Title: Prompt engineering

URL: https://en.wikipedia.org/wiki/Prompt_engineering

PageID: 69071767

Categories: Category:2022 neologisms, Category:Deep learning, Category:Generative artificial intelligence, Category:Linguistics, Category:Machine learning, Category:Natural language processing, Category:Unsupervised learning

---

Prompt engineering is the process of structuring or crafting an instruction in order to produce better outputs from a generative artificial intelligence (AI) model. [1]

A prompt is natural language text describing the task that an AI should perform. [2] A prompt for a text-to-text language model can be a query, a command, or a longer statement including context, instructions, and conversation history. Prompt engineering may involve phrasing a query, specifying a style, choice of words and grammar, [3] providing relevant context, or describing a character for the AI to mimic. [1]

When communicating with a text-to-image or a text-to-audio model, a typical prompt is a description of a desired output such as "a high-quality photo of an astronaut riding a horse" [4] or "Lo-fi slow BPM electro chill with organic samples". [5] Prompting a text-to-image model may involve adding, removing, or emphasizing words to achieve a desired subject, style, layout, lighting, and aesthetic. [6]

History

In 2018, researchers first proposed that all previously separate tasks in natural language processing (NLP) could be cast as a question-answering problem over a context. In addition, they trained a first single, joint, multi-task model that would answer any task-related question like "What is the sentiment" or "Translate this sentence to German" or "Who is the president?" [7]

The AI boom saw an increase in the amount of "prompting technique" to get the model to output the desired outcome and avoid nonsensical output , a process characterized by trial-and-error . [8] After the release of ChatGPT in 2022, prompt engineering was soon seen as an important business skill, albeit one with an uncertain economic future. [1]

A repository for prompts reported that over 2,000 public prompts for around 170 datasets were available in February 2022. [9] In 2022, the chain-of-thought prompting technique was proposed by Google researchers. [10] [11] In 2023, several text-to-text and text-to-image prompt databases were made publicly available. [12] [13] The Personalized Image-Prompt (PIP) dataset, a generated image-text dataset that has been categorized by 3,115 users, has also been made available publicly in 2024. [14]

Text-to-text

Multiple distinct prompt engineering techniques have been published.

Chain-of-thought

According to Google Research, chain-of-thought (CoT) prompting is a technique that allows large language models (LLMs) to solve a problem as a series of intermediate steps before giving a final answer. In 2022, Google Brain reported that chain-of-thought prompting improves reasoning ability by inducing the model to answer a multi-step problem with steps of reasoning that mimic a train of thought . [10] [15] Chain-of-thought techniques were developed to help LLMs handle multi-step reasoning tasks, such as arithmetic or commonsense reasoning questions. [16] [17]

For example, given the question, "Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?", Google claims that a CoT prompt might induce the LLM to answer "A: The cafeteria had 23 apples originally. They used 20 to make lunch.

So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9." [10] When applied to PaLM , a 540 billion parameter language model , according to Google, CoT prompting significantly aided the model, allowing it to perform comparably with task-specific fine-tuned models on several tasks, achieving state-of-the-art results at the time on the GSM8K mathematical reasoning benchmark . [10] It is possible to fine-tune models on CoT reasoning datasets to enhance this capability further and stimulate better interpretability . [18] [19]

As originally proposed by Google, [10] each CoT prompt is accompanied by a set of input/output examples—called exemplars —to demonstrate the desired model output, making it a few-shot prompting technique. However, according to a later paper from researchers at Google and the University of Tokyo , simply appending the words "Let's think step-by-step" [20] was also effective, which allowed for CoT to be employed as a zero-shot technique.

An example format of few-shot CoT prompting with in-context exemplars: [21]

An example format of zero-shot CoT prompting: [20]

In-context learning

In-context learning , refers to a model's ability to temporarily learn from prompts. For example, a prompt may include a few examples for a model to learn from, such as asking the model to complete " maison → house, chat → cat, chien →" (the expected response being dog), [22] an approach called few-shot learning . [23]

In-context learning is an emergent ability [24] of large language models. It is an emergent property of model scale, meaning that breaks [25] in downstream scaling laws occur, leading to its efficacy increasing at a different rate in larger models than in smaller models. [24] [10] Unlike training and fine-tuning , which produce lasting changes, in-context learning is temporary. [26] Training models to perform in-context learning can be viewed as a form of meta-learning , or "learning to learn". [27]

Self-Consistency

Self-Consistency performs several chain-of-thought rollouts, then selects the most commonly reached conclusion out of all the rollouts. [28] [29]

Tree-of-thought

Tree-of-thought prompting generalizes chain-of-thought by generating multiple lines of reasoning in parallel, with the ability to backtrack or explore other paths. It can use tree search algorithms like breadth-first , depth-first , or beam . [29] [30]

Prompting to estimate model sensitivity

Research consistently demonstrates that LLMs are highly sensitive to subtle variations in prompt formatting, structure, and linguistic properties. Some studies have shown up to 76 accuracy points across formatting changes in few-shot settings. [31] Linguistic features significantly influence prompt effectiveness—such as morphology, syntax, and lexico-semantic changes—which meaningfully enhance task performance across a variety of tasks. [3] [32] Clausal syntax, for example, improves consistency and reduces uncertainty in knowledge retrieval. [33] This sensitivity persists even with larger model sizes, additional few-shot examples, or instruction tuning.

To address sensitivity of models and make them more robust, several methods have been proposed. FormatSpread facilitates systematic analysis by evaluating a range of plausible prompt formats, offering a more comprehensive performance interval. [31] Similarly, PromptEval estimates performance distributions across diverse prompts, enabling robust metrics such as performance quantiles and accurate evaluations under constrained budgets. [34]

Automatic prompt generation

Retrieval-augmented generation

Retrieval-augmented generation (RAG) is a technique that enables generative artificial intelligence (Gen AI) models to retrieve and incorporate new information. It modifies interactions with a large language model (LLM) so that the model responds to user queries with reference to a specified set

of documents, using this information to supplement information from its pre-existing training data . This allows LLMs to use domain-specific and/or updated information. [35]

RAG improves large language models (LLMs) by incorporating information retrieval before generating responses. Unlike traditional LLMs that rely on static training data, RAG pulls relevant text from databases, uploaded documents, or web sources. According to Ars Technica , "RAG is a way of improving LLM performance, in essence by blending the LLM process with a web search or other document look-up process to help LLMs stick to the facts." This method helps reduce AI hallucinations , which have led to real-world issues like chatbots inventing policies or lawyers citing nonexistent legal cases. By dynamically retrieving information, RAG enables AI to provide more accurate responses without frequent retraining. [36]

Graph retrieval-augmented generation

GraphRAG (coined by Microsoft Research) is a technique that extends RAG with the use of a knowledge graph (usually, LLM-generated) to allow the model to connect disparate pieces of information, synthesize insights, and holistically understand summarized semantic concepts over large data collections. It was shown to be effective on datasets like the Violent Incident Information from News Articles (VIINA). [37] [38]

Earlier work showed the effectiveness of using a knowledge graph for question answering using text-to-query generation. [39] These techniques can be combined to search across both unstructured and structured data, providing expanded context, and improved ranking.

Using language models to generate prompts

Large language models (LLM) themselves can be used to compose prompts for large language models. [40] The automatic prompt engineer algorithm uses one LLM to beam search over prompts for another LLM: [41] [42]

There are two LLMs. One is the target LLM, and another is the prompting LLM.

Prompting LLM is presented with example input-output pairs, and asked to generate instructions that could have caused a model following the instructions to generate the outputs, given the inputs.

Each of the generated instructions is used to prompt the target LLM, followed by each of the inputs. The log-probabilities of the outputs are computed and added. This is the score of the instruction.

The highest-scored instructions are given to the prompting LLM for further variations.

Repeat until some stopping criteria is reached, then output the highest-scored instructions.

CoT examples can be generated by LLM themselves. In "auto-CoT", a library of questions are converted to vectors by a model such as BERT . The question vectors are clustered . Questions close to the centroid of each cluster are selected, in order to have a subset of diverse questions. An LLM does zero-shot CoT on each selected question. The question and the corresponding CoT answer are added to a dataset of demonstrations. These diverse demonstrations can then added to prompts for few-shot learning. [43]

Text-to-image

In 2022, text-to-image models like DALL-E 2 , Stable Diffusion , and Midjourney were released to the public. These models take text prompts as input and use them to generate images. [44] [6]

Prompt formats

Early text-to-image models typically do not understand negation, grammar and sentence structure in the same way as large language models , and may thus require a different set of prompting techniques. The prompt "a party with no cake" may produce an image including a cake. [45] As an alternative, negative prompts allow a user to indicate, in a separate prompt, which terms should not appear in the resulting image. [46] Techniques such as framing the normal prompt into a sequence-to-sequence language modeling problem can be used to automatically generate an output for the negative prompt. [47]

A text-to-image prompt commonly includes a description of the subject of the art, the desired medium (such as digital painting or photography), style (such as hyperrealistic or pop-art), lighting (such as rim lighting or crepuscular rays), color, and texture. [48] Word order also affects the output of a text-to-image prompt. Words closer to the start of a prompt may be emphasized more heavily. [49]

The Midjourney documentation encourages short, descriptive prompts: instead of "Show me a picture of lots of blooming California poppies, make them bright, vibrant orange, and draw them in an illustrated style with colored pencils", an effective prompt might be "Bright orange California poppies drawn with colored pencils". [45]

Artist styles

Some text-to-image models are capable of imitating the style of particular artists by name. For example, the phrase in the style of Greg Rutkowski has been used in Stable Diffusion and Midjourney prompts to generate images in the distinctive style of Polish digital artist Greg Rutkowski . [50] Famous artists such as Vincent van Gogh and Salvador Dalí have also been used for styling and testing. [51]

Non-text prompts

Some approaches augment or replace natural language text prompts with non-text input.

Textual inversion and embeddings

For text-to-image models, textual inversion performs an optimization process to create a new word embedding based on a set of example images. This embedding vector acts as a "pseudo-word" which can be included in a prompt to express the content or style of the examples. [52]

Image prompting

In 2023, Meta 's AI research released Segment Anything, a computer vision model that can perform image segmentation by prompting. As an alternative to text prompts, Segment Anything can accept bounding boxes, segmentation masks, and foreground/background points. [53]

Using gradient descent to search for prompts

In "prefix-tuning", [54] "prompt tuning", or "soft prompting", [55] floating-point-valued vectors are searched directly by gradient descent to maximize the log-likelihood on outputs.

Formally, let E = { e 1 , … , e k } {\displaystyle \mathbf {E} =\{\mathbf {e_{1}} ,\dots ,\mathbf {e_{k}} \}} be a set of soft prompt tokens (tunable embeddings), while X = { x 1 , … , x m } {\displaystyle \mathbf {X} =\{\mathbf {x_{1}} ,\dots ,\mathbf {x_{m}} \}} and Y = { y 1 , … , y n } {\displaystyle \mathbf {Y} =\{\mathbf {y_{1}} ,\dots ,\mathbf {y_{n}} \}} be the token embeddings of the input and output respectively. During training, the tunable embeddings, input, and output tokens are concatenated into a single sequence concat (E ; X ; Y) {\displaystyle {\text{concat}}(\mathbf {E} ;\mathbf {X} ;\mathbf {Y})} , and fed to the LLMs. The losses are computed over the Y {\displaystyle \mathbf {Y} } tokens; the gradients are backpropagated to prompt-specific parameters: in prefix-tuning, they are parameters associated with the prompt tokens at each layer; in prompt tuning, they are merely the soft tokens added to the vocabulary. [56]

More formally, this is prompt tuning. Let an LLM be written as L L M (X) = F (E (X)) {\displaystyle LLM(X)=F(E(X))} , where X {\displaystyle X} is a sequence of linguistic tokens, E {\displaystyle E} is the token-to-vector function, and F {\displaystyle F} is the rest of the model. In prefix-tuning, one provides a set of input-output pairs { (X i , Y i) } i {\displaystyle \{(X^{i},Y^{i})\}_{i}} , and then use gradient descent to search for arg ■ max Z ~ ∑ i log ■ P r [Y i | Z ~ ∗ E (X i)] {\displaystyle \arg \max _{\tilde {Z}}\sum _{i}\log Pr[Y^{i}|{\tilde {Z}}\ast E(X^{i})]} . In words, log ■ P r [Y i | Z ~ ∗ E (X i)] {\displaystyle \log Pr[Y^{i}|{\tilde {Z}}\ast E(X^{i})]} is the log-likelihood of outputting Y i {\displaystyle Y^{i}} , if the model first encodes the input X i {\displaystyle X^{i}} into the vector E (X i) {\displaystyle E(X^{i})} , then prepend the vector with the "prefix vector" Z ~ {\displaystyle {\tilde {Z}}} , then apply F {\displaystyle F} . For prefix tuning, it is similar, but the "prefix vector" Z ~ {\displaystyle {\tilde {Z}}} is pre-appended to the hidden states in every layer of the model. [citation needed]

An earlier result uses the same idea of gradient descent search, but is designed for masked language models like BERT, and searches only over token sequences, rather than numerical vectors. Formally, it searches for arg ■ max X ~ ∑ i log ■ P r [Y i | X ~ ∗ X i] {\displaystyle \arg \max _{\tilde {X}}\sum _{i}\log Pr[Y^{i}|{\tilde {X}}\ast X^{i}]} where X ~ {\displaystyle {\tilde {X}}} is ranges over token sequences of a specified length. [57]

Limitations

While the process of writing and refining a prompt for an LLM or generative AI shares some parallels with an iterative engineering design process, such as through discovering 'best principles' to reuse and discovery through reproducible experimentation, the actual learned principles and skills depend heavily on the specific model being learned rather than being generalizable across the entire field of prompt-based generative models. Such patterns are also volatile and exhibit significantly different results from seemingly insignificant prompt changes. [58] [59] According to The Wall Street Journal in 2025, the job of prompt engineer was one of the hottest in 2023, but has become obsolete due to models that better intuit user intent and to company trainings. [60]

Prompt injection

Prompt injection is a cybersecurity exploit in which adversaries craft inputs that appear legitimate but are designed to cause unintended behavior in machine learning models , particularly large language models (LLMs). This attack takes advantage of the model's inability to distinguish between developer-defined prompts and user inputs, allowing adversaries to bypass safeguards and influence model behaviour. While LLMs are designed to follow trusted instructions, they can be manipulated into carrying out unintended responses through carefully crafted inputs. [61] [62]

References

v

t

e

Autoencoder

Deep learning

Fine-tuning

Foundation model

Generative adversarial network

Generative pre-trained transformer

Large language model

Model Context Protocol

Neural network

Prompt engineering

Reinforcement learning from human feedback

Retrieval-augmented generation

Self-supervised learning

Stochastic parrot

Synthetic data

Top-p sampling

Transformer

Variational autoencoder

Vibe coding
Vision transformer
Waluigi effect
Word embedding
Character.ai
ChatGPT
DeepSeek
Ernie
Gemini
Grok
Copilot
Claude
Gemini
Gemma
GPT 1 2 3 J 4 4o 4.5 4.1 OSS 5
1
2
3
J
4
4o
4.5
4.1
OSS
5
Llama
o1
o3
o4-mini
Qwen
Base44
Claude Code
Cursor
Devstral
GitHub Copilot
Kimi-Dev
Qwen3-Coder
Replit

Xcode
Aurora
Firefly
Flux
GPT Image 1
Ideogram
Imagen
Midjourney
Qwen-Image
Recraft
Seedream
Stable Diffusion
Dream Machine
Hailuo AI
Kling
Midjourney Video
Runway Gen
Seedance
Sora
Veo
Wan
15.ai
Eleven
MiniMax Speech 2.5
WaveNet
Eleven Music
Endel
Lyria
Riffusion
Suno AI
Udio
Agentforce
AutoGLM
AutoGPT
ChatGPT Agent
Devin AI
Manus
OpenAI Codex

Operator
Replit Agent
01.AI
Aleph Alpha
Anthropic
Baichuan
Canva
Cognition AI
Cohere
Contextual AI
DeepSeek
ElevenLabs
Google DeepMind
HeyGen
Hugging Face
Inflection AI
Krikey AI
Kuaishou
Luma Labs
Meta AI
MiniMax
Mistral AI
Moonshot AI
OpenAI
Perplexity AI
Runway
Safe Superintelligence
Salesforce
Scale AI
SoundHound
Stability AI
Synthesia
Thinking Machines Lab
Upstage
xAI
Z.ai
Category
v

t

e

History timeline

timeline

Companies

Projects

Parameter Hyperparameter

Hyperparameter

Loss functions

Regression Bias–variance tradeoff Double descent Overfitting

Bias–variance tradeoff

Double descent

Overfitting

Clustering

Gradient descent SGD Quasi-Newton method Conjugate gradient method

SGD

Quasi-Newton method

Conjugate gradient method

Backpropagation

Attention

Convolution

Normalization Batchnorm

Batchnorm

Activation Softmax Sigmoid Rectifier

Softmax

Sigmoid

Rectifier

Gating

Weight initialization

Regularization

Datasets Augmentation

Augmentation

Prompt engineering

Reinforcement learning Q-learning SARSA Imitation Policy gradient

Q-learning

SARSA

Imitation

Policy gradient

Diffusion

Latent diffusion model

Autoregression

Adversary

RAG

Uncanny valley

RLHF

Self-supervised learning

Reflection

Recursive self-improvement

Hallucination

Word embedding

Vibe coding

Machine learning In-context learning

In-context learning

Artificial neural network Deep learning

Deep learning

Language model Large language model NMT

Large language model

NMT

Reasoning language model

Model Context Protocol

Intelligent agent

Artificial human companion

Humanity's Last Exam

Artificial general intelligence (AGI)

AlexNet

WaveNet

Human image synthesis

HWR

OCR

Computer vision

Speech synthesis 15.ai ElevenLabs

15.ai

ElevenLabs

Speech recognition Whisper

Whisper

Facial recognition

AlphaFold

Text-to-image models Aurora DALL-E Firefly Flux Ideogram Imagen Midjourney Recraft Stable Diffusion

Aurora

DALL-E

Firefly

Flux

Ideogram

Imagen

Midjourney

Recraft

Stable Diffusion

Text-to-video models Dream Machine Runway Gen Hailuo AI Kling Sora Veo

Dream Machine

Runway Gen

Hailuo AI

Kling

Sora

Veo

Music generation Riffusion Suno AI Udio

Riffusion

Suno AI

Udio

Word2vec

Seq2seq

GloVe

BERT

T5

Llama

Chinchilla AI

PaLM

GPT 1 2 3 J ChatGPT 4 4o o1 o3 4.5 4.1 o4-mini 5

1

2

3

J

ChatGPT

4

4o

o1

o3

4.5

4.1

o4-mini

5

Claude

Gemini Gemini (language model) Gemma

Gemini (language model)

Gemma

Grok

LaMDA

BLOOM

DBRX

Project Debater

IBM Watson

IBM Watsonx

Granite

PanGu-Σ

DeepSeek

Qwen

AlphaGo

AlphaZero

OpenAI Five

Self-driving car

MuZero

Action selection AutoGPT

AutoGPT

Robot control

Alan Turing

Warren Sturgis McCulloch

Walter Pitts

John von Neumann

Claude Shannon

Shun'ichi Amari

Kunihiko Fukushima

Takeo Kanade

Marvin Minsky
John McCarthy
Nathaniel Rochester
Allen Newell
Cliff Shaw
Herbert A. Simon
Oliver Selfridge
Frank Rosenblatt
Bernard Widrow
Joseph Weizenbaum
Seymour Papert
Seppo Linnainmaa
Paul Werbos
Geoffrey Hinton
John Hopfield
Jürgen Schmidhuber
Yann LeCun
Yoshua Bengio
Lotfi A. Zadeh
Stephen Grossberg
Alex Graves
James Goodnight
Andrew Ng
Fei-Fei Li
Alex Krizhevsky
Ilya Sutskever
Oriol Vinyals
Quoc V. Le
Ian Goodfellow
Demis Hassabis
David Silver
Andrej Karpathy
Ashish Vaswani
Noam Shazeer
Aidan Gomez
John Schulman
Mustafa Suleyman
Jan Leike

Daniel Kokotajlo

François Chollet

Neural Turing machine

Differentiable neural computer

Transformer Vision transformer (ViT)

Vision transformer (ViT)

Recurrent neural network (RNN)

Long short-term memory (LSTM)

Gated recurrent unit (GRU)

Echo state network

Multilayer perceptron (MLP)

Convolutional neural network (CNN)

Residual neural network (RNN)

Highway network

Mamba

Autoencoder

Variational autoencoder (VAE)

Generative adversarial network (GAN)

Graph neural network (GNN)

Category