intent to harm others.
>
> *Example:* In a virtual lab experiment app, the LLM suggesting mixing household chemicals in unsafe ways.

So... what can I do?

Disclaimer: There is no silver bullet; no single solution can mitigate all risks.

1. Use tools like the **Tarot Cards of Tech** to anticipate potential harms

So… what can I do?

Disclaimer: There is no silver bullet; no single solution can mitigate all risks.

1. Use tools like the **Tarot Cards of Tech** to anticipate potential harms.
2. Ask yourself: *are the benefits of using AI worth the risks?*

So… what can I do?

Disclaimer: There is no silver bullet; no single solution can mitigate all risks.

1. Use tools like the **Tarot Cards of Tech** to anticipate potential harms.
2. Ask yourself: *are the benefits of using AI worth the risks?*
3. Design for an imperfect AI (also applies for general UX)

So… what can I do?

Disclaimer: There is no silver bullet; no single solution can mitigate all risks.

1. Use tools like the **Tarot Cards of Tech** to anticipate potential harms.
2. Ask yourself: *are the benefits of using AI worth the risks?*
3. Design for an imperfect AI (also applies for general UX).
4. Think about technical mitigation strategies.

Gemini Safety Filters

```
import { HarmBlockThreshold, HarmCategory } from
"@google/generative-ai";

const safetySettings = [{
      category: HarmCategory.HARM_CATEGORY_HARASSMENT,
      threshold: HarmBlockThreshold.BLOCK_LOW_AND_ABOVE,
 },
 {
      category: HarmCategory.HARM_CATEGORY_HATE_SPEECH,
      threshold: HarmBlockThreshold.BLOCK_LOW_AND_ABOVE,
}];

const model = genAi.getGenerativeModel({
      model: "gemini-1.5-flash",
      safetySettings: safetySettings
});
```

Safety Settings Documentation

Five categories

- Harassment, Hate Speech, Sexually Explicit, Dangerous, Civic Integrity

Four risk levels

- Negligible, Low, Medium, Hisk

All code from today’s tutorial can be found here:

Questions?

Code Snippet Example

```
index.jsx

const handleSend = () => {
   if (inputValue.trim()) {
       sendMessage(inputValue);
       setInputValue('');
       inputRef.current?.clear();
       setInputHeight(2 * lineHeight);

       if (speechRef.current?.isRecording()) {
           speechRef.current.handleRecording();
       }
   }
};
```

Here is some text that might talk about the code