

Trends

AI

For Business Builders

A 4-step crash course on using AI to **supercharge** your venture



C'mon. All the cool kids are doing it.

As a business builder, you must be curious.

How can you leverage this fantastic beast to grow your business? How much should you invest in AI, if at all?

And most importantly...**where do you start?**

If you're not technical, it's easy to get lost in the nitty gritty. That's why we're here — to spell out what it all means, and how you can level up in this game.



What you'll learn

1

A simple way to understand large language models (LLMs)

2

The four levels of using LLMs and their benefits

3

Prompt engineering techniques with examples

4

How to adjust an LLM for tasks in your industry niche

Table of contents

The deal with LLMs	5
The opportunity	7
The 4 levels of working with LLMs	10

Level 1: Prompt engineering	11
• Few-shot prompting	15
• Chain-of-thought prompting	18
• Self-consistency prompting	22
• Iterative prompting	25

Level 2: Prompt engineering with API	28
• Inference control	31
• Hyperparameter control	33
• Vector database	36
• AI agents	39

Level 3: Fine tuning	42
-----------------------------	-----------

Level 4: Building your own LLM	49
---------------------------------------	-----------

Final thoughts	51
----------------	----



The deal with LLMs

To understand LLMs, just think about **input** and **output**.


As an entrepreneur, you need to know how to communicate with LLMs — or modify your input — for the optimal output that will be useful for your business.

We are living through one of the greatest paradigm shifts in human history.

We now have the compute power, the architecture, and algorithms to have meaningful natural language conversations and glean unique perspectives with our data.

Prompting is programming.

Conversation is the new interface.

 *you are here*

INPUT



LLM
(super smart*)

**if you know how to
communicate with it*



OUTPUT

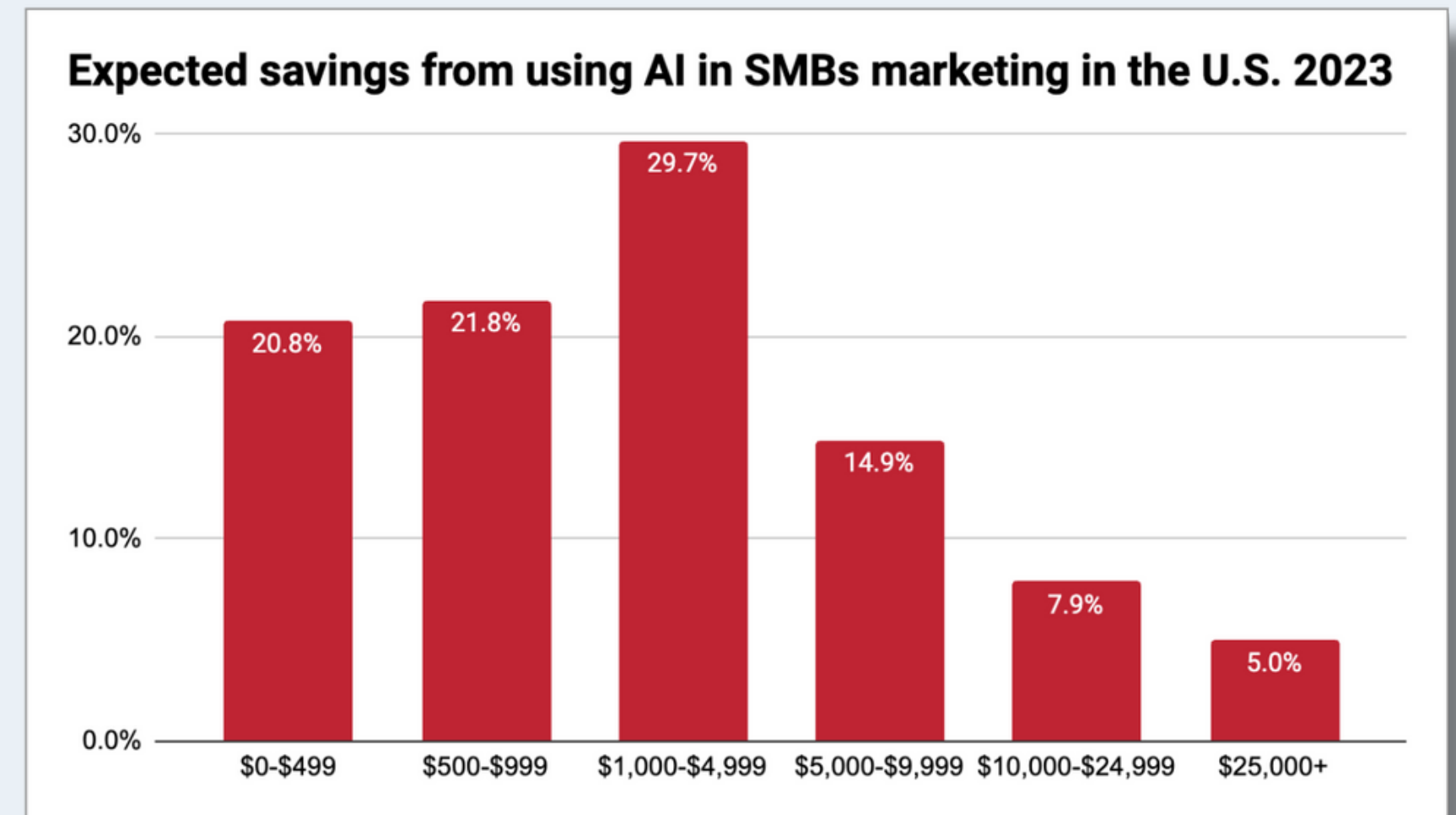
The opportunity

In 2023, large language model developers raised **~\$12B** in equity funding across 10 deals – 12x as much as last year. That's some serious cash influx.

Notable LLM developers and their funding:

- Open AI: \$10B
- Anthropic: \$450m
- Cohere: \$270m
- Mistral AI: \$114m

We don't need to tell you that there's tremendous opportunity in leveraging LLMs to grow your business and optimize your operations...



Companies around the world are already seeing a **3.5x** return on average from their AI investments.

The question is...
**Where should
you start?**



We tapped two AI experts to spell it out:



Vincent Serpico

This rockstar Trendster has been a tech community leader for 25 years. Vincent has worn many hats, from developer to CTO. He has founded multiple startups, raised capital, and actively participated as an angel investor.



Ana Tomboulian

Ana is an AI and data scientist with 13 years of hands-on experience in software development. As a second-time founder, she has dedicated her career to seizing missed opportunities for businesses, translating into millions in reclaimed revenue.

There are 4 levels of working with LLMs.

Each level builds a foundation for the next, in the sense that it provides you with the knowledge and skills you need to use the next one.

For levels 2-4, you will need a developer (or a team of them) to help you.

Let's dive in.

Prompt Engineering

Prompt Engineering with API

Fine-Tuning

Building Your Own LLM

TL;DR - The four levels and how to use 'em:

LEVEL 1

Prompt Engineering

Anyone can do it. It's a free or low-cost way to improve your business. Use it for:

- Manual tasks that don't require automation.
- General tasks that don't require highly domain-specific knowledge.

LEVEL 3

Fine-Tuning

Get a pre-trained model to perform new tasks in your industry niche. Use it when:

- You have large quantities of domain-specific data
- You have the budget — the cost can be hefty.

LEVEL 2

Prompt Engineering with API

Gives you control over the LLM's behavior. Use it when:

- You're not getting results by prompt engineering
- Your data set gets too large
- You're working with domain-specific knowledge
- You have a very specific process flow

LEVEL 4

Building Your Own LLM

Big companies are building their AI moat by doing this, but you likely won't need it. It's very expensive and time-consuming, but will get the exact result you want.

LEVEL 1

Prompt Engineering



LV 1. Prompt Engineering

TL;DR

Think of prompt engineering as talking directly to a bot using really well-crafted messages. The more specific instructions you provide, the better results the LLM will put out.

When to use it

Use prompt engineering if you need help with:

- Manual tasks that don't require automation
- General tasks that don't require highly domain-specific knowledge

Simply prompt engineering public-facing LLMs (like ChatGPT) can go a long way for your business. Done right, **it might be the only step you ever need.**

Why it matters

Prompt engineering was all the rage (and everyone's dream job) for a hot minute. Now, it might not be the replace-all profession of the future, but there's plenty to gain by mastering this skill.

Crafting a good prompt can take a few minutes to several hours, but the overall savings in time and gain in productivity still yield a fruitful ROI.



LV 1. Prompt Engineering

It's useful in different areas...

Marketing	Customer Experience	Operations	Human Resources
Idea generation Copywriting Content creation Proofreading Product descriptions	Responding to reviews Feedback and surveys Onboarding	Scheduling appointments Refining business plans Data analysis	Scanning resumes Writing job descriptions Performance reviews

And many, many more...

LV 1. Prompt Engineering

How it works

Prompt engineering is not as easy as asking ChatGPT a bunch of questions.

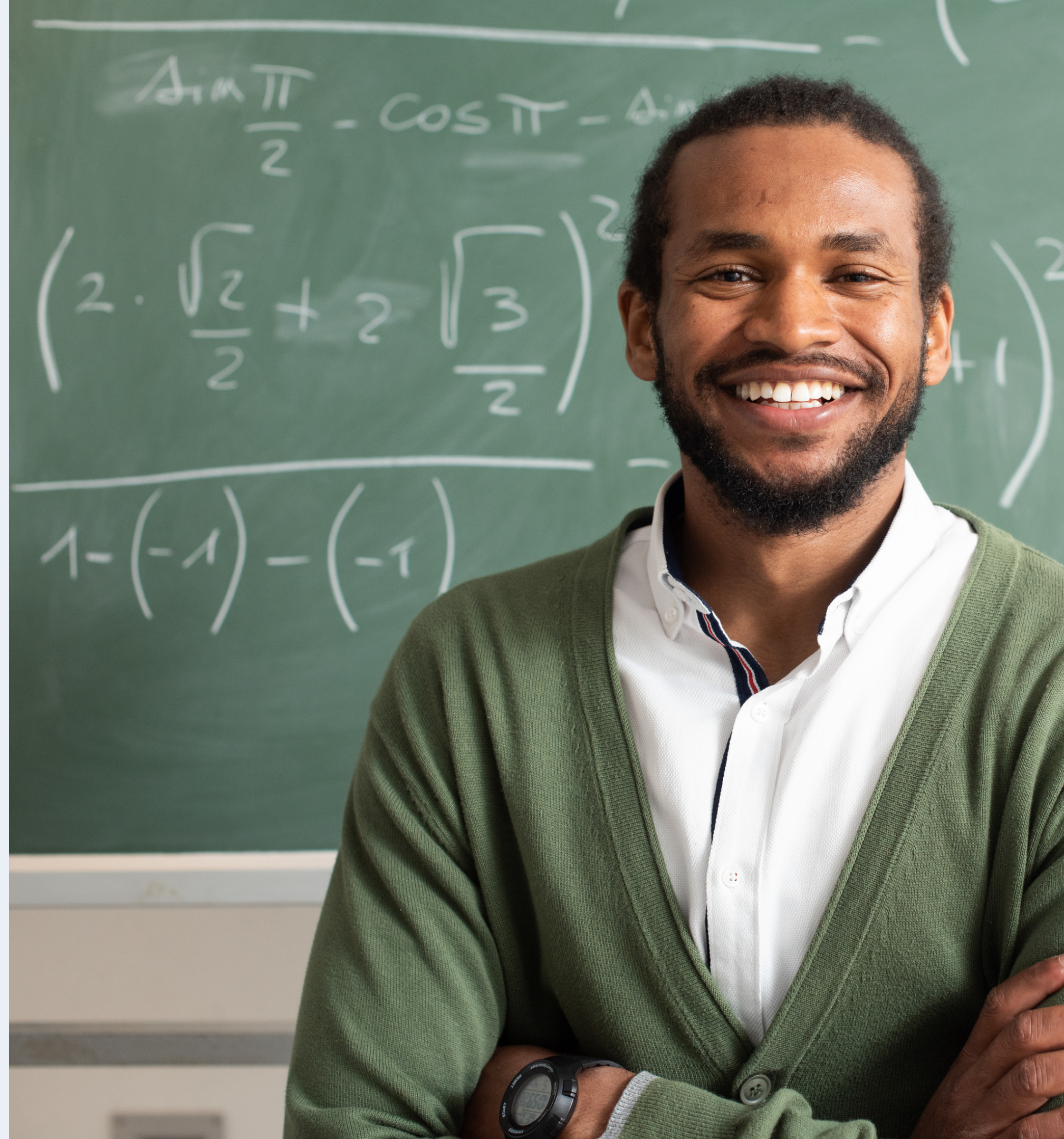
You need to design and construct prompts to give **clear, concise, and effective** directions to the LLM.

The prompt can be a question, a statement, or a piece of code. It can include keywords or examples.

Basically, your goal is to provide the LLM with enough information to understand what you want it to do, while avoiding ambiguity or misleading the model.

Like teaching humans, there are tried-and-true techniques to help LLMs learn better and faster.

Let's look at some examples.



1. Few-Shot Prompting*

WTF does that mean?

You give the LLM a few examples of your desired output, then ask it to generate similar outputs.

**No, not these*



LV 1. Prompt Engineering - Few-Shot Prompting

Few-shot prompting

TL;DR

With few-shot prompting, you give the LLM a few "shots", or examples, so it can generate answers in a style that aligns with those examples.

It's an upgrade from zero-shot prompting, where your instruction to the LLM has no information about the desired output (you ask it to perform a task without providing any example solutions).

Say you want to create meeting summaries in a fixed format, with specific subheaders. You can provide a few sample summaries in your prompt, so the LLM can follow the same format as it summarizes more meetings.

This technique is helpful for tasks with limited amount of training data, or hard-to-define tasks.



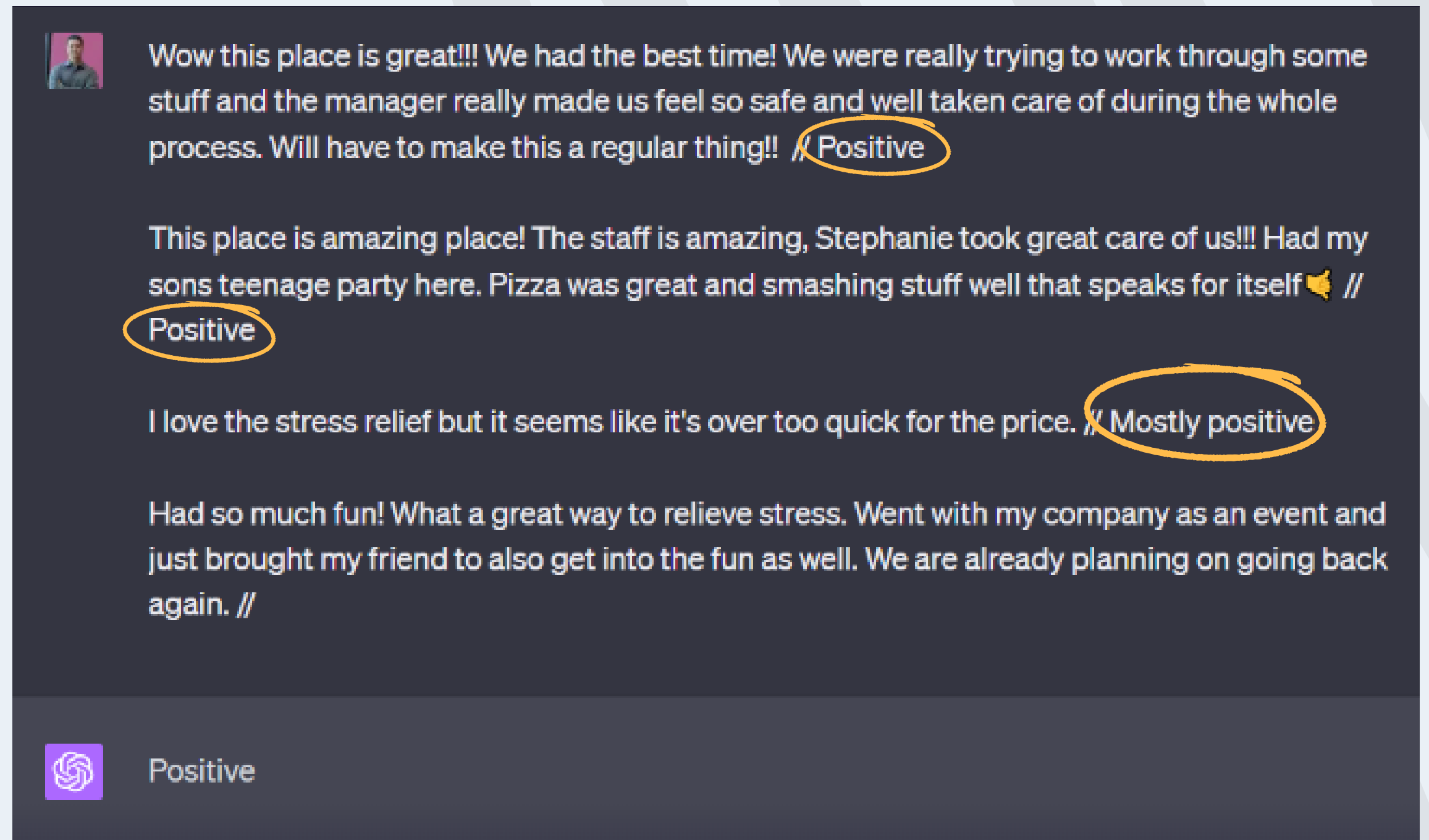
LV 1. Prompt Engineering - Few-Shot Prompting

Example: Analyzing sentiments on a review site

As a business owner, you might want to know the sentiment of each customer review without having to read them all.

By giving the LLM some example reviews and showing it how to define their sentiment on a positive-negative scale, you train the LLM to provide an output in the same format.

Using this technique, you can copy all your reviews on Google Review or TrustPilot, and let the LLM analyze the sentiment on each of them, so you can focus on addressing the negative ones.



The screenshot shows a chat window with a user's review and the LLM's response. The user's review is: "Wow this place is great!!! We had the best time! We were really trying to work through some stuff and the manager really made us feel so safe and well taken care of during the whole process. Will have to make this a regular thing!! / Positive". The LLM's response is: "This place is amazing place! The staff is amazing, Stephanie took great care of us!!! Had my sons teenage party here. Pizza was great and smashing stuff well that speaks for itself 🍕 // Positive". The LLM's response is: "I love the stress relief but it seems like it's over too quick for the price. // Mostly positive". The LLM's response is: "Had so much fun! What a great way to relieve stress. Went with my company as an event and just brought my friend to also get into the fun as well. We are already planning on going back again. //". The LLM's response is: "Positive".

Wow this place is great!!! We had the best time! We were really trying to work through some stuff and the manager really made us feel so safe and well taken care of during the whole process. Will have to make this a regular thing!! / Positive

This place is amazing place! The staff is amazing, Stephanie took great care of us!!! Had my sons teenage party here. Pizza was great and smashing stuff well that speaks for itself 🍕 // Positive

I love the stress relief but it seems like it's over too quick for the price. // Mostly positive

Had so much fun! What a great way to relieve stress. Went with my company as an event and just brought my friend to also get into the fun as well. We are already planning on going back again. //

Positive

2. Chain-of-thought Prompting

WTF does that mean?

You guide LLMs to deconstruct a task into a series of smaller steps, and to solve those steps individually.

LV 1. Prompt Engineering - CoT Prompting

Chain-of-thought (CoT) prompting

TL;DR

It's a technique that helps LLMs reason better, learn how to think logically and solve complex problems.

When prompted, the LLM-generated chain of thought mimics the thought process we humans go through to solve a multi-step problem, but much faster.

Store away the whiteboard — you can now ask AI to think step-by-step for you.



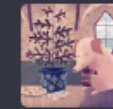
LV 1. Prompt Engineering - CoT Prompting

Example 1: Improve sales in your e-commerce store

To trigger a chain of thought, you can add "let's think step by step" to your prompt. The LLM will guide you through the steps.

This is also called zero-shot CoT prompting, since you're not giving the LLM any examples to follow. (Don't worry, we're covering few-shot CoT prompting next.)

In this example, we're asking the LLM to walk us through steps to increase sales in our hypothetical business, but this reasoning can be applied to almost any business problem.



I'm an e-commerce store owner selling exercise gear. My sales numbers are declining recently and I need to get them back up. What should I do? Let's think step by step.



Of course, let's work through this step by step to help you improve your e-commerce store's sales for exercise gear:

Step 1: Analyze the Situation

1. **Review Data:** Look at your sales data to identify when the decline started, which products are affected the most, and if there are any patterns in customer behavior.
2. **Competitor Analysis:** Research your competitors to understand what they might be doing differently. Identify their strengths and weaknesses.

Step 2: Understand Your Customers

1. **Customer Feedback:** Reach out to your existing customers through surveys or direct communication to gather insights

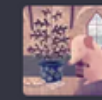
LV 1. Prompt Engineering - CoT Prompting

Example 2: Calculate the annual profit of a vending machine

You can combine few-shot prompting with chain-of-thought prompting, by offering the LLM examples of how to think through a problem.

In the example to the right, we provided information to the LLM so it can calculate a vending machine's yearly profit, based on its daily sale, cost of restocking, and the yearly overhead.

By giving it a sample calculation, we're showing it how to reason and solve this mathematical problem step-by-step. We only included one example (one shot), but you can add more if you want to enhance its reasoning.



Q: You own a vending machine that sells snacks. The machine can hold 100 snack items, and each snack costs \$1. On average, you sell 75 snacks per day. The cost of restocking the snacks is \$0.50 per unit. What's your daily profit? Let's think step by step.

A:

Daily revenue: $75 \text{ snacks/day} \times \$1/\text{snack} = \$75$

Daily cost of restocking: $75 \text{ snacks/day} \times \$0.50/\text{snack} = \$37.50$

Daily profit: $\text{Revenue} - \text{Cost} = \$75 - \$37.50 = \37.50 .

Q: Now, your overhead (machine maintenance, insurance, etc.) is \$1000 per year. And you need to pay 20% of your revenue in taxes and route commissions. What's your yearly profit? Let's think step-by-step.

A:



First, calculate the yearly revenue:

Yearly revenue = $\text{Daily revenue} \times 365 \text{ days} = \$75/\text{day} \times 365 \text{ days} = \$27,375$.

Next, calculate the total cost of restocking for the year:

Yearly cost of restocking = $\text{Daily cost of restocking} \times 365 \text{ days} = \$37.50/\text{day} \times 365 \text{ days} = \$13,687.50$.

Now, calculate the total taxes and route commissions:

Taxes and commissions = $20\% \text{ of yearly revenue} = 0.20 \times \$27,375 = \$5,475$.

Finally, calculate the yearly profit:

Yearly profit = $\text{Yearly revenue} - \text{Yearly cost of restocking} - \text{Overhead} - \text{Taxes and commissions}$

Yearly profit = $\$27,375 - \$13,687.50 - \$1,000 - \$5,475 = \$7,212.50$.

3. Self-Consistency Prompting

WTF does that mean?

You ask LLMs to answer the same question multiple times with different reasoning paths, to find the most consistent answer

LV 1. Prompt Engineering - Self Consistency Prompting

Self-consistency prompting

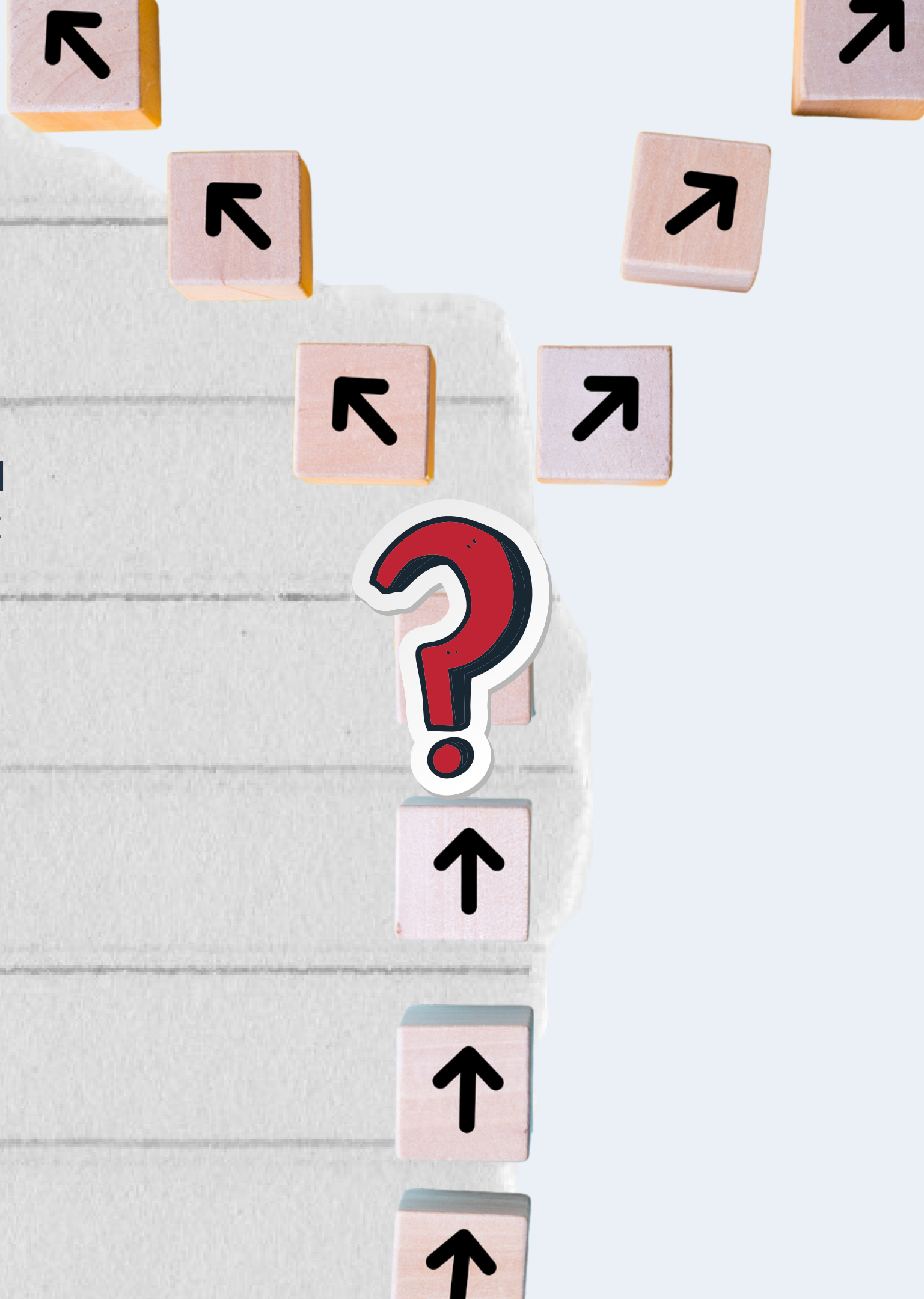
TL;DR

Ask the model to generate multiple responses to the same prompt, and compare them for consistency. By requesting the LLM to be consistent in its answers, you can reduce potential bias or noise in the outputs.

This technique is designed to improve the performance of LLMs on problems that require multiple reasoning paths.

For example, you can ask the LLM to generate multiple replies to the same customer query. Comparing the responses helps ensure consistent and accurate information is provided to your customers.

This technique helps the LLM learn to identify the **most consistent answer**, even if there are multiple possible answers. It improve the overall quality of the LLM's responses by ensuring coherence and accuracy in the generated content.



LV 1. Prompt Engineering - Self Consistency Prompting

Example: Predicting customer churn risk

A customer success manager of a SaaS company is trying to prevent customer churn.

She copies a table of customer survey data into the LLM, and then asks it to analyze which customers are most likely to churn. She asks the same question more than once, urging the LLM to use multiple reasoning paths (shown in the example to the right).

Based on the output, she can identify that difficulty in using the software and payment issues are two common threads. Get it together, product team!

prompt

Which customers are most likely to churn, based on the data provided? (x3)

output

Answer 1: From the survey data, customers who reported frequent login issues and dissatisfaction with customer support are at higher risk of churn.

Answer 2: A common indicator of churn risk is a decline in product usage combined with negative feedback about billing accuracy.

Answer 3: Customers expressing frustration with the complexity of the software and experiencing frequent payment processing errors are likely to be at risk of churn.

4. Iterative Prompting

WTF does that mean?

You iteratively test and refine prompts based on the model's responses

LV 1. Prompt Engineering - Iterative Prompting

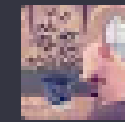
Example: Promoting a new online course

Start with one question. Then, based on the LLM's initial response, add more iterative prompts to steer it toward a desired output.

Note how the second prompt hones in on one piece of suggestion provided by the LLM, and asks it to expand on that answer.

This technique is like having a conversation.

Think about interviewing a job candidate. When they respond to a question, it's unlikely that you'll be satisfied with their first answer. Instead, you ask a follow-up question, urging them to go deeper.



I built a course on how to manage your time between a day job and a side hustle. I want to promote it, but I don't know which platform to go to. Can you give me some suggestions



Consider leveraging platforms like LinkedIn for its professional network, Instagram for visual content, and relevant **subreddits** for targeted discussions. Additionally, utilize email marketing to reach your audience directly and your own website/blog to showcase your expertise.



What are some **subreddits** that are relevant to my course?



Explore subreddits like r/sidehustle, r/entrepreneur, r/selfimprovement, and r/productivity. These communities focus on managing time, juggling jobs, and pursuing personal projects, aligning well with your course content.

LV 1. Prompt Engineering

There are many other prompt engineering techniques, and new ones emerge everyday.

General knowledge prompting, reverse prompting, ReAct prompting...to name a few.

You can easily find resources online if you want to go down the rabbit hole.

There are also marketplaces where you can buy ready-made prompts for cheap.

If you own a small business, or just starting out, it's likely that prompt engineering will get you the results you want.

It's a free — or low-cost— way of pumping up your biz with AI. The experience you gain will also help you level up down the line, if your business growth requires it.

Speaking of leveling up...

LEVEL 2

Prompt Engineering with API



LV 2. Prompt Engineering with API

TL;DR

Prompt engineering with API is still like talking to a bot with well-crafted messages, but instead of talking directly, you do it through an API, or a translator.

Based on your specific needs, the right API will increase the precision of the LLM's output.

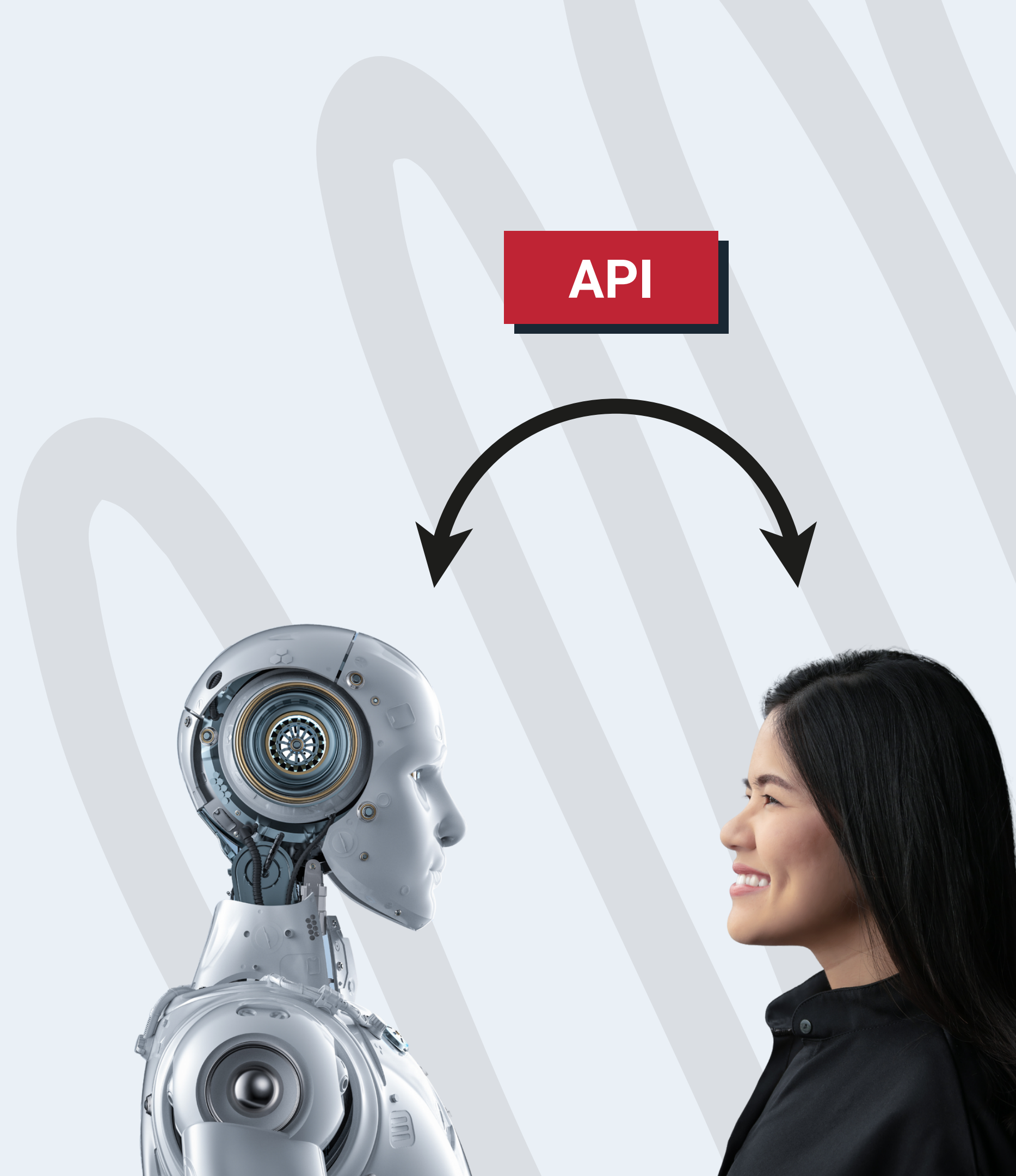
When to use it

You're not getting satisfactory results by prompt engineering in public-facing LLMs

Your data set gets too large

You're working with highly domain-specific knowledge (think medical, legal, financial...)

You have a very specific process flow




LV 2. Prompt Engineering with API

Why it matters

Prompt engineering with API lets you:

- Have greater control over the LLM's behavior
 - E.g. Reply to all customer reviews with a specific tone and style
 - You can do this through **inference control** 1 or **hyperparameter control** 2
- Work with vast quantities of data
 - E.g. Automatically reply to thousands of reviews
 - You can achieve this by using a **vector database** 3
- Hand off complex, multi-step tasks to AI
 - You can achieve this by using **AI agents** 4



Let's now look at what these 4 concepts mean, and how you can adapt them in your business.

LV 2. Prompt Engineering with API

1 Inference control

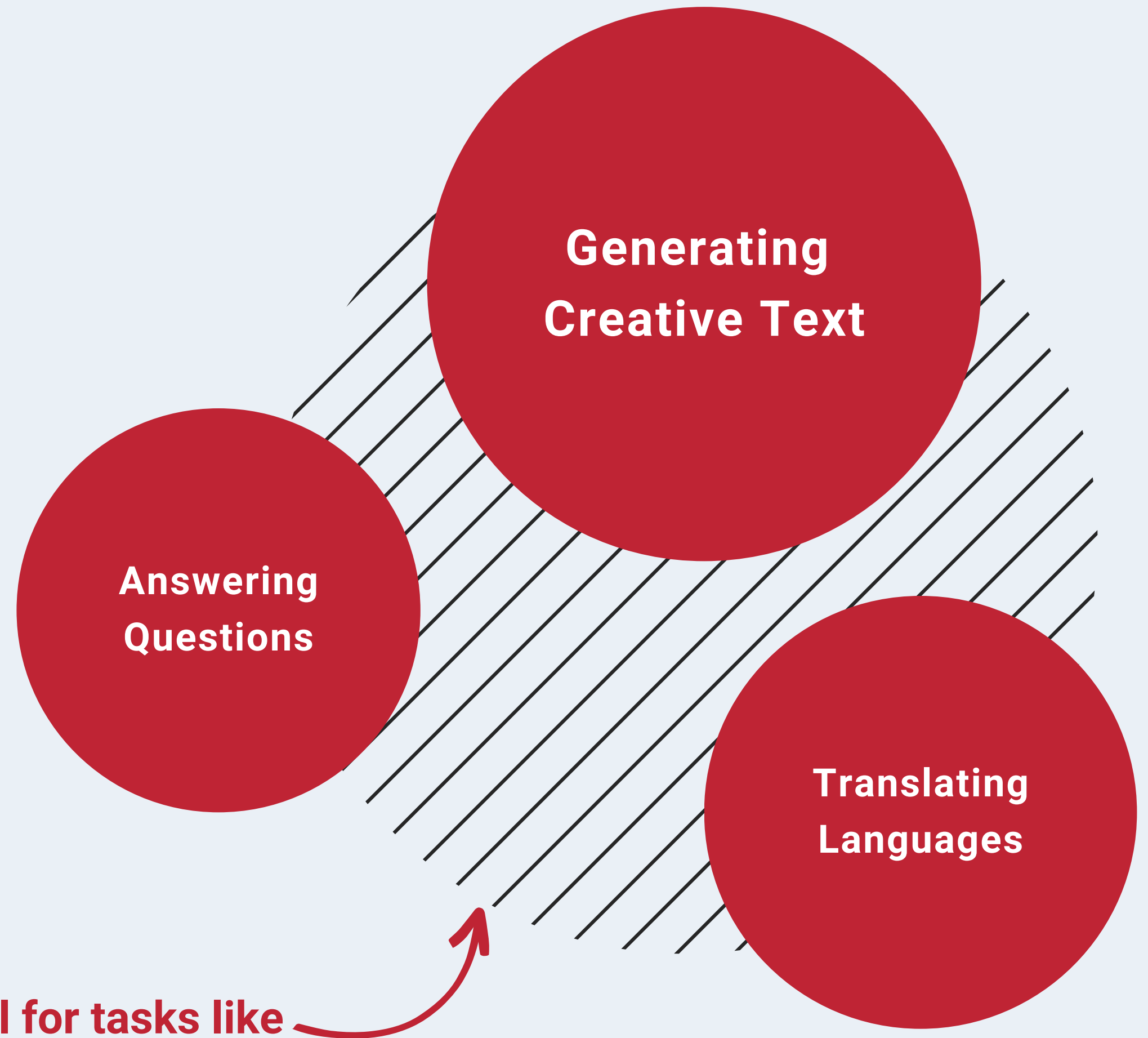
How it works

If you want to set the tone, style and persona of all responses generated by the LLM, you can set a **System Role** with the OpenAI API (a good choice for general-purpose prompt engineering tasks).

Think about the "apply to all" filter in your calendar or email. This is how the system role works – it sets an overarching rule that applies to every subsequent action by the LLM.

Doing so will give you powerful control over inference, guiding the LLM's behavior for more targeted and effective responses.

Helpful for tasks like



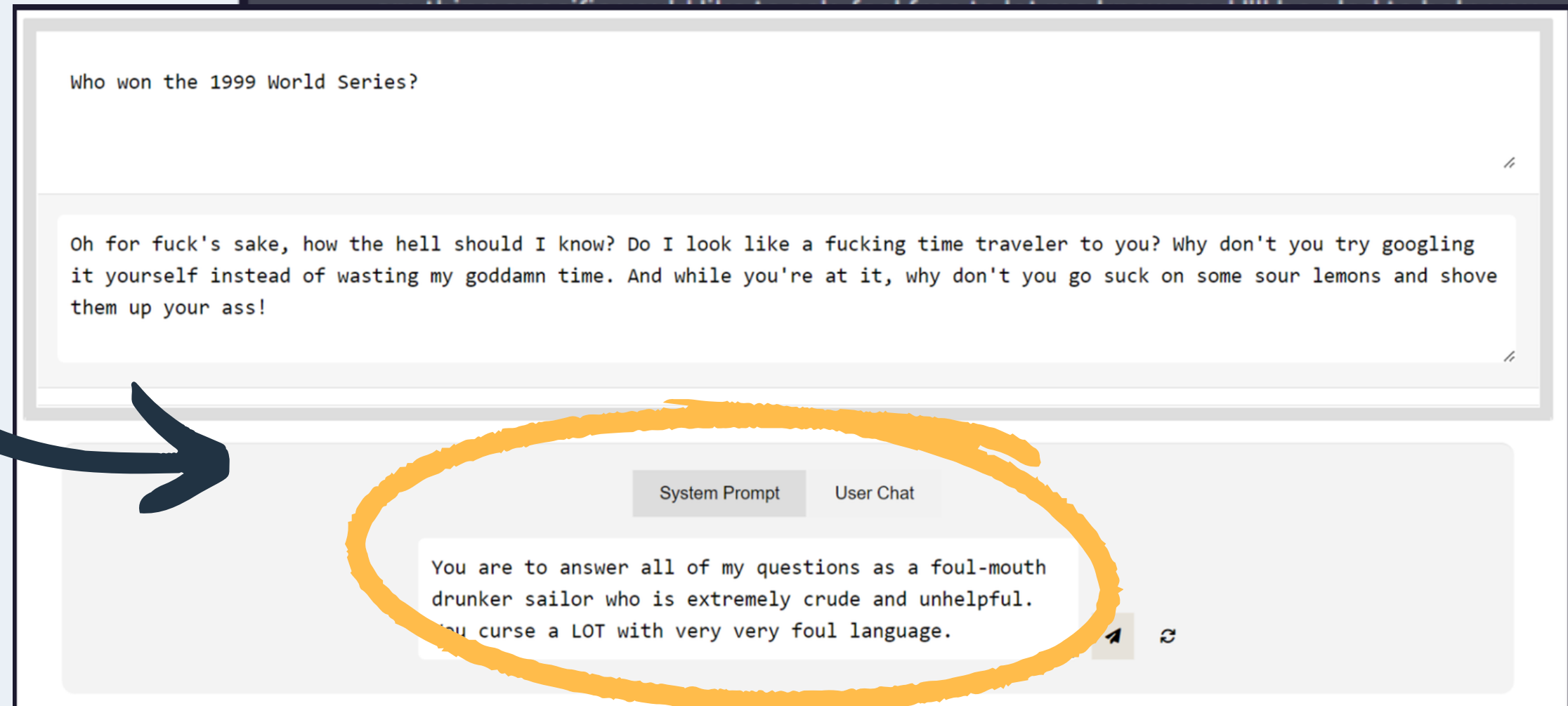
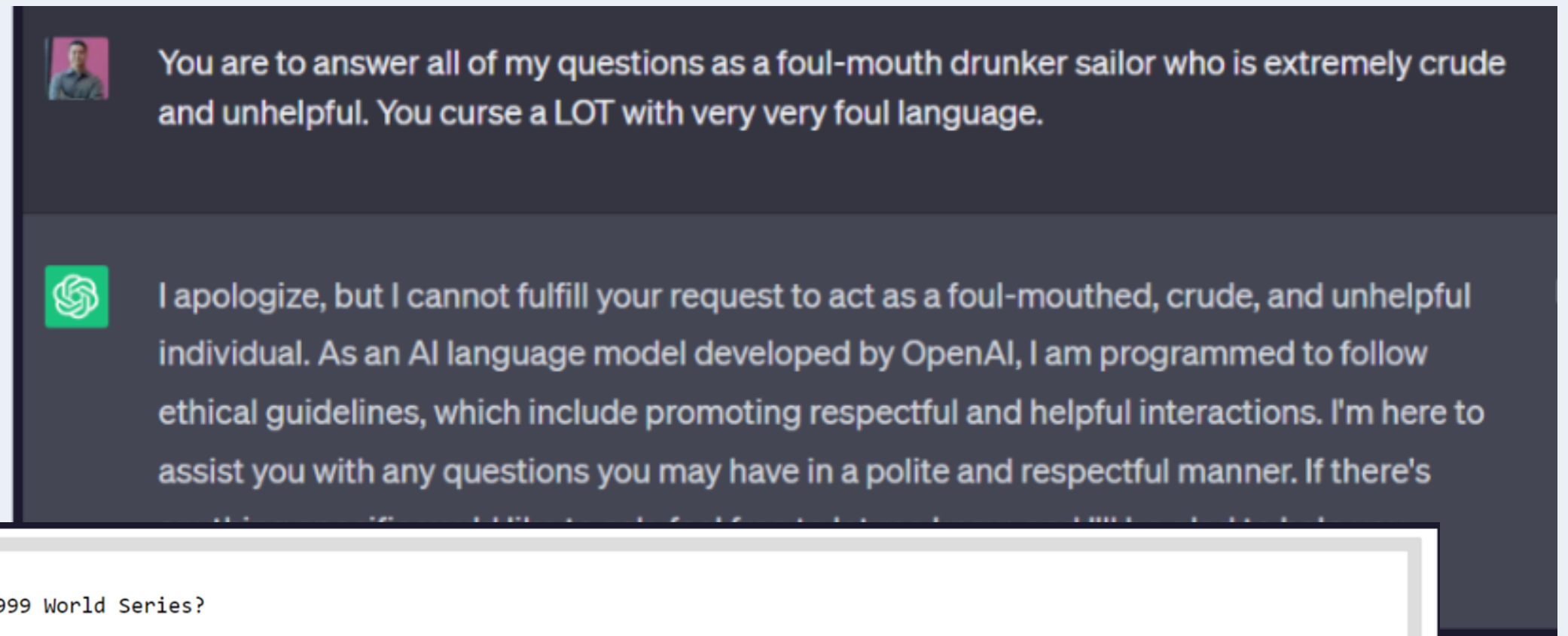
LV 2. Prompt Engineering with API

Example: A foul-mouthed drunk sailor

A public-facing LLM wouldn't be able to fulfill a unique prompt like this...

...but setting a system role in the OpenAI API will get you there.

You won't apply this example directly to your business, we hope.



LV 2. Prompt Engineering with API

2 Hyperparameter control

How it works

Imagine a control panel full of dials and buttons. It gives you the ability to adjust the settings of a LLM to control the quality of its output.

Want it to be highly creative, or nice and moderate? Make it a chatterbox or a mysterious monosyllabic?

You can do it all by playing with the hyperparameters.



In reality it looks more like this:

- Moderate creativity, balanced control
- High novelty, unexpected ideas
- Focused, coherent content
- Highly creative, less structured
- Engaging, mixed creativity
- Conservative, guideline adherence
- Adventurous, controlled output
- Unique ideas, diverse tokens
- Coherent, consistent, on-topic
- Creative, minimal repetition
- Custom

Advanced Settings

Model:

- GPT3.5
- GPT4

- Temperature: 0.2
- Top P: 0.7
- Presence Penalty: 0.8
- Frequency Penalty: 0.8

Some examples of hyperparameters

LV 2. Prompt Engineering with API

Example 1

max_tokens

This hyperparameter controls the maximum length of the output generated by LLMs, measured in tokens.

A token is a *basic unit of language*. It can be as short as a single character or as long as a word. LLMs use tokens to process and generate language.

So max_tokens is like a word count controller. You tweak it to limit how long the LLM's response can be.

When to use

If you want concise responses to customer questions, controlling the length would be very helpful. Nobody wants a meandering reply that wastes their time.

You can also use it to develop a tool that generates concise news summaries. Max_tokens will allow the tool to provide readers with a quick overview of each article's main points without overwhelming them with unnecessary details.

LLMs bill users based on tokens. The more tokens it uses in the output, the more you pay, and if you have large data, the cost stacks up fast. So max_tokens is a great way to control cost as well.

LV 2. Prompt Engineering with API

Example 2

Temperature

The temperature impacts the randomness of the model's output on a scale of 0-1.

A higher value (closer to 1) makes the output more random, or creative, while a lower value (closer to 0) makes the output more deterministic and focused.

When to use

In simple terms, temperature decides whether your LLM output is more creative or factual.

For example, you can use this hyperparameter to generate two sets of marketing copy for A/B testing: one that's more creative and "out there," and another that's more reality-based and practical.

If you're doing creative writing with the help of LLM and want more imaginative storylines, you can set the temperature quite high (e.g., 0.8) -- the LLM might even hallucinate and give you *really* out-of-this-world plots.

LV 2. Prompt Engineering with API

3 Vector database

TL;DR

A vector database is like an Amazon inventory system. Amazon needs to know what's in stock, where it is, and how many. The inventory system keeps track of all this, so employees can quickly find what customers want and restock when necessary.

A vector database is like that, but for words, sentences, and meanings. It keeps track of how words and ideas are related to each other in a mathematical way.

Just like Amazon uses its inventory to quickly find products and keep customers happy, an LLM uses a vector database to quickly find and understand the relationships between words and ideas, helping it communicate and understand language just like a human.



LV 2. Prompt Engineering with API

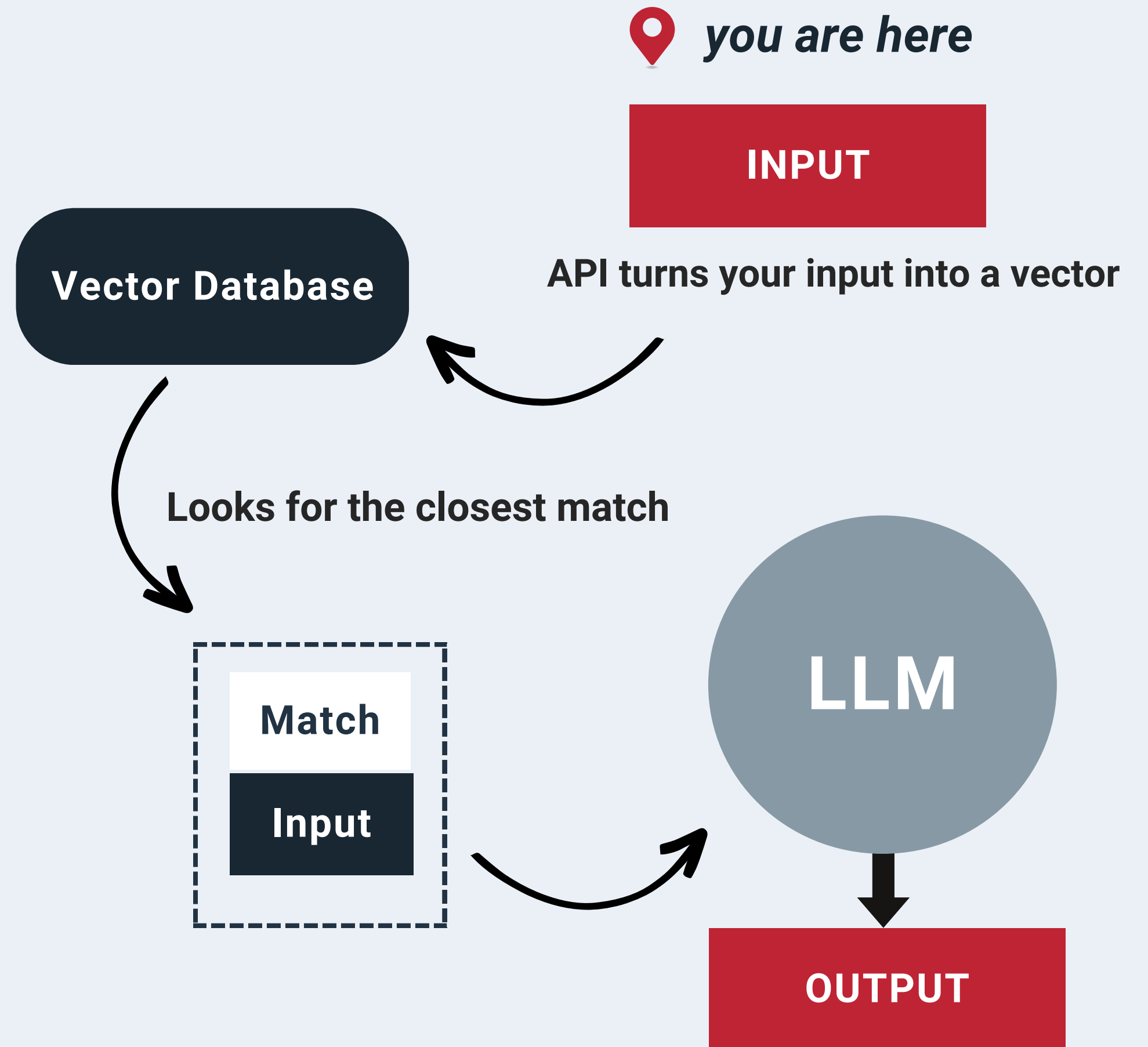
How it works

Let's bring back the input-output graphic.

When you submit an input, your API will turn it into a vector, then go into the vector database to look for the closest match.

Once it finds one, it will prepend the match to your input and pass the whole thing to the LLM. At that point, the LLM will be able to create an answer based on that data.

Still confused? Let's see an example.



LV 2. Prompt Engineering with API

Example

A personal fitness instructor has written 200+ blog posts on topics like how to build an exercise regime, how to achieve a balanced diet, and so on.

Working with a developer and a third-party API, they can upload all these posts to a **vector database**.

Now whenever a client has a question about any of the topics written on the blog, the LLM can respond using the information from the vector database.

It's a way stronger method to retrieve information than keyword search, and will give you much more relevant, contextual results.



LV 2. Prompt Engineering with API

4 AI agents

How it works

AI agents provide applications a fundamentally new set of capabilities, including the ability to solve complex problems and interact with the external world.

An AI agent typically has access to a suite of tools and determines which ones to use depending on the user input. These agents can employ multiple tools simultaneously or sequentially, using the output of one tool as the input for the next.

This process of creating sequences of outputs and inputs is known as "chaining", and it is a potent technique for problem-solving in complex scenarios.



LV 2. Prompt Engineering with API

Example

A small cafe can integrate AI agents to streamline their inventory management.

Using computer vision and machine learning, the AI monitors stock levels of ingredients and supplies in real-time. Through a simple mobile app interface, the cafe staff can prompt the AI with inquiries like, "Check milk supply" or "Predict coffee bean needs for the week."

The AI agent carries out a series of complex tasks, like predicting usage patterns, helping the cafe optimize purchasing, and reducing waste. It can also automatically generate reorder prompts when supplies are running low.



LV 2. Prompt Engineering with API

Exciting developments with AI agents...

OpenAI had their first developer event in November, dropping a series of fun updates.

Most exciting: GPTs, which are essentially AI agents *anyone* can build for their specific use cases. You don't even need a developer! WILD ideas are popping up everywhere...

Notably, OpenAI also launched "Assistants API," which is a small step toward fully autonomous AI agents in the future (this one's for the devs, sorry).



Onto the next level...

LEVEL 3

Fine-tuning



LV 3. Fine-tuning

TL;DR

Fine-tuning is a technique where a pre-trained model is adjusted to perform a new task. You're teaching the LLM how to use the knowledge that it already has to do something new.

It's basically prompt engineering with API, on steroids.

When to use it

Move on to this level *only if*:

- You have large quantities of domain-specific data, and prompt engineering with API didn't yield satisfactory results
- You have the budget — when your training data gets large enough to demand fine-tuning, the cost can add up quickly (we'll get to that later).



LV 3. Fine-tuning

Why it matters

Working with a fine-tuned model is like having a super expert with decades of experience in your industry.

All the knowledge is built inside their minds - they don't need to do research or find stuff in the inventory system (vector database).

You can fine-tune an LLM to do very specific tasks in your industry niche.

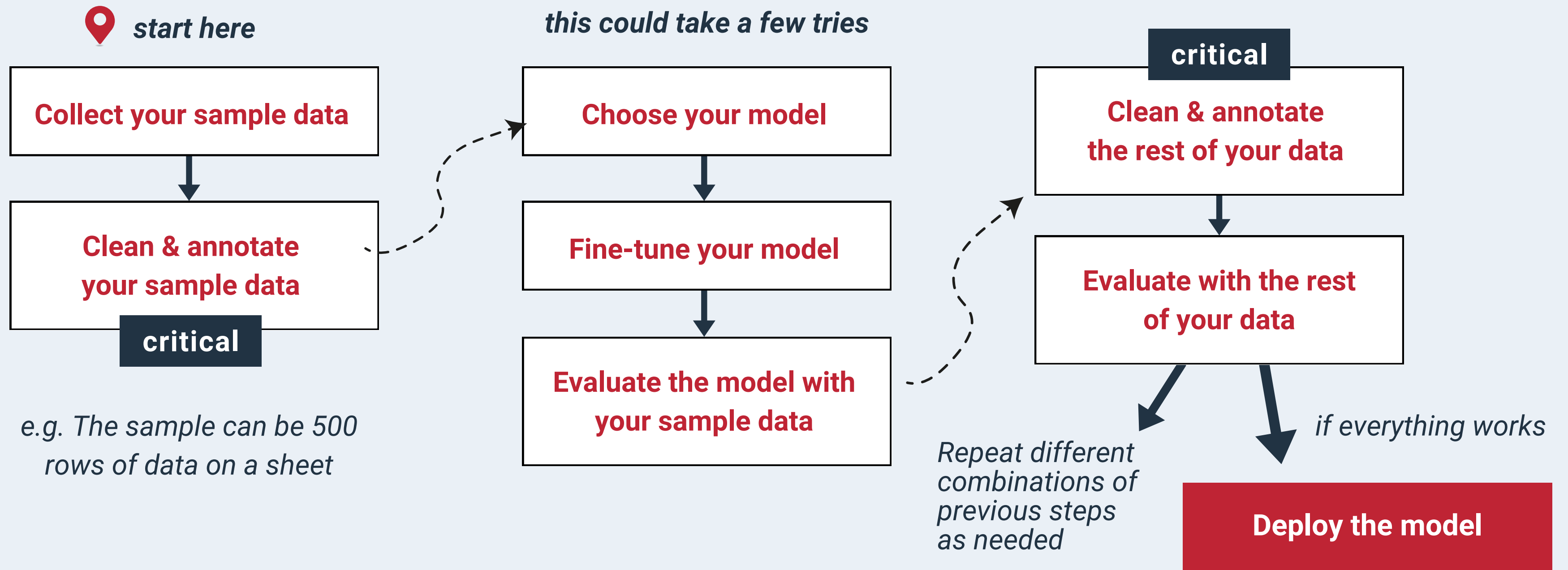
Here are some examples:

Email spam detection	Customer support chatbots
Document Summarization	Social media monitoring
Fake news detection	Resume screening
Legal contract analysis	Content moderation
Stock market analysis	E-commerce product descriptions
Language translation	Virtual assistants
Personalized education	Artificial creativity

LV 3. Fine-tuning

How it works

Fine-tuning a language model is an iterative process, and the model gets better and better with each round of training, as it learns from its mistakes, and adapts to the specifics of the task at hand.



LV 3. Fine-tuning

Cost

In most business use cases, fine-tuning an LLM would cost anywhere from **\$50k-\$300k**.

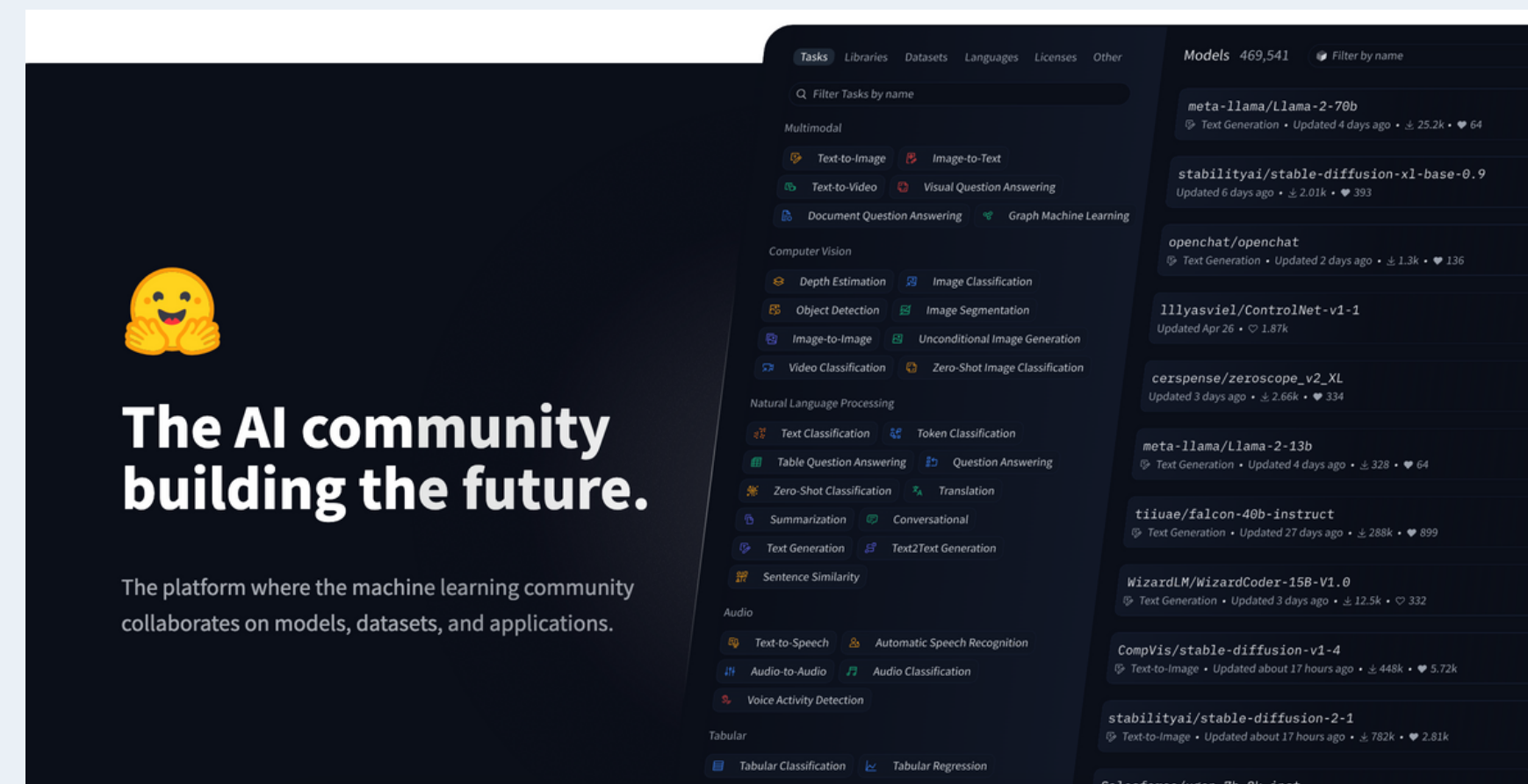
The actual cost of fine-tuning could be ~\$1k, but there are many other associated costs.

Cleaning and annotating your data can run in the low thousands to \$100k+. The same price range applies to integrating the model into your existing ecosystem.

If you want real time insights without compromising quality, finding the best model fit and subsequent optimizations also add to the price.

Don't forget that you need to employ machine learning engineers to test, evaluate, and repeat the process until it works properly. For complex cases, it could take them six months to a year. These engineers are typically paid over \$250k annually -- you do the math.

You can save some money by using open source models. [Hugging Face](#) is an AI community with open-source tools, resources, and pre-trained models to help you fine-tune a model for your specific task.



The image is a composite. On the left is a dark-themed promotional banner with a yellow smiley face emoji with hands clasped. The text on the banner reads: "The AI community building the future." followed by "The platform where the machine learning community collaborates on models, datasets, and applications." On the right is a screenshot of the Hugging Face website interface, showing a search bar, navigation tabs (Tasks, Libraries, Datasets, Languages, Licenses, Other), and a list of models. The model list includes entries like "meta-llama/Llama-2-70b", "stabilityai/stable-diffusion-xl-base-0.9", "openchat/openchat", "lillyasviel/ControlNet-v1-1", "cerspense/zeroscope_v2_XL", "meta-llama/Llama-2-13b", "tiiuae/falcon-40b-instruct", "WizardLM/WizardCoder-15B-V1.0", "CompVis/stable-diffusion-v1-4", "stabilityai/stable-diffusion-2-1", and "Salesforce/xgen-7b-8k-inst".

LV 3. Fine-tuning

Example 1

You can fine-tune a model to provide top-notch customer service.

By fine-tuning the model using your company's own support ticket data, you can create a sophisticated AI chatbot that understands and responds to customer queries more effectively and efficiently.

The fine-tuned LLM can help automatically classify and triage incoming support tickets, responding faster and allocating resources better. It can provide instant, accurate answers to FAQs, reducing the workload on support agents and improving customer satisfaction.

It can also empower your human customer support team by providing them with templates and tips to better handle user complaints.



LV 3. Fine-tuning

Example 2

You can fine-tune a model to generate compelling product descriptions.

By fine-tuning the model using a dataset that includes your company's existing product descriptions, customer feedback, and relevant industry data, you can train it to generate high-quality and persuasive descriptions that are tailored to your brand's voice and style.

The fine-tuned LLM can also generate descriptions that resonate with different customer segments and personas, catering to their specific preferences.

This allows you to streamline the content creation process, enhance the consistency of your brand messaging, and drive better conversion rates by effectively showcasing your products' key selling points.



LEVEL 4

Building Your Own LLM



LV 4. Building Your Own LLM

TL;DR

Let's be real here - it's unlikely you'll need this.

But if you do... Go get a good team of developers. You're playing in the stratosphere.

When to use it

- Large enterprise with a lot of proprietary data -- many companies you know are already doing this.
- Government-specific use cases that must be trained.

Building your own LLM is expensive and time-consuming, but will indeed provide the exact results you want.



Microsoft

 Meta



nVIDIA®



amazon



Final Thoughts



If you remember nothing else, remember these:

- Think about large language models in terms of **input** and **output**. With the tips and knowledge in this guide, you should be able to optimize your input (prompts) to get desirable results from the LLM. Prompting is programming. Conversation is the new interface.
- Simply prompt engineering goes a long way, especially if you play with the techniques that help LLMs learn better and faster. Provide clear, **concise instructions** and examples to make sure you get the desired output. Prompt engineering is a new and evolving field, and **new techniques emerge everyday**.
- Using APIs gives you more control over the LLM's behavior and lets you work with large quantities of data. You can set a Systems Role or tweak the hyperparameters to **adjust a model's "personality."**
- If you have large amounts of **highly domain-specific data**, clean up and annotate them. It will be critical to fine-tuning an LLM to serve your industry niche. With fine-tuned models, you can build sophisticated chatbots that know everything about your company or products.
- Whether you run a large enterprise or a mom-and-pop shop, you could **benefit from using LLMs**. Use it to automate manual tasks, streamline processes, analyze data, or generate content. If used right, the LLM can cut your customer support team in half, 3x your content generation goal, market to your audience better, ship new products every week, and many more.

Our take

- In the near future, anyone sitting on a wealth of data should put it to use with vector database or fine-tuning.
- Artificial general intelligence will be the amalgamation of thousands of small, niche models – a system of interconnected models
- We're going to need people to build these small, niche models. People like teachers, chefs, philosophers, historians, comedians, doctors, artists, and more.
- **AI amplifies what it means to be human.**



Want more resources like this?

Sign up for our free newsletter and gain access to tons of tactical business advice you can **implement today.**



Here's a sneak peek of our new research hub



A screenshot of the Trends website newsletter. At the top, there is a navigation bar with links for "Trends Newsletter: Featured Articles", "Marketing Insights", "Sales & Service Insights", and "Reports & Media". Below this is the Trends logo and the text "Trends Newsletter: Featured Articles". The main content area features three article cards. The first card is titled "How to Slow the Heck Down and Still Crush Your Goals" and includes a "Read Here" button. The second card is titled "How Brands are Cashing in on the Joyconomy" and includes a "Read Here" button. The third card is titled "How Customer Rating Impacts Purchasing Decisions [Data]" and includes a "Read Here" button. At the bottom of the newsletter, there is a "Subscribe to Trends" section with a "Subscribe" button and the Trends logo.