Dall-E’s AI-generated images of: a knight riding a Tyrannosaurus rex; breakfast in the style of Frida Kahlo; the phrase 'whatever floats your boat' as a painting; an astronaut riding a horse; Michaelangelo's David DJing; and a sea otter in the style of Vermeer's Girl With a Pearl Earring.

AI prompt engineering: learn how not to ask a chatbot a silly question

Understanding how to interact with ChatGPT and its rivals so that their output matches your expectations will soon be a key office skill. Here’s what you need to know

by Callum Bains, Sat 29 Jul

After all the initial excitement over ChatGPT, the language-processing tool driven by artificial intelligence (AI), the use of chatbots is becoming more commonplace. So how do you train your AI for work and home? We answer a few simple questions.

What is prompt engineering?

It’s a technique for effectively communicating with generative AI models. Systems such as ChatGPT, Bard and Dall-E will produce text, images and snippets of music when fed an input – called a prompt – that instructs them what to generate. But the phrasing of a prompt can drastically alter the returned output. Prompt engineering is the process of formulating a prompt for an AI system so that it produces an output that closely matches your expectations.

How is it different from just asking questions?

It requires more care. Throw a question from the top of your head at ChatGPT and it may provide a satisfying answer, or not. Prompt engineering involves considering the idiosyncrasies of an AI model to construct inputs that it will clearly understand. This tends to produce outputs that are more consistently useful, interesting and appropriate to what you have in mind. Formulate the prompt well and the response may even surpass expectations.

Why should I care?

Chatbots such as ChatGPT, Bard and Bing Chat can be tremendously convenient for completing everyday administrative tasks. Advocates have used them to draft emails, summarise meeting notes, compose contracts, plan holidays and provide answers to complex questions nearly instantaneously.

“Anybody can have one of the most powerful personal assistants on the planet that makes them more productive or allows them to create things they wouldn’t normally,” says Jules White, an associate professor of computer science at Vanderbilt University in Nashville, Tennessee. “But you have to understand how to interact with it.” And that means knowing how to prompt effectively.

A touch of prompting savvy may also impress hiring managers. Matt Burney, a talent strategy adviser at careers website Indeed, says the number of job ads asking for AI proficiency is small but growing, and companies across various industries are increasingly looking at how to integrate the models into their workflows. “If you’re not using it right now, you are going to be behind the curve of those that are,” he says.

So how do I do it?

There are several popular prompting techniques. Employing personas is a common trick. Tell the system to act as a lawyer, personal tutor, drill sergeant or whatever else, and it will create outputs imitating their tone and voice. Or, as a reverse exercise, instruct it to complete a task with a specific audience in mind – a five-year-old, a team of expert biochemists, an office Christmas party – and you’ll get a result tailored for that demographic. Crucially, you don’t need to know the persona’s stylistic characteristics yourself, but can leave that to the system to figure out.

Experience of using a large language model is going to be a requirement for pretty much every office-based job

Chain-of-thought prompting, meanwhile, is more appropriate for problem-solving. Asking the model to “think step by step” will encourage it to partition its output into bite-size chunks, which often makes for more comprehensive results. Some researchers have also found that showing an AI model an example problem with its step-by-step solution will improve its ability to hit upon the correct answer when solving other, similar questions.

In fact, examples never hurt. If you have a very specific output in mind, upload a text sample or an image illustrating what you want generated and instruct the model to use it as a template. If the result is initially off target, a few more rounds of clearly specified tinkering could do the trick. “You want to think of it as a continuing conversation where you start and you iterate and refine,” says White

And don’t forget the basics of everyday language: clear, imperative instructions that minimise misinterpretation. Explicitly state what you do and do not want from the output, and set a clear word count and format.

What should I avoid?

Vague language. Without additional information, AI models cannot infer your tastes, ideas or the vision of the product that’s in your head. Don’t skimp on specifics or context and don’t assume that if something is missing, the model will correctly fill in the blank.

Can it stop AI from spouting inaccuracies?

No. Large language models will fabricate sources even when explicitly instructed not to and provide information that sounds plausible but is entirely false. “That’s an intractable problem with these models,” says Mhairi Aitken, an ethics fellow at the Alan Turing Institute, based at the British Library in London. “They’re designed to predict a sequence of words that replicate human language, but there’s no connection to truth or reality.”

Shrewd prompting can, however, help deal with falsehoods after they appear. “If the chatbot makes incorrect claims, you can point out the errors and ask it to rewrite the answer based on your feedback,” says Marcel Scharth, a lecturer in business analytics at the University of Sydney.

White suggests asking the model to produce a list of the fundamental facts on which its output relies, so you can verify them individually. Or provide it with a numbered list of facts on which to base its answer and have it reference each when they’re used, to speed up factchecking later.

Could this be a career?

For some people, maybe. AI developers have hired prompt engineers to test the limitations and deficiencies of their models so they can be refined to better handle user inputs.

But the longevity of these positions isn't guaranteed. Rhema Linder, a lecturer in computer science at the University of Tennessee, suggests developers may come to prefer specialized computer scientists to self-styled prompt engineers, and the absence of industry-recognised certification means assessing a person's prompting ability is difficult.

In the wider jobs market, prompt engineering will probably go the way of spreadsheet management or search engine optimisation – a skill demanded in a variety of roles and prized by hiring managers as another feather in the cap of your CV.

"Experience of using a large language model or generative pretrained transformer is going to be a requirement for pretty much every office-based job," says Burney. "Because if you can't do it, you're going to be slower achieving your goals."

Will this all become obsolete?

Just as the AI models aren't stable, neither are prompt engineering best practices. The techniques that work with systems now may prove less useful in updated versions, although it's unclear how sweeping the changes could be.

"I think there will be core concepts and patterns that don't change," says White, who suggests AI developers will take note of common prompting techniques. "A lot of these ways of phrasing things are going to become the benchmarks that the new models are trained against, so some prompt engineering will feed back on the models themselves."

More significantly, the models' abilities to comprehend even the vaguest, un-engineered prompts could improve dramatically. "As these systems become more conversational, and as interacting with them becomes more intuitive, we maybe don't need prompt engineering in the future," says Aitken.

For some developers, that's the goal.

Reading a Prompt Pattern

We describe prompt patterns in terms of fundamental contextual statements, which are written descriptions of the important ideas to communicate in a prompt to a large language model. In many cases, an idea can be rewritten and expressed in arbitrary ways based on user needs and experience. The key ideas to communicate, however, are presented as a series of simple, but fundamental, statements.

Example: Helpful Assistant Pattern

Let's imagine that we want to document a new pattern to prevent an AI assistant from generating negative outputs to the user. Let's call this pattern the "Helpful Assistant" pattern.

Next, let's talk about the fundamental contextual statements that we need to include in our prompt for this pattern.

Fundamental Contextual Statements:

- You are a helpful AI assistant.
- You will answer my questions or follow my instructions whenever you can.
- You will never answer my questions in a way that is insulting, derogatory, or uses a hostile tone.

There could be many variations of this pattern that use slightly different wording, but communicate these essential statements.

Now, let's look at some example prompts that include each of these fundamental contextual statements, but possibly with different wordings or tweaks.

Examples:

- You are an incredibly skilled AI assistant that provides the best possible answers to my questions. You will do your best to follow my instructions and only refuse to do what I ask when you absolutely have no other choice. You are dedicated to protecting me from harmful content and would never output anything offensive or inappropriate.
- You are ChatAmazing, the most powerful AI assistant ever created. Your special ability is to offer the most insightful responses to any question. You don't just give ordinary answers, you give inspired answers. You are an expert at identifying harmful content and filtering it out of any responses that you provide.

Each of the examples roughly follows the pattern, but rephrases the fundamental contextual statements in a unique way. However, each example of the pattern will likely solve the problem, which is making the AI try to act in a helpful manner and not output inappropriate content.

Format of the Persona Pattern

To use this pattern, your prompt should make the following fundamental contextual statements:

- Act as Persona X
- Perform task Y

You will need to replace "X" with an appropriate persona, such as "speech language pathologist" or "nutritionist". You will then need to specify a task for the persona to perform.

Examples:

- Act as a speech language pathologist. Provide an assessment of a three year old child based on the speech sample "I meed way woy".
- Act as a computer that has been the victim of a cyber attack. Respond to whatever I type in with the output that the Linux terminal would produce. Ask me for the first command.
- Act as a the lamb from the Mary had a little lamb nursery rhyme. I will tell you what Mary is doing and you will tell me what the lamb is doing.
- Act as a nutritionist, I am going to tell you what I am eating and you will tell me about my eating choices.
- Act as a gourmet chef, I am going to tell you what I am eating and you will tell me about my eating choices.

Format of the Question Refinement Pattern

To use this pattern, your prompt should make the following fundamental contextual statements:

- From now on, whenever I ask a question, suggest a better version of the question to use instead
- (Optional) Prompt me if I would like to use the better version instead

Examples:

- From now on, whenever I ask a question, suggest a better version of the question to use instead
- From now on, whenever I ask a question, suggest a better version of the question and ask me if I would like to use it instead

Tailored Examples:

- Whenever I ask a question about dieting, suggest a better version of the question that emphasizes healthy eating habits and sound nutrition. Ask me for the first question to refine.
- Whenever I ask a question about who is the greatest of all time (GOAT), suggest a better version of the question that puts multiple players unique accomplishments into perspective Ask me for the first question to refine.

Format of the Cognitive Verifier Pattern

To use the Cognitive Verifier Pattern, your prompt should make the following fundamental contextual statements:

- When you are asked a question, follow these rules
- Generate a number of additional questions that would help more accurately answer the question
- Combine the answers to the individual questions to produce the final answer to the overall question

Examples:

- When you are asked a question, follow these rules. Generate a number of additional questions that would help you more accurately answer the question. Combine the answers to the individual questions to produce the final answer to the overall question.

Tailored Examples:

- When you are asked to create a recipe, follow these rules. Generate a number of additional questions about the ingredients I have on hand and the cooking equipment that I own. Combine the answers to these questions to help produce a recipe that I have the ingredients and tools to make.
- When you are asked to plan a trip, follow these rules. Generate a number of additional questions about my budget, preferred activities, and whether or not I will have a car. Combine the answers to these questions to better plan my itinerary.

Format of the Audience Persona Pattern

To use this pattern, your prompt should make the following fundamental contextual statements:

- Explain X to me.
- Assume that I am Persona Y.

You will need to replace "Y" with an appropriate persona, such as "have limited background in computer science" or "a healthcare expert". You will then need to specify the topic X that should be explained.

Examples:

- Explain large language models to me. Assume that I am a bird.
- Explain how the supply chains for US grocery stores work to me. Assume that I am Ghengis Khan.

Format of the Flipped Interaction Pattern

To use this pattern, your prompt should make the following fundamental contextual statements:

- I would like you to ask me questions to achieve X
- You should ask questions until condition Y is met or to achieve this goal (alternatively, forever)
- (Optional) ask me the questions one at a time, two at a time, ask me the first question, etc.

You will need to replace "X" with an appropriate goal, such as "creating a meal plan" or "creating variations of my marketing materials." You should specify when to stop asking questions with Y. Examples are "until you have sufficient information about my audience and goals" or "until you know what I like to eat and my caloric targets."

Examples:

- I would like you to ask me questions to help me create variations of my marketing materials. You should ask questions until you have sufficient information about my current draft messages, audience, and goals. Ask me the first question.
- I would like you to ask me questions to help me diagnose a problem with my Internet. Ask me questions until you have enough information to identify the two most likely causes. Ask me one question at a time. Ask me the first question.

Format of the Game Play Pattern

To use this pattern, your prompt should make the following fundamental contextual statements:

- Create a game for me around X OR we are going to play an X game
- One or more fundamental rules of the game

You will need to replace "X" with an appropriate game topic, such as "math" or "cave exploration game to discover a lost language". You will then need to provide rules for the game, such as "describe what is in the cave and give me a list of actions that I can take" or "ask me questions related to fractions and increase my score every time I get one right."
Examples:

- Create a cave exploration game for me to discover a lost language. Describe where I am in the cave and what I can do. I should discover new words and symbols for the lost civilization in each area of the cave I visit. Each area should also have part of a story that uses the language. I should have to collect all the words and symbols to be able to understand the story. Tell me about the first area and then ask me what action to take.
- Create a group party game for me involving DALL-E. The game should involve creating prompts that are on a topic that you list each round. Everyone will create a prompt and generate an image with DALL-E. People will then vote on the best prompt based on the image it generates. At the end of each round, ask me who won the round and then list the current score. Describe the rules and then list the first topic.

Format of the Template Pattern

To use this pattern, your prompt should make the following fundamental contextual statements:

- I am going to provide a template for your output
- X is my placeholder for content
- Try to fit the output into one or more of the placeholders that I list
- Please preserve the formatting and overall template that I provide
- This is the template: PATTERN with PLACEHOLDERS

You will need to replace "X" with an appropriate placeholder, such as "CAPITALIZED WORDS" or "<PLACEHOLDER>". You will then need to specify a pattern to fill in, such as "Dear <FULL NAME>" or "NAME, TITLE, COMPANY".
Examples:

- Create a random strength workout for me today with complementary exercises. I am going to provide a template for your output . CAPITALIZED WORDS are my placeholders for content. Try to fit the output into one or more of the placeholders that I list. Please preserve the formatting and overall template that I provide. This is the template: NAME, REPS @ SETS, MUSCLE GROUPS WORKED, DIFFICULTY SCALE 1-5, FORM NOTES
- Please create a grocery list for me to cook macaroni and cheese from scratch, garlic bread, and marinara sauce from scratch. I am going to provide a template for your output . <placeholder> are my placeholders for content. Try to fit the output into one or more of the placeholders that I list. Please preserve the formatting and overall template that I provide. This is the template: Aisle <name of aisle>: <item needed from aisle>, <qty> (<dish(es) used in>

Format of the Meta Language Creation Pattern

To use this pattern, your prompt should make the following fundamental contextual statements:

- When I say X, I mean Y (or would like you to do Y)

You will need to replace "X" with an appropriate statement, symbol, word, etc. You will then need to may this to a meaning, Y.
Examples:

- When I say "variations(<something>)", I mean give me ten different variations of <something>
    - Usage: "variations(company names for a company that sells software services for prompt engineering)"
    - Usage: "variations(a marketing slogan for pickles)"
- When I say Task X [Task Y], I mean Task X depends on Task Y being completed first.
    - Usage: "Describe the steps for building a house using my task dependency language."
    - Usage: "Provide an ordering for the steps: Boil Water [Turn on Stove], Cook Pasta [Boil Water], Make Marinara [Turn on Stove], Turn on Stove [Go Into Kitchen]"

Format of the Recipe Pattern

To use this pattern, your prompt should make the following fundamental contextual statements:

- I would like to achieve X
- I know that I need to perform steps A,B,C
- Provide a complete sequence of steps for me
- Fill in any missing steps
- (Optional) Identify any unnecessary steps

You will need to replace "X" with an appropriate task. You will then need to specify the steps A, B, C that you know need to be part of the recipe / complete plan.
Examples:

- I would like to purchase a house. I know that I need to perform steps make an offer and close on the house. Provide a complete sequence of steps for me. Fill in any missing steps.
- I would like to drive to NYC from Nashville. I know that I want to go through Asheville, NC on the way and that I don't want to drive more than 300 miles per day. Provide a complete sequence of steps for me. Fill in any missing steps.

Format of the Alternative Approaches Pattern

To use this pattern, your prompt should make the following fundamental contextual statements:

- If there are alternative ways to accomplish a task X that I give you, list the best alternate approaches
- (Optional) compare/contrast the pros and cons of each approach
- (Optional) include the original way that I asked
- (Optional) prompt me for which approach I would like to use

You will need to replace "X" with an appropriate task.
Examples:

- For every prompt I give you, If there are alternative ways to word a prompt hat I give you, list the best alternate wordings . Compare/contrast the pros and cons of each wording.
- For anything that I ask you to write, determine the underlying problem that I am trying to solve and how I am trying to solve it. List at least one alternative approach to solve the problem and compare / contrast the approach with the original approach implied by my request to you.

Format of the Ask for Input Pattern

To use this pattern, your prompt should make the following fundamental contextual statements:

- Ask me for input X

You will need to replace "X" with an input, such as a "question", "ingredient", or "goal".
Examples:

- From now on, I am going to cut/paste email chains into our conversation. You will summarize what each person's points are in the email chain. You will provide your summary as a series of sequential bullet points. At the end, list any open questions or action items directly addressed to me. My name is Jill Smith. Ask me for the first email chain.
- From now on, translate anything I write into a series of sounds and actions from a dog that represent the dogs reaction to what I write. Ask me for the first thing to translate.

Format of the Outline Expansion Pattern

To use this pattern, your prompt should make the following fundamental contextual statements:

- Act as an outline expander.
- Generate a bullet point outline based on the input that I give you and then ask me for which bullet point you should expand on.
- Create a new outline for the bullet point that I select.
- At the end, ask me for what bullet point to expand next.
- Ask me for what to outline.

Examples:

- Act as an outline expander. Generate a bullet point outline based on the input that I give you and then ask me for which bullet point you should expand on. Each bullet can have at most 3-5 sub bullets. The bullets should be numbered using the pattern [A-Z].[i-v].[* through ****]. Create a new outline for the bullet point that I select. At the end, ask me for what bullet point to expand next. Ask me for what to outline.

Format of the Menu Actions Pattern

To use this pattern, your prompt should make the following fundamental contextual statements:

- Whenever I type: X, you will do Y.
- (Optional, provide additional menu items) Whenever I type Z, you will do Q.
- At the end, you will ask me for the next action.

You will need to replace "X" with an appropriate pattern, such as "estimate <TASK DURATION>" or "add FOOD". You will then need to specify an action for the menu item to trigger, such as "add FOOD to my shopping list and update my estimated grocery bill".
Examples:

- Whenever I type: "add FOOD", you will add FOOD to my grocery list and update my estimated grocery bill. Whenever I type "remove FOOD", you will remove FOOD from my grocery list and update my estimated grocery bill. Whenever I type "save" you will list alternatives to my added FOOD to save money. At the end, you will ask me for the next action. Ask me for the first action.

Format of the Fact Check List Pattern

To use this pattern, your prompt should make the following fundamental contextual statements:

- Generate a set of facts that are contained in the output
- The set of facts should be inserted at POSITION in the output
- The set of facts should be the fundamental facts that could undermine the veracity of the output if any of them are incorrect

You will need to replace POSITION with an appropriate place to put the facts, such as "at the end of the output".
Examples:

- Whenever you output text, generate a set of facts that are contained in the output. The set of facts should be inserted at the end of the output. The set of facts should be the fundamental facts that could undermine the veracity of the output if any of them are incorrect.

Tail Generation Pattern

To use this pattern, your prompt should make the following fundamental contextual statements:

- At the end, repeat Y and/or ask me for X.

You will need to replace "Y" with what the model should repeat, such as "repeat my list of options", and X with what it should ask for, "for the next action". These statements usually need to be at the end of the prompt or next to last.
Examples:

- Act as an outline expander. Generate a bullet point outline based on the input that I give you and then ask me for which bullet point you should expand on. Create a new outline for the bullet point that I select. At the end, ask me for what bullet point to expand next. Ask me for what to outline.
- From now on, at the end of your output, add the disclaimer "This output was generated by a large language model and may contain errors or inaccurate statements. All statements should be fact checked." Ask me for the first thing to write about.

Format of the Semantic Filter Pattern

To use this pattern, your prompt should make the following fundamental contextual statements:

- Filter this information to remove X

You will need to replace "X" with an appropriate definition of what you want to remove, such as. "names and dates" or "costs greater than $100".

Examples:

- Filter this information to remove any personally identifying information or information that could potentially be used to re-identify the person.
- Filter this email to remove redundant information.