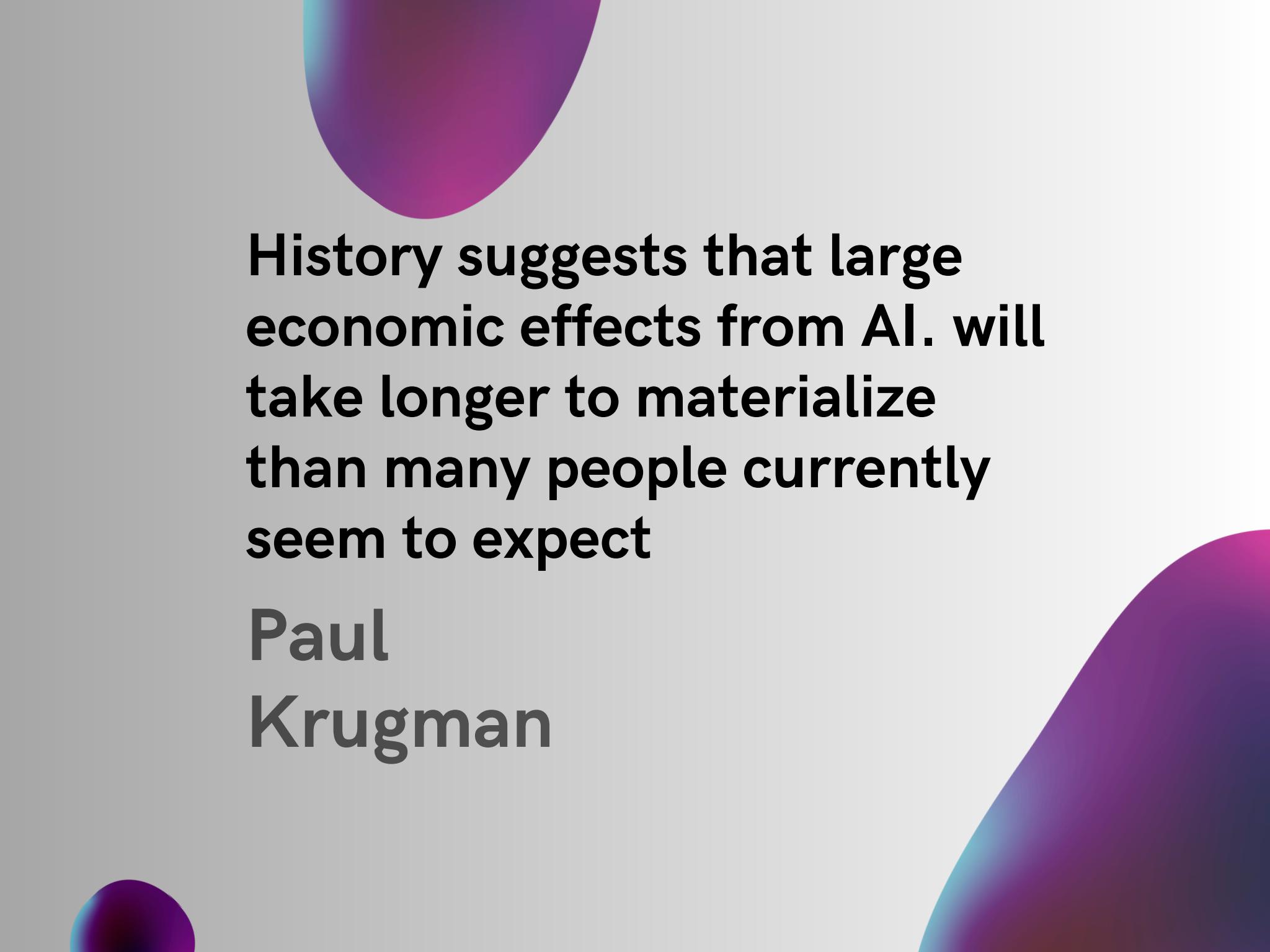


Prompt Engineering Best Practices



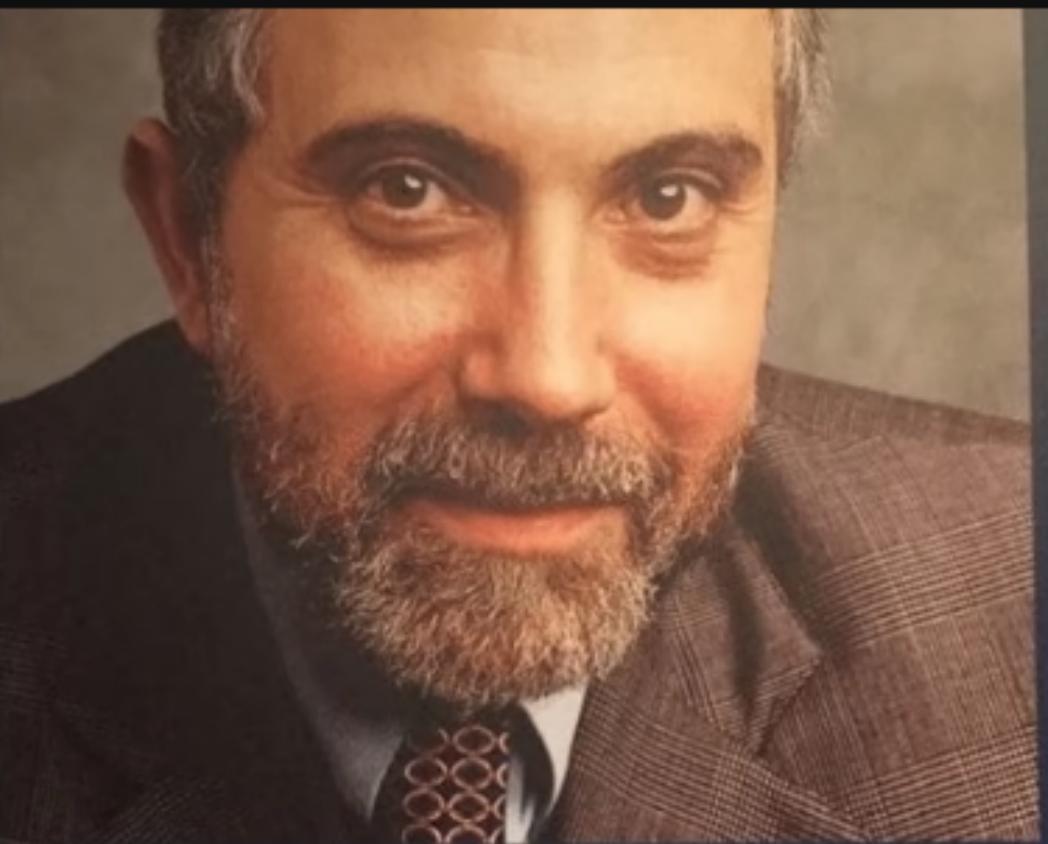
Objectives

- Large Language Models
- The basics of prompt engineering
- Prompt engineering best practices
- Share some resources



History suggests that large economic effects from AI. will take longer to materialize than many people currently seem to expect

**Paul
Krugman**

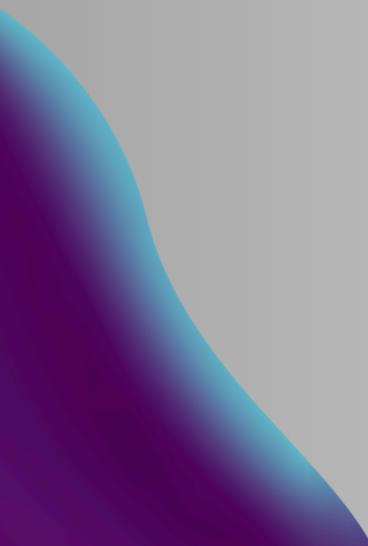


"By 2005 or so, it will become clear that the Internet's impact on the economy has been no greater than the fax machine's."

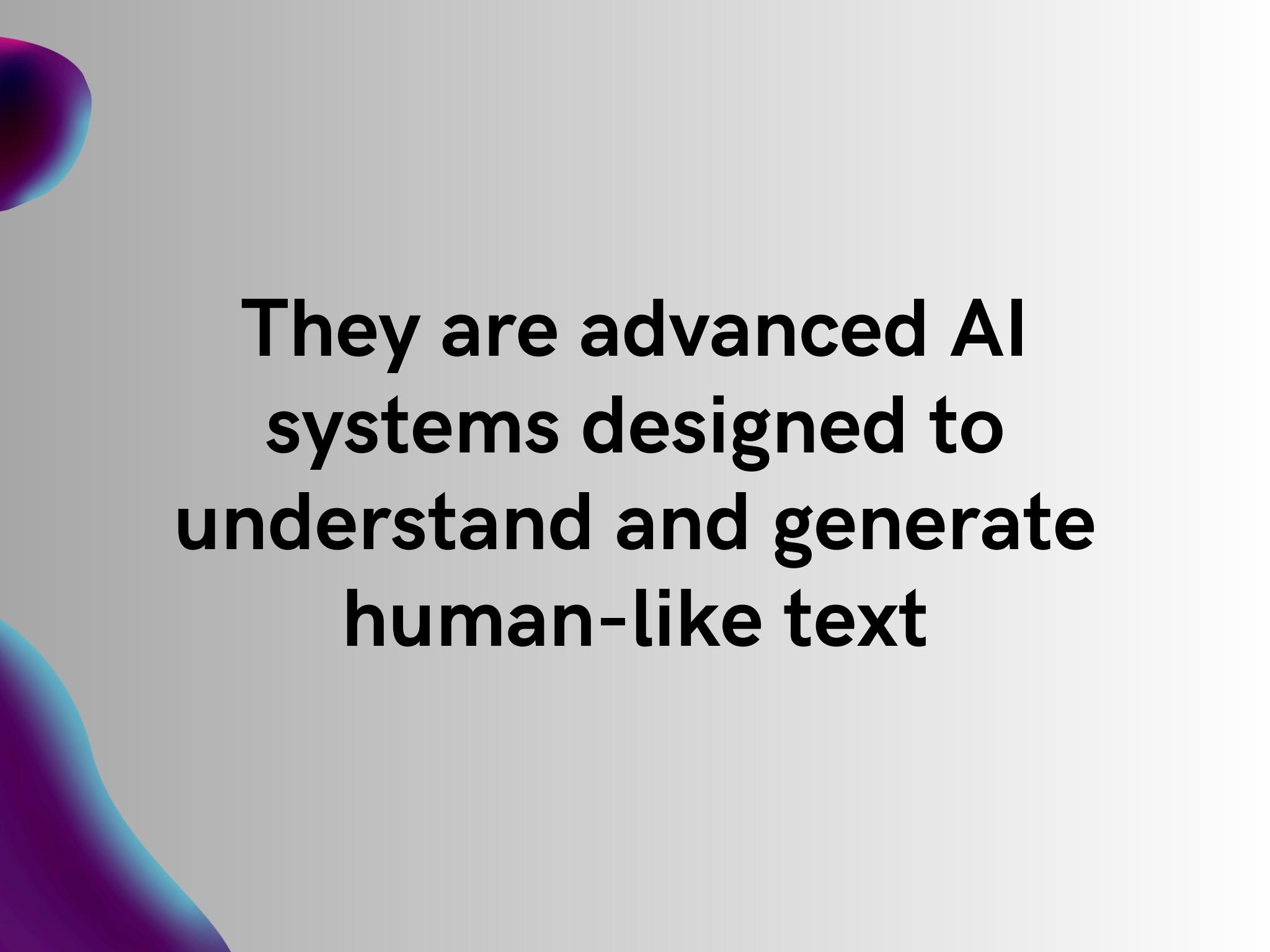
Nobel Prize-winning economist
Paul Krugman

1998

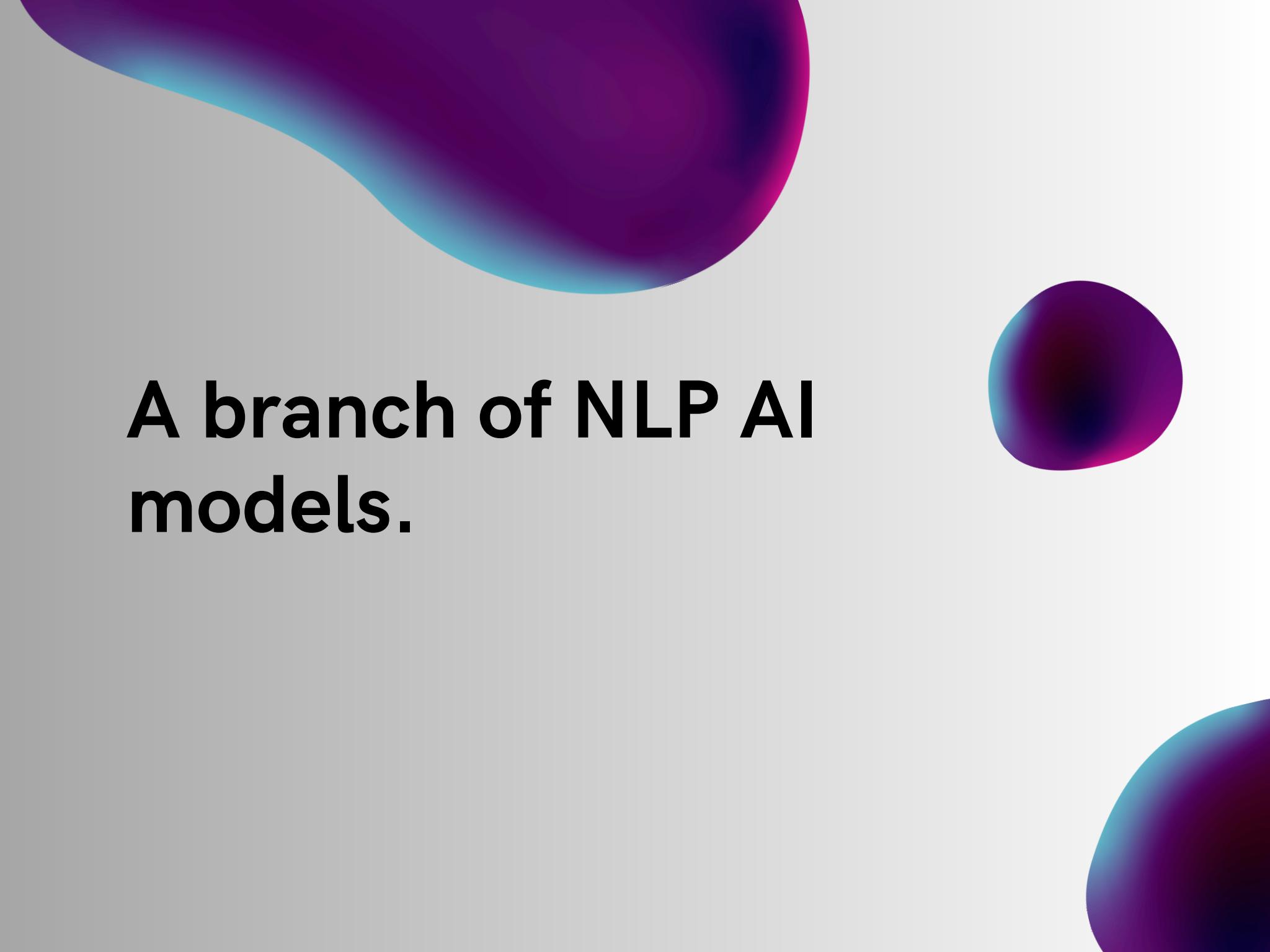




**So what ARE
LARGE LANGUAGE
MODELS (LLMs)?**

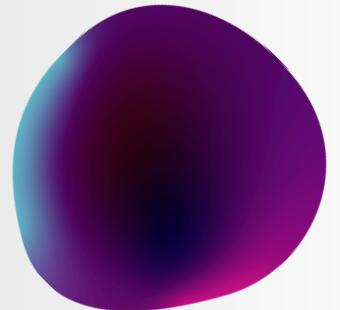


**They are advanced AI
systems designed to
understand and generate
human-like text**

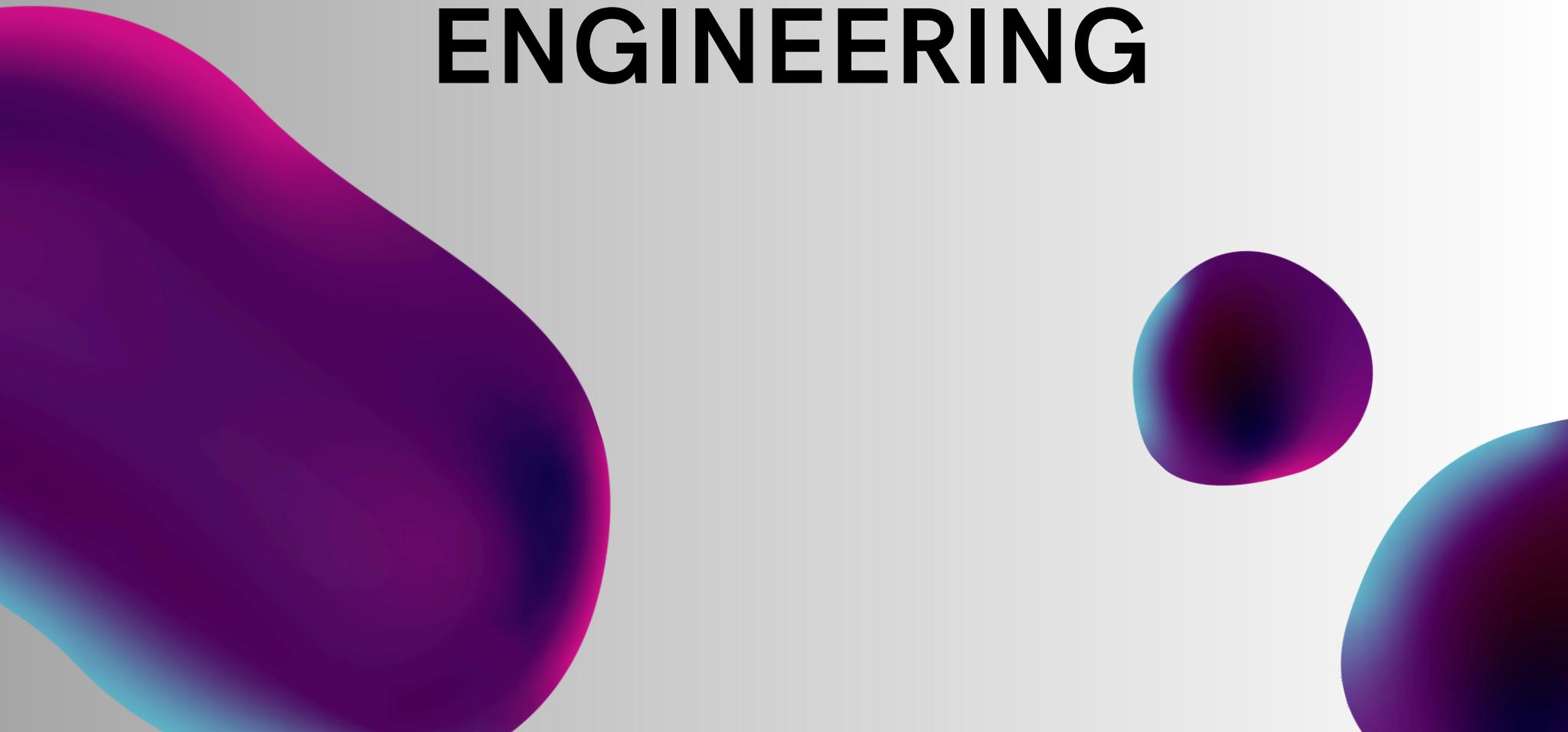


A branch of NLP AI
models.

We're used to doing
statistics with
numbers but LLMs
manage to do statistics
from text as well



PROMPT ENGINEERING



It useful to think of prompting as a linguistic user interface. We had graphical user interfaces and now we have linguistic ones)

Write clear instructions

"Write code to calculate the Fibonacci sequence"

**"Write a Rust function to efficiently calculate the Fibonacci sequence.
Add comments to the code to show what each piece does"**

requires vastly
dynamic

2 O benjami

Fibonacci in

Jan 8, 2018 – In
logic of the Fib
calling the same
integers...

3 umccor

Fibonacci in

Mar 13, 2021 – 1
different appro
Fibonacci sequ
speed of these

4 users.ru

Optimizing F

Computatio

Mar 14, 2021 – 1
implementation
numbers quickl
runs in about 12

5 coderev

Recursive Fi

memoizatio
Sep 28, 2018 –

```
use std::collections::HashMap;

// Function to calculate the nth Fibonacci number using memoization
pub fn fib_memoization(n: usize, memo: &mut HashMap<usize, usize>) -> usize {
    // Check if the value is already calculated and stored in the memo
    if let Some(v) = memo.get(&n) {
        return *v;
    }

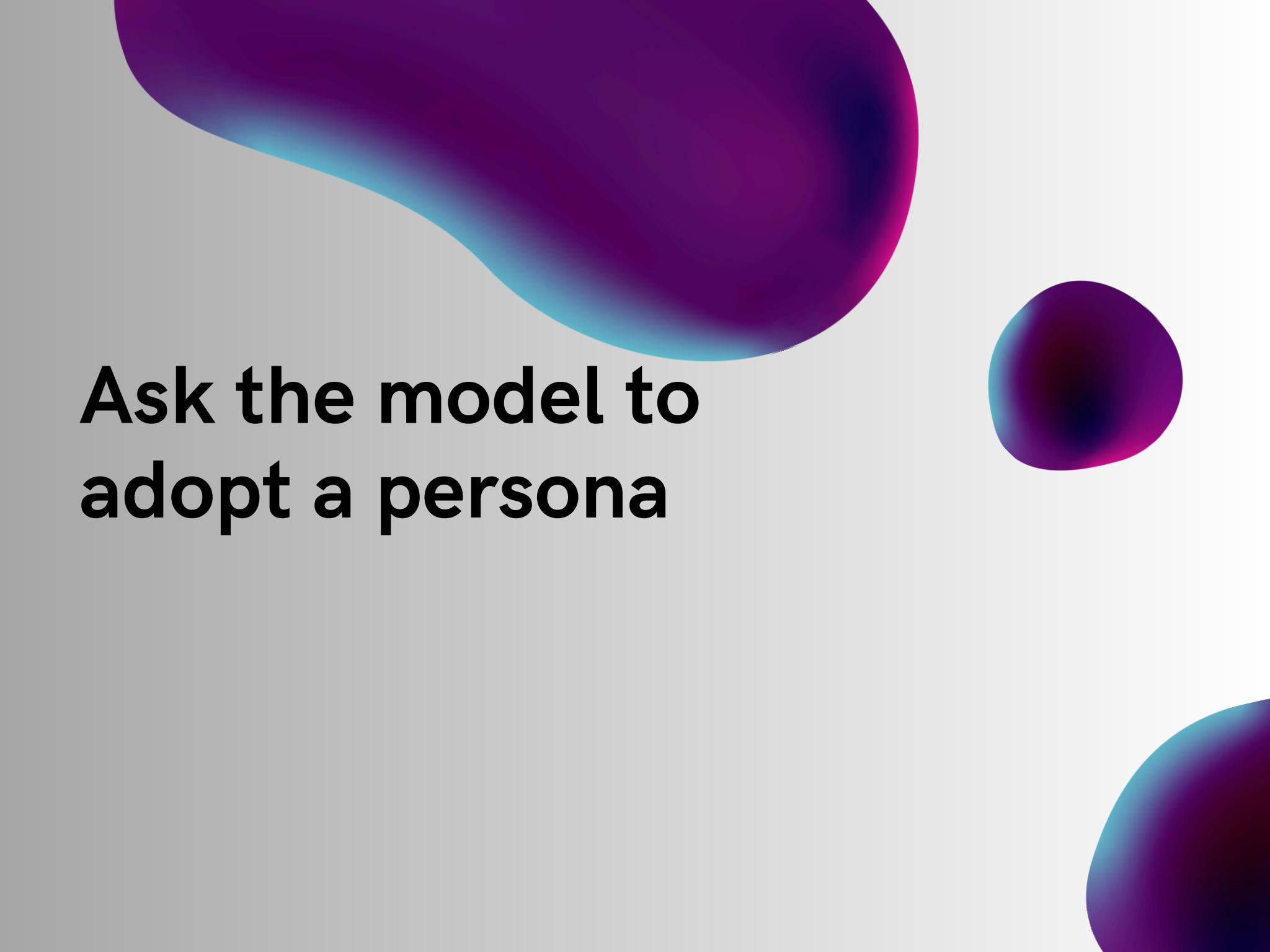
    // Base cases for the Fibonacci sequence
    let v = match n {
        0 | 1 => 1,
        _ => fib_memoization(n - 2, memo) + fib_memoization(n - 1, memo),
    };

    // Store the calculated value in the memo for future use
    memo.insert(n, v);
    v
}

fn main() {
    let n = 10; // Example input for the 10th Fibonacci number
    let mut memo = HashMap::new();
    let result = fib_memoization(n, &mut memo);
    println!("The {}th Fibonacci number is {}", n, result);
}
```

Ask a followup question

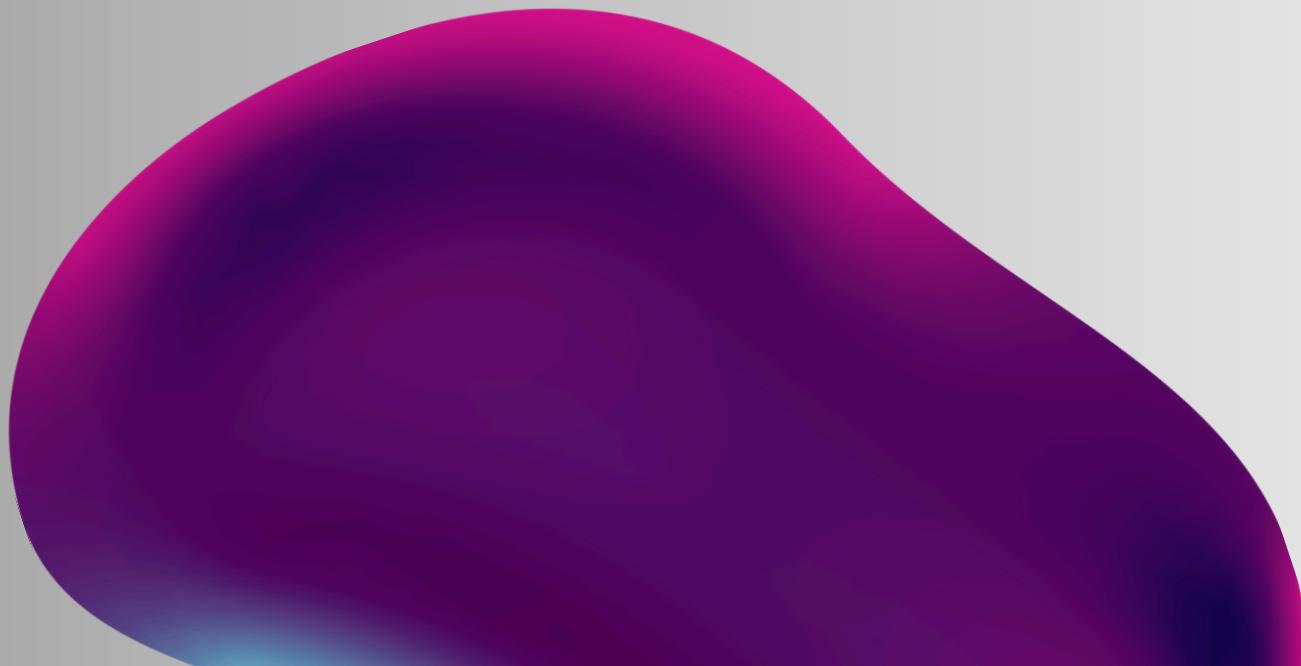
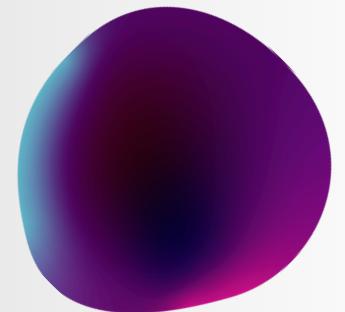




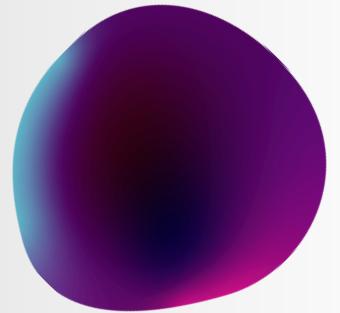
**Ask the model to
adopt a persona**

**When I ask you to
write something reply
with a document that
contains at least one
joke.**

**Use delimiters to
indicate distinct parts
of the input**



**Split Complex tasks
into simpler sub-tasks**



Beyond the basics

Hyperparameters



The hyperparameters are the parameters that define the behavior of the model.

The hyperparameter “temperature” influences how the model completion may change each time, even with the same input. Setting the temperature to 0 should give you the same output each time

Fine Tuning

Fine-tuning is the process of using a dataset of curated prompt-completion pairs to customize a model's behavior for your use case.

Fine-tuning can increase the accuracy of the completions for your prompts



ELIF

Explain it like I'm Five!

Prompt Engineering

+

Basic Coding
Knowledge

=

Build Anything

Replying to @karpathy and @ch402



**PRE-SOFTWARE:
SPECIAL-PURPOSE
COMPUTER**



**SOFTWARE 1.0:
DESIGN
THE ALGORITHM**



**SOFTWARE 2.0:
DESIGN
THE DATASET**



**SOFTWARE 3.0:
DESIGN
THE PROMPT**



Resources

- gwern -> <https://gwern.net/gpt-3#effective-prompt-programming>
- <https://www.freecodecamp.org/news/learn-prompt-engineering-full-course/>
- phind.com

