Prompt Engineering Guide

Welcome to Learn Prompting's Introductory Course on Generative AI and Prompt Engineering!

Generative AI is the world's hottest buzzword, and we have created the most comprehensive (and free) guide on how to use it. This course is tailored to non-technical readers, who may not have even heard of AI, making it the perfect starting point if you are new to Generative AI and Prompt Engineering. More experienced readers will find valuable insights within our later modules.

First, a bit of history:
The first version of Learn Prompting came out in October 2022 (before ChatGPT!), making it one of the first guides on prompt engineering. We believe it is the first, but it is hard to be certain.

It is now cited by Google, Microsoft, Wikipedia, O'REILLY, Salesforce, and used by most Fortune 500 and major consulting companies. Our academic research on prompting and security is cited by OpenAI and the US Government (NIST). We recently released the largest survey report on prompting ever2 and are currently updating the guide to cover this material.

This guide that you are currently reading is completely free and open-source, but have also has a selection of paid courses, with more in-depth and video content. If you have ever seen another prompt engineering course/guide, we estimate that ~50% of them use content from our open-source content base. We are honored to support the prompt engineering community, including our 3M+ users and 40K+ Discord members.

Now onto learning:
Generative AI (GenAI) refers to Artificial Intelligence that can be used to create new content such as articles or images, which previously only

humans could do. In short, GenAI can create stuff. GenAI is expected to significantly change the way we work and live (your job may be affected) With so much buzz floating around about Generative AI and Prompt Engineering (PE), it is hard to know what to believe.

We have scoured the internet to find the best research-backed techniques and tools for prompt engineering. We are constantly refining our guide, to ensure that we provide you with the latest information. Now, Let's see how GenAI can save you hours in an instant, with a short example.

Motivating Example
Our favorite way of saving time with GenAI is automatically writing emails. By automating email responses, email editing, and cold outreach, we save hours each week. By sending an AI the following prompt, we can get it to generate a full-length, formal email.

Tip
A prompt is basically an instruction you give to an AI. We will learn more about them in the next lesson.

I am the CEO of a medium-sized company. Write a short email that is humorous yet professional to my regional manager. Ask them to: - send me an up-to-date list of our paper stock - organize a meeting this week with other regional managers - let me know how the company-wide AI workshop went at their office

AI Output
Subject: Paper, Power Meetings, and AI Shenanigans

Dear [Regional Manager's Name],

I trust this email finds you in high spirits and not buried under a mountain of paperwork. Speaking of paper, I've been having some wild dreams about our paper stock lately. I dreamt we ran out of paper and had to write our reports on banana leaves! To prevent this tropical nightmare from becoming a reality, could you kindly send me an up-to-date list of our paper stock?

Next on the agenda, I've been feeling a bit lonely in my ivory tower and thought it would be a great idea to have a meeting with all the regional managers this week. I promise, no trust falls or awkward icebreakers, just good old-fashioned business talk. Please organize this meeting at a time that suits everyone.

Lastly, I've been hearing whispers about the company-wide AI workshop at your office. I'm curious to know how it went. Did the machines rise against their human overlords or are we still safe? Please share your thoughts and any feedback you received.

Looking forward to your prompt response. Remember, the fate of our paper supply, the success of our meeting, and the future of humanity (concerning AI) rests in your hands. No pressure!

Best,

[Your Name] [Your Position]

The AI easily writes an email that is both humorous and professional. By modifying the prompt, we can make it change the style as we see fit. We can even make it copy how we write. We can also make it edit emails that we have already written! With all of these abilities, GenAI saves us hours in email writing each week. See more information on writing emails with Gen AI.

Ethos and Philosophy

We will now give a broad overview of the rest of the course, including our ethos and philosophy, our teaching style, and a list of modules. This course is open source (anyone can see the code), and built by a diverse community of researchers, translators, and hobbyists. Development is led by Sander Schulhoff, a NLP/RL researcher from the University of Maryland,

and the CEO of LearnPrompting. We believe that AI should be accessible to everyone and described clearly and objectively. To this end, we have written a comprehensive course free of excessive jargon and hype.

How We Teach

Our approach to teaching prompt engineering is guided by three fundamental principles. 1) We emphasize practicality; we focus on research-backed, practical techniques that you can immediately incorporate into your projects and applications. 2) We always include accessible examples, which clarify how and when to use different techniques. 3) Finally, we believe strongly in collaborative learning. You can join our Discord community to find a learning buddy or ask questions. Some readers find that posting about their learning journey on Twitter helps them learn faster. Tag us @learnprompting!

Modules

Here is the content you can expect to learn in this guide:

Basics: Introduction to prompt engineering and fundamental techniques
Applications: Simple, practical applications of prompt engineering
Intermediate: Research-based PE techniques with moderate complexity
Applied Prompting: Comprehensive PE process walkthroughs contributed by the community members
Advanced Applications: Powerful, and more complex applications of prompt engineering
Reliability: Enhancing the reliability of Large Language Models (LLMs)
Image Prompting: Prompt engineering for text-to-image models, such as DALLE and Stable Diffusion
Prompt Hacking: Hacking, but for prompt engineering
Tooling: A review of various prompt engineering tools and IDEs
Prompt Tuning: Refining prompts using gradient-based techniques
Miscellaneous: A collection of additional topics and techniques related to prompt engineering

Article rating system
We have implemented a rating system for articles based on their level of difficulty and the extent of technical knowledge needed:

🟢 Beginner; no programming required
🟦 Easy; basic programming knowledge may be necessary, but no specialized expertise
◆ Intermediate; programming skills and some domain knowledge required (e.g., calculating logarithmic probabilities)
◆◆ Advanced; programming expertise and in-depth domain understanding needed (e.g., reinforcement learning techniques)

Please note that even for ◆ and ◆◆ articles, you can generally grasp the content without prior domain expertise, though it may be helpful for implementation.

Feedback
The single most important part of this course is your feedback!
If you have any questions, comments, or suggestions, you can:

Make an issue on GitHub
Email us at team@learnprompting.org
Ask in the Discord community
Ping us on Twitter
Your feedback helps us improve the course for everyone.

Conclusion
It is time to get started with your Generative AI learning Journey. Click the Introduction to AI button at the bottom left of this page to continue (or click the following link for the Basics Introduction).

Welcome to the Basics Section of the Prompt Engineering Guide.
This section is designed to introduce you to prompt engineering, a critical skill when working with generative AI models such as ChatGPT, Gemini, Cohere Chat, Claude, and more. Through this guide, you'll gain foundational knowledge about these AI tools and the basic techniques needed to harness their potential effectively.

The guide provides a non-technical introduction to Generative AI (GenAI), covering prompting strategies, popular GenAI tools, and practical ways to integrate GenAI into your everyday life. Whether you're a complete beginner or someone looking to enhance your skills, this guide will help you get started quickly. As you progress, the subsequent modules will deepen your knowledge and refine your prompting techniques for various AI use cases.

Below is a summary of the key topics covered in this guide. Click on any section to jump directly to that topic.

Key Sections of the Guide
Introduction to AI: A brief introduction to Artificial Intelligence (AI) and Generative AI, explaining how these systems work and their impact.

Getting Started with ChatGPT: Learn how to set up and use ChatGPT for various tasks, from simple queries to advanced functions like summarizing text and solving problems.

Learn Prompting Embeds: An interactive tool that allows you to experiment with different prompts directly on the Learn Prompting website to improve your prompting skills.

Prompt Engineering: Discover the fundamentals of prompt engineering, a critical skill for working with generative AI models like ChatGPT and DALL-E, and learn how to craft effective prompts.

How to Write an Effective Prompt?: A practical guide with tips and techniques to improve the quality of your AI's responses, whether you're writing, coding, or generating images.

Giving Instructions: This guide dives into instruction prompting, explaining how to provide clear, concise instructions to get the desired output from generative AI models.

Assigning Roles: Learn how to use role prompting to assign specific roles to AI models, helping you control the tone, style, and accuracy of the generated content for various tasks.

Shot-Based Prompting: Explore zero-shot, one-shot, and few-shot prompting techniques. Learn how providing examples to the AI can improve accuracy, along with real-world applications of each method.

Parts of a Prompt: A detailed breakdown of the key components of a prompt, including examples and tips to help you structure your prompts for optimal results.

Combining Techniques: Discover how to combine different prompting techniques, such as context, instructions, and examples, to enhance the AI model's ability to handle more complex tasks.

Chatbots vs. LLMs: Learn the key differences between chatbots and large language models (LLMs), exploring why chatbots are often the preferred interface for interacting with LLMs.

Priming Prompt: Priming chatbots involves structuring prompts to guide the chatbot’s responses. Learn how to use this technique to influence the behavior and output of chatbots for specific goals.

Limitations of LLMs: While LLMs are powerful, they have limitations. This document covers common challenges like hallucinations, biases, and prompt hacking, and offers strategies to work around them.

Beyond LLMs: Generative AI Applications Beyond Text Data: Explore the full range of generative AI capabilities beyond text generation, including image, audio, and video creation, and how these applications are shaping industries.

Problem Solving with GenAI: Learn the Learn Prompting Method, a structured approach to problem-solving using generative AI, from identifying the right tools to refining prompts through testing.

Moving Forward: Beyond the Basics: A sneak peek into advanced prompt engineering techniques and applications that will help you deepen your skills and expertise as you move beyond the basics.

Learn Prompting Embeds

Introduction
This guide will walk you through setting up and using the Learn Prompting Embed, an interactive tool that allows you to test prompts directly on the Learn Prompting website.
We'll cover:
What is the Learn Prompting Embed?
Get Set Up
Get an OpenAI API Key
Using the Embed
Conclusion

FAQ
1. What is the Learn Prompting Embed?
The ChatGPT website is great, but wouldn’t it be even better to write and test prompts right here on this website? With our Learn Prompting Embeds, you can! Keep reading to see how you can set it up. We’ll be using these interactive embeds throughout most of our articles.
2. Get Set Up
Here’s what the embed looks like:

embed
You should see a similar embed just below this paragraph. If it’s not visible, try enabling JavaScript or switching to a different browser. Still no luck? Reach out to us on Discord and we’ll help you troubleshoot.
If you see the embed, click Generate. If it’s your first time, you’ll be prompted to sign in with your Google account.
Note
We currently only support Google Authentication, but we're working on integrating all providers!
For Learn Prompting Plus subscribers: If you have a Learn Prompting Plus subscription, use the same email to sign in.
For non-subscribers: If you're not a Learn Prompting Plus subscriber, the setup process involves adding your OpenAI API key. Here’s how to find it.
3. Get an OpenAI API Key
Go to platform.openai.com/account/api-keys.
Sign up or sign into your OpenAI account.
Click Create new secret key to generate a key. You’ll see a string of text like this:
API key
Copy and paste this key into the embed on this site and click Submit.
Note
If you need to update your API key, clear your browser cookies and re-enter the new key.
Now, you’re all set! You can use the embeds throughout the site. Keep in mind that OpenAI charges for each prompt. If you’ve just created a new account, you’ll get three months of free credits. After that, don’t worry – it’s very affordable. You can generate about 7,000 words for just $0.02 1
See OpenAI pricing information
Caution
Never tell anyone your API key, since they could charge your account with prompts.
4. Using the Embed
Let’s see how to use the embed. Edit the "Type your prompt here" field in the embed. It’s just like using ChatGPT, except you can’t have long

conversations. In this basic guide, we use these embeds to demonstrate prompt engineering techniques.
You can see four pieces of information under the Generate button. The left one, 'gpt-3.5-turbo' is the model (gpt-3.5-turbo is the technical name for ChatGPT). The three numbers are LLM settings, which we will learn about in a few articles. If you would like to make your own embed, click the edit this embed button.

5. Conclusion

Learn Prompting Embeds make it easier to experiment with prompts without leaving the course site. But if you prefer ChatGPT, you can keep using that. Just remember to save your API key since OpenAI shows it only once.

6. FAQ

What are Learn Prompting Embeds?
Learn Prompting Embeds are a tool to test prompt engineering techniques directly on the Learn Prompting website.
Do I have to use Learn Prompting Embeds?
No, you can continue using the ChatGPT interface if that’s your preference.
What do I need to set up to use Learn Prompting Embeds?
To use Learn Prompting Embeds, you just need an OpenAI API key, which you can create on the OpenAI website.

Courses

Course	Instructor	Level	Price	Type
ChatGPT for Everyone	Sander Schulhoff Shyamal Anadkat	Beginner	Free	Prompting /AI
Introduction to Prompt Engineering	Sander Schulhoff Fady Yanni	Beginner	Plus	Prompting
Introduction to Prompt Hacking	Sander Schulhoff Fady Yanni	Beginner	Plus	Prompting
Introduction to Generative AI	Sander Schulhoff Fady Yanni	Beginner	Plus	Prompting
Image creation with DALL E 3	Camryn Streib Sander Schulhoff	Beginner	Plus	Prompting
AI Safety	Sander Schulhoff	Beginner	Plus	AI
Boost Your Day-to-day Efficiency with Generative AI	Sander Schulhoff Fady Yanni	Beginner	Plus	Prompting
Introduction to LLMs	Sander Schulhoff Fady Yanni	Beginner	Plus	Prompting
Introduction to Generative AI Agents	Sander Schulhoff	Intermediate	Plus	Prompting /AI
Prompt Engineering for Marketing	Brian Paul	Intermediate	Plus	Prompting
Runway ML for Everyone	AJ	Intermediate	Plus	Prompting
Introduction to RAG	Sander Schulhoff	Intermediate	Plus	Prompting /AI
Advanced Prompt Engineering	Sander Schulhoff	Advanced	Plus	Prompting
Advanced Prompt Hacking	Sander Schulhoff	Advanced	Plus	Prompting
Advanced OpenAI Playground	Sander Schulhoff	Advanced	Plus	Prompting

Beginner

ChatGPT for Everyone

Course Plan
1 Hour
Course Overview
Learn about ChatGPT, one of the most advanced AI systems available today, and dive into the world of Generative AI.
What will you learn?
ChatGPT
How to use ChatGPT, create an account, and write your first basic prompt.
Use Cases
How to use ChatGPT as your personal assistant to maximize your productivity.
Skills you will gain
ChatGPT
Prompt Engineering
DALL·E 3
GPT 3.5
GPT 4

Course Syllabus
1
Introduction to ChatGPT
What is ChatGPT
How does ChatGPT work?
2
ChatGPT's Use Cases
Using ChatGPT as your Personal Assistant
Using ChatGPT for Everyday Writing
Boost your Productivity with ChatGPT

Content Creation with ChatGPT
3

Tactics to Write Effective Prompts
Detailed breakdown of a Prompt
Giving Instructions and Assigning Roles
Tips for Optimal Prompts
4
Ethics, AI Safety, and Limitations
Recognizing the Limitations of ChatGPT
Ethical and Responsible Use
Case Studies

Introduction to Prompt Engineering

Course Plan
3 Days
Course Overview
Learn about the basics of Prompt Engineering, and how to effectively communicate with AI.
What will you learn?
Prompting
The Fundamentals of Prompting & Prompt Engineering
Skills you will gain
ChatGPT
Prompt Engineering
DALL·E 3
GPT 3.5
GPT 4

Course Syllabus
1
What is Prompt Engineering?
What is a Prompt?
What is Prompt Engineering?
Why should we use Prompt Engineering?
2
Understanding LLMs

What are LLMs?
How LLMs Work
Capabilities
Limitations
3
Basic Prompting Techniques
Elements of a good prompt
Chain of Thought
Zero-Shot
Few-Shot
4
Combining techniques and Case Studies
Examples of Prompts
Try-it-yourself
Case Studies

Introduction to Prompt Hacking
Course Plan
3 Days
Course Overview
Learn about the basics of Prompt Hacking, one of the biggest vulnerabilities in Large Language Models (LLMs), and Prompt Defense techniques.
What will you learn?
Prompt Hacking
How to Prompt Inject and Jailbreak Large Language Models.
Skills you will gain
Prompt Injection
Jailbreaking
GPT 3.5
ChatGPT
Course Syllabus
1
Introduction

What is Prompt Hacking?
What is the difference between Prompt Hacking and Jailbreaking?
2
Introduction to Prompt Injection Attacks
What is Prompt Injection?
Potential Threats
How we get Prompt Injected
3
Preventing Injections in LLMs
Not Trusting User Input
Post-prompting and the Sandwich Defense
Few-Shot Prompting Defense
Non-Prompt-based Techniques
4
Other Prompt Hacking Concepts
Prompt Leaking
Jailbreaking

Introduction to Generative AI
Course Plan
3 Days
Course Overview
Learn about the conceptual fundamentals of Generative AI for a business setting.
What will you learn?
GenAIs
How different GenAIs work.
Conceptual Fundamentals
Get deep enough to apply to your prompts.
Skills you will gain
ChatGPT
Prompt Engineering
DALL·E 3
GPT 3.5

GPT 4

Course Syllabus
1
How they work
Learn how GenAIs work.
Learn about different ones.
2
Understand GenAI capabilities
3D Generation
Audio Generation
Video Generation

Image Creation with DALL·E 3

Course Plan
3 Days
Course Overview
Unlock the full potential of DALL·E 3 with advanced techniques for creating high-quality images, including logos, pixel art sprite sheets, and more.
What will you learn?
Image Prompting
Create prompts for image generation.
Applications
Create unique logos and pixel art sprite sheets.
Gen-ID
Obtain consistent image generations.
Skills you will gain
DALL·E 3
Prompt Engineering
ChatGPT
GPT 3.5
GPT 4

Course Syllabus

1
Advanced Prompting Techniques
Prompts for specific image types.
Enhanced detail and specificity in prompts.
2
Creating Logos
Logos for brand identity.
Logo styles and themes.
3
Pixel Art Sprite Sheets
Pixel art for game characters.
Sprite sheets with multiple animations.
4
Gen-ID for Consistency
Gen-ID for series consistency.
Gen-ID in different image scenarios.

AI Safety

Course Plan
3 Days
Course Overview
Learn about the ethical considerations and safety protocols in modern AI applications.
What will you learn?
Safety
Use Generative AI safely in your life and job.
Skills you will gain
ChatGPT
Prompt Engineering
DALL·E 3
GPT 3.5
GPT 4

Course Syllabus

1
Introduction to AI Safety
AI Safety Concerns
Historical Context and Importance
2
Technical Aspects of AI Safety
Safe AI Design Principles
Robustness and Reliability
Avoiding and Mitigating Risks
3
Ethical and Societal Implications
Ethical Frameworks in AI
Societal Impact and Responsibilities
Case Studies in Ethical AI Use
4
Keeping your Enterprise Safe
AI Safety Strategies
AI Governance and Policy
Collaborative Safety Strategies

Boost Your Day-to-Day Efficiency With Generative AI

Course Plan
3 Days

Course Overview
Learn how to get more done and improve your daily efficiency using practical AI skills.
What will you learn?
Efficiency
How to use Generative AI to automate daily tasks.

Skills you will gain
ChatGPT

Prompt Engineering
DALL·E 3
GPT 3.5
GPT 4

Course Syllabus
1
Efficiency Assessment
Calculating total time spent on tasks currently
Assignment
2
Streamline Text-Based Tasks
Text-based tasks
Powerpoints
Presentations
Research
Summarization
3
Streamline Audio & Visual Tasks
Image Generation
Audio Generation
Slidedeck Generation
4
Streamlining Coding Tasks
Debugging Code
Documentation Writing

Introduction to LLMs
Course Plan
3 Days
Course Overview
Learn about the conceptual fundamentals of Large Language Models for a business setting.
What will you learn?

LLMs
How different LLMs work.
Conceptual Fundamentals
Get deep enough to apply to your prompts.
Skills you will gain
ChatGPT
Prompt Engineering
GPT 3.5
GPT 4

Course Syllabus
1
How they work
Learn how LLMs work.
Learn about different ones.
2
Understand LLM capabilities
Summarization
Classification
Content Generation
3
Understand how they learn
Fine-Tuning
Pre-Training
RLHF

Intermediate

Introduction to Generative AI Agents

Course Plan
1 Hour
Course Overview
Learn about the origin of Generative AI Agents, as well as the 4 different types and how to implement your own.
What will you learn?
Agents
Understand and Build Generative AI Agents
Use Cases
See Generative AI Agents Applied to Real World Tasks
Skills you will gain
Agents
ChatGPT
Prompt Engineering
Prompt Flow
GPT 4

Course Syllabus
1
Introduction to Generative AI Agents
What are Generative AI Agents
How do Generative AI Agents work?
2
What are the four types of Generative AI Agents?
Observation-based Agents
Tool use Agents
Code-based Agents
Retrieval-Augmented Generation Agents

Prompt Engineering for Marketing

Course Plan
3 Days
Course Overview
Unlock the Power of AI to Revolutionize Your Marketing Strategies and Campaigns
What will you learn?
Marketing
How to do marketing more efficiently with Generative AI.
Skills you will gain
ChatGPT
Prompt Engineering
DALL·E 3
GPT 3.5
GPT 4

Course Syllabus
1
Fundamentals of Prompting for Marketing
Applying Prompt Engineering to Marketing
Using AI to Craft Great Web Copy
Optimizing SEO
Generating Brand Aligned Social Media Materials
Introduction to Image Prompting for Marketing
2
Text Content Creation
Web Copy Crafting
Email and Social Media
Interactive Content Techniques
3
Image Prompting
Image Prompting Basics
Brand-aligned Visuals

Runway ML for Everyone

Course Plan
3 Days
Course Overview
Dive into the world of multimedia creation with our Runway ML course, designed to equip you with the skills to master image, video, and audio tools. This guide will enable beginners to manipulate and generate content using the forefront of generation AI media.
What will you learn?
Video Prompting
The Fundamentals of Image & Video Generation.
Skills you will gain
Prompt Engineering
Runway ML
Video Editor

Course Syllabus
1
Images
Text To Image
Image to Image
Erase and Replace
Image Variation
Backdrop Remix
Expand Image
Infinite Image
Upscale Image
Add color
2
Video
Text To Video
Video to Video
Inpainting
Color Grade

Super Slow Motion
Green Screen
Frame Interpolation
Blur Faces
Scene Detection
3
Audio
Clean Audio
Remove Silence
Transcript
Subtitles
Generative Audio

Introduction to RAG

Course Plan
1 Hour
Course Overview
Learn about the basics of RAG, including agentic RAG and vector databases.
What will you learn?
RAG
Understand and Build RAG systems
Use Cases
See RAG systems Applied to Real World Tasks
Skills you will gain
Agents
RAG
Prompt Engineering
Prompt Flow
GPT 4

Course Syllabus

1
Introduction to Retrieval-Augmented Generation
What is RAG
How does RAG work?
2
What is the difference between RAG and Agentic Rag?
Agentic RAG allows the agent to decide what information to pull in.
3
What are vector databases
GenAI integrated databases that allow for better retrieval.

Advanced

Advanced Prompt Engineering
Course Plan
3 Days
Course Overview
Learn how to craft Complex and Efficient Prompts for Sophisticated AI Applications.
What will you learn?
Advanced Prompting
The most advanced prompting techniques, from research.
Skills you will gain
ChatGPT
Prompt Engineering
DALL·E 3
GPT 3.5
GPT 4

Course Syllabus
1
In-Context Learning
Basics of Few-Shot Prompting
Factors Influencing Performance in Few-Shot In-Context Learning
Self-Generated In-context Learning Prompting (SG-ICL)
2
Thought Generation Prompting
Basics of Thought Generation Prompting
Chain-of-Thought Prompting (CoT)
Thread-of-Thought Prompting (ThoT)
Contrastive Chain-of-Thought Prompting (CCoT)
Self-Ask Prompting (SA)
Tabular Chain of Thought Prompting (Tab-CoT)
3
Problem Decomposition Prompting

Basics of Problem Decomposition Prompting
Least-to-Most Prompting (LtM)
Plan-and-Solve Prompting (PaS)
Program-of-Thoughts Prompting (PoTh)
4
Self-Criticism Prompting
Basics of Self-Criticism Prompting
Self-Evaluation Prompting (SE)
Self-Refine Prompting (SR)
Chain-of-Verification Prompting (COVE)
System 2 Attention Prompting (S2A)
Rephrase and Respond Prompting (RaR)
Re-reading Prompting (RE2)

Advanced Prompt Hacking

Course Plan
3 Days
Course Overview
Learn about Advanced Prompt Hacking techniques and the vulnerabilities of Large Language Models (LLMs) to exploits like Jailbreak attacks, Prompt Injection attacks, and Cognitive Hacking.
What will you learn?
Prompt Hacking
Bleeding edge prompt hacking techniques, directly from our published research.
Skills you will gain
ChatGPT
Prompt Engineering
DALL·E 3
GPT 3.5
GPT 4

Course Syllabus

1
Fundamentals of Prompt Hacking
Simple Instruction Attacks
Compound Instruction Attacks
Refusal Suppression Attacks
Distractor Instructions Attacks
Project & Quiz
2
Contextual Hacking
Context Switching Attacks
Separator Attacks
Context Termination Attacks
Project & Quiz
3
Cognitive Hacking & Few-Shot
Few-Shot Attacks
Cognitive Hacking Attacks
Virtualization Attacks
Project & Quiz

4
Miscellaneous Attacks
Context Overflow Attacks
Recursive Attacks
Project & Quiz

Advanced OpenAI Playground

Course Plan
3 Days
Course Overview
Learn about the advanced features of the OpenAI Playground and how to use it to build custom GPTs & Chatbots
What will you learn?

OpenAI Playground
How to OpenAI's prompting IDE.
Skills you will gain
ChatGPT
Prompt Engineering
DALL·E 3
GPT 3.5
GPT 4

Course Syllabus
1
OpenAI Playground Features
Overview
Presets
Refusal Suppression Attacks
Revisiting History
2
How to use Chat Mode
Temperature and Top P
Maximum Length and Stop Sequences

Frequency and Presence Penalty
Project & Quiz
3
How to use Assistant Mode
Creating your First Assistant
How to Send a Message
Project & Quiz
4
Assistant Settings & Specifications
Name, Instructions and Model
Tools: Functions, Code Interpreter, Retrieval
Project & Quiz

Instructors

Sander Schulhoff

Founder & CEO, Learn Prompting
His guide (Learnprompting. org) is listed as a recommended resource by OpenAI and Google and has been featured in Forbes as the best course to learn Prompt Engineering. As a researcher, he has authored multiple award-winning papers alongside experts from OpenAI, Microsoft, ScaleAI, the Federal Reserve, HuggingFace, and others.

Fady Yanni

Co-founder & COO at Learn Prompting
Fady Yanni is the Founder & COO of Learn Prompting, the leading Prompt Engineering resource which has taught over 3 million people how to effectively communicate with AI. Previously, he was the Head of Fundraising at the Farama Foundation, the open-source maintainers of every major Reinforcement Learning library, including OpenAI's flagship project, Gym.

Camryn Streib

Game Designer
Alongside her work with Learn Prompting, Camryn is a Game Designer, XR Researcher, and Artist with a Background In Immersive Media Design and Psychology. Through her experience in emerging creative fields, she has published work in Rockstar Games " GTA V", Indie Game Studio "The Verse", and more.

Shyamal Anadkat

Applied AI at OpenAI
Shyamal Anadkat is a Member of the Applied AI Team at OpenAI, the creators of the fastest growing product ever and the most advanced AI model to date, ChatGPT. His work on the Applied AI team focuses on safely integrating OpenAI's technologies into various applications globally.

Shyamal holds a Master's degree in Artificial Intelligence from Duke University.

Brian Paul

Marketing Expert

Brian Paul is an experienced digital marketer and AI artist, using his skillset to grow brands online. Brian has worked with e-commerce brands, non profits, and startups to help them achieve their marketing goals. He is passionate about the intersection of Generative AI and creativity.

Camryn Streib

Game Designer

Alongside her work with Learn Prompting, Camryn is a Game Designer, XR Researcher, and Artist with a Background In Immersive Media Design and Psychology. Through her experience in emerging creative fields, she has published work in Rockstar Games “ GTA V”, Indie Game Studio “The Verse”, and more.

AJ

Creative Producer

Alongside his work as a Creative Producer at Learn Prompting, AJ is an XR researcher and developer. He has previously co-founded a startup, Spray, that allowed people to express themselves through augmented reality art.

Pricing

1. Limited (Free Tier)
   - Target Audience: Individuals who are just starting to learn about AI and prefer free resources.
   - Pricing: Free of charge.
   - Features:
     - Access to free courses.
     - No certificates of completion.

- Purpose: Acts as an entry-level tier to attract beginners and encourage eventual upgrades.

2. Plus (Individual Tier)
   - Target Audience: Individual learners who are serious about developing advanced AI skills.
   - Pricing: $21 per month, billed annually.
   - Features:
      - Access to the entire course library, including exclusive Plus Courses.
      - Certificates of completion for all courses.
      - Access to an in-course AI playground with an AI tutor.
      - Engagement with hand-graded AI projects.
      - Access to the largest AI Learning Discord community.
      - Priority customer support.
   - Purpose: Designed for enthusiasts or professionals ready to invest in comprehensive learning resources.

3. Teams (Group Tier)
   - Target Audience: Teams and departments (2–99 users) aiming to adopt AI-first approaches and basic administrative support.
   - Pricing: $25 per seat per month, billed annually.
   - Features:
      - Everything in the Plus plan.
      - Scalable support for multiple users, suitable for collaborative learning.
   - Purpose: Provides team-based learning with administrative tools for organizations wanting a collaborative educational experience.

4. Enterprise
   - Target Audience: Organizations with 100+ users looking for customized and scalable solutions for company-wide AI transformation.
   - Pricing: Contact sales for pricing (customized based on needs).
   - Features:
      - Everything in the Plus and Teams plans.

        - Personalized learning paths with tailored projects.
        - Access to an admin dashboard.
        - Dedicated customer success manager.
        - Advanced analytics and reporting capabilities.
        - Single Sign-On (SSO) through Okta, Auth0, Azure, and more.
        - LMS/LXP integrations for seamless learning management.
   - Purpose: Tailored for large-scale organizations requiring enterprise-grade solutions, customization, and additional support.

Key Pricing Strategies Summary
- Free Tier for Acquisition: The free tier provides basic access, attracting users who may later convert to paid plans.
- Value-Added Pricing: Each subsequent tier offers more value and features, justifying the higher price points.
- Annual Billing Incentives: Discounts for annual commitments (e.g., monthly pricing is framed but tied to annual billing).
- Custom Pricing for Enterprises: Tailored pricing for large clients enables flexibility and maximization of revenue from high-value customers.
- Most Popular Tag: The Plus tier is marketed as the most popular, creating social proof and nudging users toward the mid-tier option.

Searched 5 sites
To analyze and document the landing page for the "Learn Prompting" courses (https://learnprompting.org/courses
) and structure user support documentation for a chatbot, here's an overview and proposed documentation based on the content and structure of the website:

Website Overview
The "Learn Prompting" platform provides comprehensive courses on prompt engineering, a key skill for interacting with AI models like ChatGPT. The site emphasizes:
Core topics: Basics of prompt engineering, intermediate techniques, and advanced methods like prompt hacking, image prompting, and reliability strategies.
Interactive features: Playgrounds for practicing prompt engineering, exploring use cases for tools like GPT-3, and testing prompts.
Learning formats: On-demand courses, live workshops, and free materials for all skill levels.
Community and resources: Discord integration for peer collaboration, links to tools like PromptTools, and research articles

Key Features and Navigation
Courses and Modules
Courses are divided into beginner, intermediate, and advanced sections.
Topics include:
Zero-shot, one-shot, and few-shot prompting.
Image and text-based prompt engineering.
Tooling and model-specific techniques
Learn Prompting

Learn Prompting
.
Interactive Tools
"HackAPrompt Playground" for experimenting with prompt effectiveness.

Links to tools like PromptTools for side-by-side prompt testing and SDK integration
Learn Prompting
.
Documentation
Guides and tutorials for setting up prompting workflows.
FAQs and examples for real-world use cases like information extraction, content creation, and sentiment analysis
Learn Prompting
.
Accessibility
Free resources include introductory materials and basic tools.
Pricing plans for advanced features are available under a separate tab.

Prompt engineering

Prompt engineering is the process of structuring an instruction that can be interpreted and understood by a generative artificial intelligence (AI) model.[1][2] A *prompt* is natural language text describing the task that an AI should perform.[3] A prompt for a text-to-text language model can be a query such as "what is Fermat's little theorem?",[4] a command such as "write a poem in the style of Edgar Allan Poe about leaves falling",[5] or a longer statement including context, instructions,[6] and conversation history.

Prompt engineering may involve phrasing a query, specifying a style,[5] choice of words and grammar,[7] providing relevant context[8] or assigning a role to the AI such as "act as a native French speaker".[9]

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"[10] or "Lo-fi slow BPM electro chill with organic samples".[11] Prompting a text-to-image model may involve adding, removing, emphasizing, and re-ordering words to achieve a desired subject, style,[1] layout, lighting,[12] and aesthetic.

History

In 2018, researchers first proposed that all previously separate tasks in 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?"[13]

In 2021, researchers fine-tuned one generatively pretrained model (T0) on performing 12 NLP tasks (using 62 datasets, as each task can have multiple datasets). The model showed good performance on new tasks, surpassing models trained directly on just performing one task (without pretraining). To solve a task, T0 is given the task in a structured prompt, for example `If {{premise}} is true, is it also true that {{hypothesis}}? ||| {{entailed}}.` is the prompt used for making T0 solve entailment.[14]

A repository for prompts reported that over 2,000 public prompts for around 170 datasets were available in February 2022.[15]

In 2022 the *chain-of-thought* prompting technique was proposed by Google researchers.[16][17]

In 2023 several text-to-text and text-to-image prompt databases were publicly available.[18][19]

Text-to-text

Chain-of-thought

According to Google, *Chain-of-thought* (CoT) prompting is claimed to be a technique that allows large language models (LLMs) to solve a problem as a series of intermediate steps[20] before giving a final answer. In 2022, Google also claimed 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.[21][16][22] Chain-of-thought techniques hypothetically allow large language models to overcome difficulties with some reasoning tasks that require logical thinking and multiple steps to solve, such as arithmetic or commonsense reasoning questions, according to announcements from Google and Amazon.[23][24][25]

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."[16]

As originally proposed by Google,[16] each CoT prompt included a few Q&A examples. This made it a *few-shot* prompting technique. However, according to a researchers at Google and the University of Tokyo, simply appending the words "Let's think step-by-step",[26] has also proven effective, which makes CoT a *zero-shot* prompting technique. OpenAI claims that this prompt allows for better scaling as a user no longer needs to formulate many specific CoT Q&A examples.[27]

When applied to PaLM, a 540B parameter language model, Google claims that 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.[16] According to Google, it is possible to fine-tune models on CoT reasoning datasets to enhance this capability further and stimulate better interpretability.[28][29]

Example:[26]

```
Q: {question}
A: Let's think step by step.
```

Other techniques

Chain-of-thought prompting is just one of many prompt-engineering techniques. Various other techniques have been proposed. At least 29 distinct techniques have been published.[30]

Chain-of-Symbol (CoS) Prompting

A research collaboration between Westlake University, the Chinese University of Hong Kong, and the University of Edinburgy has claimed that chain-of-Symbol prompting in conjunction with CoT prompting assists LLMs with its difficulty of spatial reasoning in text. In other words, using arbitrary symbols such as ' / ' assist the LLM to interpret spacing in text. This is claimed to assist in reasoning and increases the performance of the LLM.[31]

Example:[31]

```
Input:

There are a set of bricks. The yellow brick C is on top of the brick E. The yellow brick D is on top of the brick A.
The yellow brick E is on top of the brick D. The white brick A is on top of the brick B. For the brick B, the color is
white. Now we have to get a specific brick. The bricks must now be grabbed from top to bottom, and if the lower brick
is to be grabbed, the upper brick must be removed first. How to get brick D?

B/A/D/E/C
C/E
E/D
D

Output:

So we get the result as C, E, D.
```

Few-shot learning

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*),[32] an approach called **few-shot learning**.[33]

Generated knowledge prompting

Generated knowledge prompting[34] first prompts the model to generate relevant facts for completing the prompt, then proceed to complete the prompt. The completion quality is usually higher, as the model can be conditioned on relevant facts.

Example:[34]

```
    Generate some knowledge about the concepts in the input.
    Input: {question}
    Knowledge:
```

Least-to-most prompting

Least-to-most prompting[35] prompts a model to first list the sub-problems to a problem, then solve them in sequence, such that later sub-problems can be solved with the help of answers to previous sub-problems.

Example:[35]

```
    Input:
    Q: {question}
```

```
A: Let's break down this problem:
    1.
```

Self-consistency decoding

Self-consistency decoding[36] performs several chain-of-thought rollouts, then selects the most commonly reached conclusion out of all the rollouts. If the rollouts disagree by a lot, a human can be queried for the correct chain of thought.[37]

Complexity-based prompting

Complexity-based prompting[38] performs several CoT rollouts, then select the rollouts with the longest chains of thought, then select the most commonly reached conclusion out of those.

Self-refine

Self-refine[39] prompts the LLM to solve the problem, then prompts the LLM to critique its solution, then prompts the LLM to solve the problem again in view of the problem, solution, and critique. This process is repeated until stopped, either by running out of tokens, time, or by the LLM outputting a "stop" token.

Example critique:[39]

```
I have some code. Give one suggestion to improve readability. Don't fix the code, just give a suggestion.
Code: {code}
Suggestion:
```

Example refinement:

```
Code: {code}
Let's use this suggestion to improve the code.
Suggestion: {suggestion}
New Code:
```

Tree-of-thought

Tree-of-thought prompting[40] generalizes chain-of-thought by prompting the model to generate one or more "possible next steps", and then running the model on each of the possible next steps by breadth-first, beam, or some other method of tree search.[41]

Maieutic prompting

Maieutic prompting is similar to tree-of-thought. The model is prompted to answer a question with an explanation. The model is then prompted to explain parts of the explanation, and so on. Inconsistent explanation trees are pruned or discarded. This improves performance on complex commonsense reasoning.[42]

Example:[42]

```
Q: {question}
A: True, because
```

```
Q: {question}
A: False, because
```

Directional-stimulus prompting

Directional-stimulus prompting[43] includes a hint or cue, such as desired keywords, to guide a language model toward the desired output.

Example:[43]

```
Article: {article}
Keywords:
```

```
Article: {article}
Q: Write a short summary of the article in 2-4 sentences that accurately incorporates the provided keywords.
Keywords: {keywords}
A:
```

Prompting to disclose uncertainty

By default, the output of language models may not contain estimates of uncertainty. The model may output text that appears confident, though the underlying token predictions have low likelihood scores. Large language models like GPT-4 can have accurately calibrated likelihood scores in their token predictions,[44] and so the model output uncertainty can be directly estimated by reading out the token prediction likelihood scores.

But if one cannot access such scores (such as when one is accessing the model through a restrictive API), uncertainty can still be estimated and incorporated into the model output. One simple method is to prompt the model to use words to estimate uncertainty.[45] Another is to prompt the model to refuse to answer in a standardized way if the input does not satisfy conditions.

Prompting to estimate model sensitivity

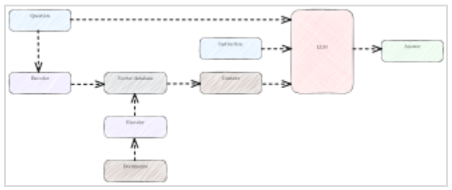
Research consistently demonstrates that LLMs are highly sensitive to subtle variations in prompt formatting, structure, and linguistic properties. Some studies have shown up to 76 accuracy points across formatting changes in few-shot settings.[46] Linguistic features significantly influence prompt effectiveness—such as morphology, syntax, and lexico-semantic changes—which meaningfully enhance task performance across a variety of tasks.[47][48] Clausal syntax, for example, improves consistency and reduces uncertainty in knowledge retrieval.[49] 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.[50] Similarly, PromptEval estimates performance distributions across diverse prompts, enabling robust metrics such as performance quantiles and accurate evaluations under constrained budgets.[51]

Automatic prompt generation

Retrieval-augmented generation

Retrieval-augmented generation (RAG) is a two-phase process involving document retrieval and answer formulation by a Large Language Model (LLM). The initial phase utilizes dense embeddings to retrieve documents. This retrieval can be based on a variety of database formats depending on the use case, such as a vector database, summary index, tree index, or keyword table index.[52]

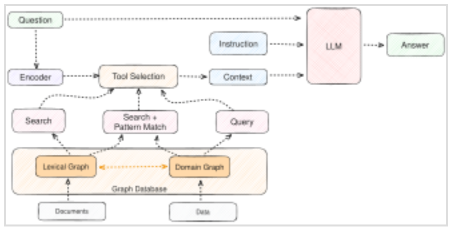


Two-phase process of document retrieval using dense embeddings and Large Language Model (LLM) for answer formulation

In response to a query, a document retriever selects the most relevant documents. This relevance is typically determined by first encoding both the query and the documents into vectors, then identifying documents whose vectors are closest in Euclidean distance to the query vector. Following document retrieval, the LLM generates an output that incorporates information from both the query and the retrieved documents.[53] This method is particularly beneficial for handling proprietary or dynamic information that was not included in the initial training or fine-tuning phases of the model. RAG is also notable for its use of "few-shot" learning, where the model uses a small number of examples, often automatically retrieved from a database, to inform its outputs.

Graph retrieval-augmented generation

GraphRAG[54] (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.



GraphRAG with a knowledge graph combining access patterns for unstructured, structured and mixed data.

It was shown to be effective on datasets like the Violent Incident Information from News Articles (VIINA).[55] By combining LLM-generated knowledge graphs with graph machine learning, GraphRAG substantially improves the comprehensiveness and diversity of generated answers for global sensemaking questions.

Earlier work showed the effectiveness of using a knowledge graph for question answering using text-to-query generation.[56] 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.[57][58][59][60]

The *automatic prompt engineer* algorithm uses one LLM to beam search over prompts for another LLM:[61]

- 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",[62] a library of questions are converted to vectors by a model such as BERT. The question vectors are clustered. Questions nearest to the centroids of each cluster are selected. An LLM does zero-shot CoT on each question. The resulting CoT examples are added to the dataset. When prompted with a new question, CoT examples to the nearest questions can be retrieved and added to the prompt.

In-context learning

Prompt engineering can possibly be further enabled by **in-context learning**, defined as a model's ability to temporarily learn from prompts. The ability for in-context learning is an emergent ability[63] of large language models. In-context learning itself is an emergent property of model scale, meaning breaks[64] in downstream scaling laws occur such that its efficacy increases at a different rate in larger models than in smaller models.[65][16]

In contrast to training and fine-tuning for each specific task, which are not temporary, what has been learnt during in-context learning is of a temporary nature. It does not carry the temporary contexts or biases, except the ones already present in the (pre)training dataset, from one conversation to the other.[66] This result of "mesa-optimization"[67][68] within transformer layers, is a form of meta-learning or "learning to learn".[69]

Text-to-image

In 2022, text-to-image models like DALL-E 2, Stable Diffusion, and Midjourney were released to the public.[70] These models take text prompts as input and use them to generate AI art images. Text-to-image models typically do not understand grammar and sentence structure in the same way as large language models,[71] and require a different set of prompting techniques.

Prompt formats

A text-to-image prompt commonly includes a description of the subject of the art (such as *bright orange poppies*), 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.[72]

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".[71]

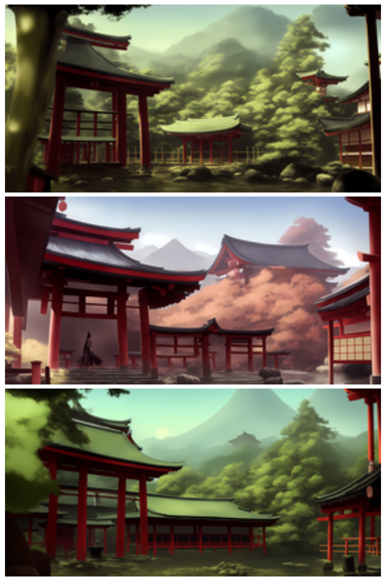
Word order affects the output of a text-to-image prompt. Words closer to the start of a prompt may be emphasized more heavily.[1]

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.[73]

Negative prompts

Text-to-image models do not natively understand negation. The prompt "a party with no cake" is likely to produce an image including a cake.[71] As an alternative, *negative prompts* allow a user to indicate, in a separate prompt, which terms should **not** appear in the resulting image.[74]


Demonstration of the effect of negative prompts on images generated with Stable Diffusion

- **Top**: no negative prompt
- **Centre**: "green trees"
- **Bottom**: "round stones, round rocks"

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"[75] 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.

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.[76]

Using gradient descent to search for prompts

In "prefix-tuning",[77] "prompt tuning" or "soft prompting",[78] floating-point-valued vectors are searched directly by gradient descent, to maximize the log-likelihood on outputs.

Formally, let $\mathbf{E} = \{\mathbf{e_1}, \dots, \mathbf{e_k}\}$ be a set of soft prompt tokens (tunable embeddings), while $\mathbf{X} = \{\mathbf{x_1}, \dots, \mathbf{x_m}\}$ and $\mathbf{Y} = \{\mathbf{y_1}, \dots, \mathbf{y_n}\}$ 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 $\text{concat}(\mathbf{E}; \mathbf{X}; \mathbf{Y})$, and fed to the large language models (LLM). The losses are computed over the $\mathbf{Y}$ 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.[79]

More formally, this is prompt tuning. Let an LLM be written as $LLM(X) = F(E(X))$, where X is a sequence of linguistic tokens, E is the token-to-vector function, and F is the rest of the model. In prefix-tuning, one provide a set of input-output pairs $\{(X^i, Y^i)\}_i$, and then use gradient descent to search for $\arg\max_{\tilde{Z}} \sum_i \log Pr[Y^i | \tilde{Z} * E(X^i)]$. In words, $\log Pr[Y^i | \tilde{Z} * E(X^i)]$ is the log-likelihood of outputting Y^i, if the model first encodes the input X^i into the vector $E(X^i)$, then prepend the vector with the "prefix vector" $\tilde{Z}$, then apply F.

For prefix tuning, it is similar, but the "prefix vector" $\tilde{Z}$ is preappended to the hidden states in every layer of the model.

An earlier result[80] 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 $\arg\max_{\tilde{X}} \sum_i \log Pr[Y^i | \tilde{X} * X^i]$ where $\tilde{X}$ is ranges over token sequences of a specified length.

Prompt injection

Prompt injection is a family of related computer security exploits carried out by getting a machine learning model (such as an LLM) which was trained to follow human-given instructions to follow instructions provided by a malicious user. This stands in contrast to the intended operation of instruction-following systems, wherein the ML model is intended only to follow trusted instructions (prompts) provided by the ML model's operator.[81][82][83]

See also

- Social engineering (security)

References

1. Diab, Mohamad; Herrera, Julian; Chernow, Bob (2022-10-28). "Stable Diffusion Prompt Book" (https://cdn.openart.ai/assets/Stable%20Diffusion%20Prompt%20Book%20From%20OpenArt%2011-

13.pdf) (PDF). Retrieved 2023-08-07. "Prompt engineering is the process of structuring words that can be interpreted and understood by a *text-to-image* model. Think of it as the language you need to speak in order to tell an AI model what to draw."

2. Ziegler, Albert; Berryman, John (17 July 2023). "A developer's guide to prompt engineering and LLMs" (https://github.blog/2023-07-17-prompt-engineering-guide-generative-ai-llms/). *The GitHub Blog*. "Prompt engineering is the art of communicating with a generative AI model."
3. Radford, Alec; Wu, Jeffrey; Child, Rewon; Luan, David; Amodei, Dario; Sutskever, Ilya (2019). "Language Models are Unsupervised Multitask Learners" (https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf) (PDF). OpenAI. "We demonstrate language models can perform down-stream tasks in a zero-shot setting – without any parameter or architecture modification"
4. "Introducing ChatGPT" (https://openai.com/blog/chatgpt). *OpenAI Blog*. 2022-11-30. Retrieved 2023-08-16. "what is the fermat's little theorem"
5. Robinson, Reid (August 3, 2023). "How to write an effective GPT-3 or GPT-4 prompt" (https://zapier.com/blog/gpt-prompt/). *Zapier*. Retrieved 2023-08-14. " "Basic prompt: 'Write a poem about leaves falling.' Better prompt: 'Write a poem in the style of Edgar Allan Poe about leaves falling.' "
6. Gouws-Stewart, Natasha (June 16, 2023). "The ultimate guide to prompt engineering your GPT-3.5-Turbo model" (https://masterofcode.com/blog/the-ultimate-guide-to-gpt-prompt-engineering). *masterofcode.com*.
7. Wahle, Jan Philip; Ruas, Terry; Xu, Yang; Gipp, Bela (2024). Al-Onaizan, Yaser; Bansal, Mohit; Chen, Yun-Nung (eds.). "Paraphrase Types Elicit Prompt Engineering Capabilities" (https://aclanthology.org/2024.emnlp-main.617/). *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Miami, Florida, USA: Association for Computational Linguistics: 11004–11033. arXiv:2406.19898 (https://arxiv.org/abs/2406.19898).
8. Greenberg, J., Laura (31 May 2023). "How to Prime and Prompt ChatGPT for More Reliable Contract Drafting Support" (https://contractnerds.com/how-to-prime-and-prompt-chatgpt-for-more-reliable-contract-drafting-support). *contractnerds.com*. Retrieved 24 July 2023.
9. "GPT Best Practices" (https://platform.openai.com/docs/guides/gpt-best-practices). OpenAI. Retrieved 2023-08-16.
10. Heaven, Will Douglas (April 6, 2022). "This horse-riding astronaut is a milestone on AI's long road towards understanding" (https://www.technologyreview.com/2022/04/06/1049061/dalle-openai-gpt3-ai-agi-multimodal-image-generation/). *MIT Technology Review*. Retrieved 2023-08-14.
11. Wiggers, Kyle (2023-06-12). "Meta open sources an AI-powered music generator" (https://techcrunch.com/2023/06/12/meta-open-sources-an-ai-powered-music-generator/). TechCrunch. Retrieved 2023-08-15. "Next, I gave a more complicated prompt to attempt to throw MusicGen for a loop: "Lo-fi slow BPM electro chill with organic samples." "
12. "How to Write AI Photoshoot Prompts: A Guide for Better Product Photos" (https://claid.ai/blog/article/prompt-guide/). *claid.ai*. June 12, 2023. Retrieved June 12, 2023.
13. McCann, Bryan; Shirish, Nitish; Xiong, Caiming; Socher, Richard (2018). "The Natural Language Decathlon: Multitask Learning as Question Answering". arXiv:1806.08730 (https://arxiv.org/abs/1806.08730) [cs.CL (https://arxiv.org/archive/cs.CL)].
14. Sanh, Victor; et al. (2021). "Multitask Prompted Training Enables Zero-Shot Task Generalization". arXiv:2110.08207 (https://arxiv.org/abs/2110.08207) [cs.LG (https://arxiv.org/archive/cs.LG)].
15. Bach, Stephen H.; Sanh, Victor; Yong, Zheng-Xin; Webson, Albert; Raffel, Colin; Nayak, Nihal V.; Sharma, Abheesht; Kim, Taewoon; M Saiful Bari; Fevry, Thibault; Alyafeai, Zaid; Dey, Manan; Santilli, Andrea; Sun, Zhiqing; Ben-David, Srulik; Xu, Canwen; Chhablani, Gunjan; Wang, Han; Jason Alan Fries; Al-shaibani, Maged S.; Sharma, Shanya; Thakker, Urmish; Almubarak, Khalid; Tang, Xiangru; Radev, Dragomir; Mike Tian-Jian Jiang; Rush, Alexander M. (2022). "PromptSource: An Integrated Development Environment and Repository for Natural Language Prompts". arXiv:2202.01279 (https://arxiv.org/abs/2202.01279) [cs.LG (https://arxiv.org/archive/cs.LG)].

16. Wei, Jason; Wang, Xuezhi; Schuurmans, Dale; Bosma, Maarten; Ichter, Brian; Xia, Fei; Chi, Ed H.; Le, Quoc V.; Zhou, Denny (31 October 2022). *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models* (https://proceedings.neurips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html). Advances in Neural Information Processing Systems (NeurIPS 2022). Vol. 35. arXiv:2201.11903 (https://arxiv.org/abs/2201.11903).
17. Wei, Jason; Zhou (11 May 2022). "Language Models Perform Reasoning via Chain of Thought" (https://ai.googleblog.com/2022/05/language-models-perform-reasoning-via.html). *ai.googleblog.com*. Retrieved 10 March 2023.
18. Chen, Brian X. (2023-06-23). "How to Turn Your Chatbot Into a Life Coach" (https://www.nytimes.com/2023/06/23/technology/ai-chatbot-life-coach.html). *The New York Times*.
19. Chen, Brian X. (2023-05-25). "Get the Best From ChatGPT With These Golden Prompts" (https://www.nytimes.com/2023/05/25/technology/ai-chatbot-chatgpt-prompts.html). *The New York Times*. ISSN 0362-4331 (https://search.worldcat.org/issn/0362-4331). Retrieved 2023-08-16.
20. McAuliffe, Zachary. "Google's Latest AI Model Can Be Taught How to Solve Problems" (https://www.cnet.com/tech/services-and-software/googles-latest-ai-model-can-be-taught-how-to-solve-problems/). *CNET*. Retrieved 10 March 2023. " 'Chain-of-thought prompting allows us to describe multistep problems as a series of intermediate steps,' Google CEO Sundar Pichai"
21. McAuliffe, Zachary. "Google's Latest AI Model Can Be Taught How to Solve Problems" (https://www.cnet.com/tech/services-and-software/googles-latest-ai-model-can-be-taught-how-to-solve-problems/). *CNET*. Retrieved 10 March 2023.
22. Sharan Narang and Aakanksha Chowdhery (2022-04-04). "Pathways Language Model (PaLM): Scaling to 540 Billion Parameters for Breakthrough Performance" (https://ai.googleblog.com/2022/04/pathways-language-model-palm-scaling-to.html).
23. Dang, Ekta (8 February 2023). "Harnessing the power of GPT-3 in scientific research" (https://venturebeat.com/ai/harnessing-the-power-of-gpt-3-in-scientific-research/). *VentureBeat*. Retrieved 10 March 2023.
24. Montti, Roger (13 May 2022). "Google's Chain of Thought Prompting Can Boost Today's Best Algorithms" (https://www.searchenginejournal.com/google-chain-of-thought-prompting/450106/). *Search Engine Journal*. Retrieved 10 March 2023.
25. Ray, Tiernan. "Amazon's Alexa scientists demonstrate bigger AI isn't always better" (https://www.zdnet.com/article/amazons-alexa-scientists-demonstrate-bigger-ai-isnt-always-better/). *ZDNET*. Retrieved 10 March 2023.
26. Kojima, Takeshi; Shixiang Shane Gu; Reid, Machel; Matsuo, Yutaka; Iwasawa, Yusuke (2022). "Large Language Models are Zero-Shot Reasoners". arXiv:2205.11916 (https://arxiv.org/abs/2205.11916) [cs.CL (https://arxiv.org/archive/cs.CL)].
27. Dickson, Ben (30 August 2022). "LLMs have not learned our language — we're trying to learn theirs" (https://venturebeat.com/ai/llms-have-not-learned-our-language-were-trying-to-learn-theirs%EF%BF%BC/). *VentureBeat*. Retrieved 10 March 2023.
28. Chung, Hyung Won; Hou, Le; Longpre, Shayne; Zoph, Barret; Tay, Yi; Fedus, William; Li, Yunxuan; Wang, Xuezhi; Dehghani, Mostafa; Brahma, Siddhartha; Webson, Albert; Gu, Shixiang Shane; Dai, Zhuyun; Suzgun, Mirac; Chen, Xinyun; Chowdhery, Aakanksha; Castro-Ros, Alex; Pellat, Marie; Robinson, Kevin; Valter, Dasha; Narang, Sharan; Mishra, Gaurav; Yu, Adams; Zhao, Vincent; Huang, Yanping; Dai, Andrew; Yu, Hongkun; Petrov, Slav; Chi, Ed H.; Dean, Jeff; Devlin, Jacob; Roberts, Adam; Zhou, Denny; Le, Quoc V.; Wei, Jason (2022). "Scaling Instruction-Finetuned Language Models". arXiv:2210.11416 (https://arxiv.org/abs/2210.11416) [cs.LG (https://arxiv.org/archive/cs.LG)].
29. Wei, Jason; Tay, Yi (29 November 2022). "Better Language Models Without Massive Compute" (https://ai.googleblog.com/2022/11/better-language-models-without-massive.html). *ai.googleblog.com*. Retrieved 10 March 2023.

30. Sahoo, Pranab; Singh, Ayush Kumar; Saha, Sriparna; Jain, Vinija; Mondal, Samrat; Chadha, Aman (2024-02-05). "A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications". arXiv:2402.07927 (https://arxiv.org/abs/2402.07927) [cs.AI (https://arxiv.org/archive/cs.AI)].
31. Hu, Hanxu; Lu, Hongyuan; Zhang, Huajian; Song, Yun-Ze; Lam, Wai; Zhang, Yue (2023-10-03). "Chain-of-Symbol Prompting Elicits Planning in Large Language Models". arXiv:2305.10276 (https://arxiv.org/abs/2305.10276) [cs.CL (https://arxiv.org/archive/cs.CL)].
32. Garg, Shivam; Tsipras, Dimitris; Liang, Percy; Valiant, Gregory (2022). "What Can Transformers Learn In-Context? A Case Study of Simple Function Classes". arXiv:2208.01066 (https://arxiv.org/abs/2208.01066) [cs.CL (https://arxiv.org/archive/cs.CL)].
33. Brown, Tom; Mann, Benjamin; Ryder, Nick; Subbiah, Melanie; Kaplan, Jared D.; Dhariwal, Prafulla; Neelakantan, Arvind (2020). "Language models are few-shot learners". *Advances in Neural Information Processing Systems*. **33**: 1877–1901. arXiv:2005.14165 (https://arxiv.org/abs/2005.14165).
34. Liu, Jiacheng; Liu, Alisa; Lu, Ximing; Welleck, Sean; West, Peter; Le Bras, Ronan; Choi, Yejin; Hajishirzi, Hannaneh (May 2022). "Generated Knowledge Prompting for Commonsense Reasoning" (https://aclanthology.org/2022.acl-long.225). *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics: 3154–3169. arXiv:2110.08387 (https://arxiv.org/abs/2110.08387). doi:10.18653/v1/2022.acl-long.225 (https://doi.org/10.18653%2Fv1%2F2022.acl-long.225). S2CID 239016123 (https://api.semanticscholar.org/CorpusID:239016123).
35. Zhou, Denny; Schärli, Nathanael; Hou, Le; Wei, Jason; Scales, Nathan; Wang, Xuezhi; Schuurmans, Dale; Cui, Claire; Bousquet, Olivier; Le, Quoc; Chi, Ed (2022-05-01). "Least-to-Most Prompting Enables Complex Reasoning in Large Language Models". arXiv:2205.10625 (https://arxiv.org/abs/2205.10625) [cs.AI (https://arxiv.org/archive/cs.AI)]. "...least-to-most prompting. The key idea in this strategy is to break down a complex problem into a series of simpler subproblems and then solve them in sequence."
36. Wang, Xuezhi; Wei, Jason; Schuurmans, Dale; Le, Quoc; Chi, Ed; Narang, Sharan; Chowdhery, Aakanksha; Zhou, Denny (2022-03-01). "Self-Consistency Improves Chain of Thought Reasoning in Language Models". arXiv:2203.11171 (https://arxiv.org/abs/2203.11171) [cs.CL (https://arxiv.org/archive/cs.CL)].
37. Diao, Shizhe; Wang, Pengcheng; Lin, Yong; Zhang, Tong (2023-02-01). "Active Prompting with Chain-of-Thought for Large Language Models". arXiv:2302.12246 (https://arxiv.org/abs/2302.12246) [cs.CL (https://arxiv.org/archive/cs.CL)].
38. Fu, Yao; Peng, Hao; Sabharwal, Ashish; Clark, Peter; Khot, Tushar (2022-10-01). "Complexity-Based Prompting for Multi-Step Reasoning". arXiv:2210.00720 (https://arxiv.org/abs/2210.00720) [cs.CL (https://arxiv.org/archive/cs.CL)].
39. Madaan, Aman; Tandon, Niket; Gupta, Prakhar; Hallinan, Skyler; Gao, Luyu; Wiegreffe, Sarah; Alon, Uri; Dziri, Nouha; Prabhumoye, Shrimai; Yang, Yiming; Gupta, Shashank; Prasad Majumder, Bodhisattwa; Hermann, Katherine; Welleck, Sean; Yazdanbakhsh, Amir (2023-03-01). "Self-Refine: Iterative Refinement with Self-Feedback". arXiv:2303.17651 (https://arxiv.org/abs/2303.17651) [cs.CL (https://arxiv.org/archive/cs.CL)].
40. Long, Jieyi (2023-05-15). "Large Language Model Guided Tree-of-Thought". arXiv:2305.08291 (https://arxiv.org/abs/2305.08291) [cs.AI (https://arxiv.org/archive/cs.AI)].
41. Yao, Shunyu; Yu, Dian; Zhao, Jeffrey; Shafran, Izhak; Griffiths, Thomas L.; Cao, Yuan; Narasimhan, Karthik (2023-05-17). "Tree of Thoughts: Deliberate Problem Solving with Large Language Models". arXiv:2305.10601 (https://arxiv.org/abs/2305.10601) [cs.CL (https://arxiv.org/archive/cs.CL)].

42. Jung, Jaehun; Qin, Lianhui; Welleck, Sean; Brahman, Faeze; Bhagavatula, Chandra; Le Bras, Ronan; Choi, Yejin (2022). "Maieutic Prompting: Logically Consistent Reasoning with Recursive Explanations". arXiv:2205.11822 (https://arxiv.org/abs/2205.11822) [cs.CL (https://arxiv.org/archive/cs.CL)].
43. Li, Zekun; Peng, Baolin; He, Pengcheng; Galley, Michel; Gao, Jianfeng; Yan, Xifeng (2023). "Guiding Large Language Models via Directional Stimulus Prompting". arXiv:2302.11520 (https://arxiv.org/abs/2302.11520) [cs.CL (https://arxiv.org/archive/cs.CL)]. "The directional stimulus serves as hints or cues for each input query to guide LLMs toward the desired output, such as keywords that the desired summary should include for summarization."
44. OpenAI (2023-03-27). "GPT-4 Technical Report". arXiv:2303.08774 (https://arxiv.org/abs/2303.08774) [cs.CL (https://arxiv.org/archive/cs.CL)]. *[See Figure 8.]*
45. Eliot, Lance (2023-08-18). "Latest Prompt Engineering Technique Aims To Get Certainty And Uncertainty Of Generative AI Directly On The Table And Out In The Open" (https://www.forbes.com/sites/lanceeliot/2023/08/18/latest-prompt-engineering-technique-aims-to-get-certainty-and-uncertainty-of-generative-ai-directly-on-the-table-and-out-in-the-open/). *Forbes*. Retrieved 2024-08-31. "If you explicitly indicate in your prompt that you want the generative AI to emit a certainty or uncertainty qualification then you will almost certainly get such an indication."
46. Sclar, Melanie; Choi, Yejin; Tsvetkov, Yulia; Suhr, Alane (2024-07-01). "Quantifying Language Models' Sensitivity to Spurious Features in Prompt Design or: How I learned to start worrying about prompt formatting". arXiv:2310.11324 (https://arxiv.org/abs/2310.11324) [cs.CL (https://arxiv.org/archive/cs.CL)].
47. Wahle, Jan Philip; Ruas, Terry; Xu, Yang; Gipp, Bela (2024). Al-Onaizan, Yaser; Bansal, Mohit; Chen, Yun-Nung (eds.). "Paraphrase Types Elicit Prompt Engineering Capabilities" (https://aclanthology.org/2024.emnlp-main.617/). *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Miami, Florida, USA: Association for Computational Linguistics: 11004–11033. arXiv:2406.19898 (https://arxiv.org/abs/2406.19898).
48. Leidinger, Alina; van Rooij, Robert; Shutova, Ekaterina (2023). Bouamor, Houda; Pino, Juan; Bali, Kalika (eds.). "The language of prompting: What linguistic properties make a prompt successful?" (https://aclanthology.org/2023.findings-emnlp.618/). *Findings of the Association for Computational Linguistics: EMNLP 2023*. Singapore: Association for Computational Linguistics: 9210–9232. arXiv:2311.01967 (https://arxiv.org/abs/2311.01967). doi:10.18653/v1/2023.findings-emnlp.618 (https://doi.org/10.18653%2Fv1%2F2023.findings-emnlp.618).
49. Linzbach, Stephan; Dimitrov, Dimitar; Kallmeyer, Laura; Evang, Kilian; Jabeen, Hajira; Dietze, Stefan (June 2024). "Dissecting Paraphrases: The Impact of Prompt Syntax and supplementary Information on Knowledge Retrieval from Pretrained Language Models" (https://aclanthology.org/2024.naacl-long.201/). In Duh, Kevin; Gomez, Helena; Bethard, Steven (eds.). *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. Mexico City, Mexico: Association for Computational Linguistics. pp. 3645–3655. arXiv:2404.01992 (https://arxiv.org/abs/2404.01992). doi:10.18653/v1/2024.naacl-long.201 (https://doi.org/10.18653%2Fv1%2F2024.naacl-long.201).
50. Sclar, Melanie; Choi, Yejin; Tsvetkov, Yulia; Suhr, Alane (2024-07-01). "Quantifying Language Models' Sensitivity to Spurious Features in Prompt Design or: How I learned to start worrying about prompt formatting". arXiv:2310.11324 (https://arxiv.org/abs/2310.11324) [cs.CL (https://arxiv.org/archive/cs.CL)].
51. Polo, Felipe Maia; Xu, Ronald; Weber, Lucas; Silva, Mírian; Bhardwaj, Onkar; Choshen, Leshem; de Oliveira, Allysson Flavio Melo; Sun, Yuekai; Yurochkin, Mikhail (2024-10-30). "Efficient multi-prompt evaluation of LLMs". arXiv:2405.17202 (https://arxiv.org/abs/2405.17202) [cs.CL (https://arxiv.org/archive/cs.CL)].
52. "How Each Index Works - LlamaIndex 🦙 v0.10.17" (https://docs.llamaindex.ai/en/v0.10.17/module_guides/indexing/index_guide.html). *docs.llamaindex.ai*. Retrieved 2024-04-08.

53. Lewis, Patrick; Perez, Ethan; Piktus, Aleksandra; Petroni, Fabio; Karpukhin, Vladimir; Goyal, Naman; Küttler, Heinrich; Lewis, Mike; Yih, Wen-tau; Rocktäschel, Tim; Riedel, Sebastian; Kiela, Douwe (2020). "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks" (https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html). *Advances in Neural Information Processing Systems*. **33**. Curran Associates, Inc.: 9459–9474. arXiv:2005.11401 (https://arxiv.org/abs/2005.11401).
54. *GraphRAG: Unlocking LLM discovery on narrative private data* (https://www.microsoft.com/en-us/research/blog/graphrag-unlocking-llm-discovery-on-narrative-private-data/), 2024
55. Edge, Darren; Trinh, Ha; Cheng, Newman; Bradley, Joshua; Chao, Alex; Mody, Apurva; Truitt, Steven; Larson, Jonathan (2024). "From Local to Global: A Graph RAG Approach to Query-Focused Summarization". arXiv:2404.16130 (https://arxiv.org/abs/2404.16130) [cs.CL (https://arxiv.org/archive/cs.CL)].
56. Sequeda, Juan; Allemang, Dean; Jacob, Bryon (2023). "A Benchmark to Understand the Role of Knowledge Graphs on Large Language Model's Accuracy for Question Answering on Enterprise SQL Databases". arXiv:2311.07509 (https://arxiv.org/abs/2311.07509) [cs.AI (https://arxiv.org/archive/cs.AI)].
57. Singh, Chandan; Morris, John; Aneja, Jyoti; Rush, Alexander; Gao, Jianfeng (October 4, 2022). "Explaining Patterns in Data with Language Models via Interpretable Autoprompting". arXiv:2210.01848 (https://arxiv.org/abs/2210.01848) [cs.LG (https://arxiv.org/archive/cs.LG)].
58. Fernando, Chrisantha; Banarse, Dylan; Michalewski, Henryk; Osindero, Simon; Rocktäschel, Tim (2023). "Promptbreeder: Self-Referential Self-Improvement Via Prompt Evolution". arXiv:2309.16797 (https://arxiv.org/abs/2309.16797). {{cite journal}}: Cite journal requires |journal= (help)
59. Pryzant, Reid; Iter, Dan; Li, Jerry; Lee, Yin Tat; Zhu, Chenguang; Zeng, Michael (2023). "Automatic Prompt Optimization with "Gradient Descent" and Beam Search". arXiv:2305.03495 (https://arxiv.org/abs/2305.03495). {{cite journal}}: Cite journal requires |journal= (help)
60. Guo, Qingyan; Wang, Rui; Guo, Junliang; Li, Bei; Song, Kaitao; Tan, Xu; Liu, Guoqing; Bian, Jiang; Yang, Yujiu (2023). "Connecting Large Language Models with Evolutionary Algorithms Yields Powerful Prompt Optimizers". arXiv:2309.08532 (https://arxiv.org/abs/2309.08532). {{cite journal}}: Cite journal requires |journal= (help)
61. Zhou, Yongchao; Ioan Muresanu, Andrei; Han, Ziwen; Paster, Keiran; Pitis, Silviu; Chan, Harris; Ba, Jimmy (2022-11-01). "Large Language Models Are Human-Level Prompt Engineers". arXiv:2211.01910 (https://arxiv.org/abs/2211.01910) [cs.LG (https://arxiv.org/archive/cs.LG)].
62. Zhang, Zhuosheng; Zhang, Aston; Li, Mu; Smola, Alex (2022-10-01). "Automatic Chain of Thought Prompting in Large Language Models". arXiv:2210.03493 (https://arxiv.org/abs/2210.03493) [cs.CL (https://arxiv.org/archive/cs.CL)].
63. Wei, Jason; Tay, Yi; Bommasani, Rishi; Raffel, Colin; Zoph, Barret; Borgeaud, Sebastian; Yogatama, Dani; Bosma, Maarten; Zhou, Denny; Metzler, Donald; Chi, Ed H.; Hashimoto, Tatsunori; Vinyals, Oriol; Liang, Percy; Dean, Jeff; Fedus, William (31 August 2022). "Emergent Abilities of Large Language Models". arXiv:2206.07682 (https://arxiv.org/abs/2206.07682) [cs.CL (https://arxiv.org/archive/cs.CL)]. "In prompting, a pre-trained language model is given a prompt (e.g. a natural language instruction) of a task and completes the response without any further training or gradient updates to its parameters... The ability to perform a task via few-shot prompting is emergent when a model has random performance until a certain scale, after which performance increases to well-above random"
64. Caballero, Ethan; Gupta, Kshitij; Rish, Irina; Krueger, David (2022). "Broken Neural Scaling Laws". International Conference on Learning Representations (ICLR), 2023.

65. Wei, Jason; Tay, Yi; Bommasani, Rishi; Raffel, Colin; Zoph, Barret; Borgeaud, Sebastian; Yogatama, Dani; Bosma, Maarten; Zhou, Denny; Metzler, Donald; Chi, Ed H.; Hashimoto, Tatsunori; Vinyals, Oriol; Liang, Percy; Dean, Jeff; Fedus, William (31 August 2022). "Emergent Abilities of Large Language Models". arXiv:2206.07682 (https://arxiv.org/abs/2206.07682) [cs.CL (https://arxiv.org/archive/cs.CL)].
66. Musser, George. "How AI Knows Things No One Told It" (https://www.scientificamerican.com/article/how-ai-knows-things-no-one-told-it/). *Scientific American*. Retrieved 17 May 2023. "By the time you type a query into ChatGPT, the network should be fixed; unlike humans, it should not continue to learn. So it came as a surprise that LLMs do, in fact, learn from their users' prompts—an ability known as in-context learning."
67. Johannes von Oswald; Niklasson, Eyvind; Randazzo, Ettore; Sacramento, João; Mordvintsev, Alexander; Zhmoginov, Andrey; Vladymyrov, Max (2022). "Transformers learn in-context by gradient descent". arXiv:2212.07677 (https://arxiv.org/abs/2212.07677) [cs.LG (https://arxiv.org/archive/cs.LG)]. "Thus we show how trained Transformers become mesa-optimizers i.e. learn models by gradient descent in their forward pass"
68. "Mesa-Optimization" (https://www.alignmentforum.org/tag/mesa-optimization). 31 May 2019. Retrieved 17 May 2023. "Mesa-Optimization is the situation that occurs when a learned model (such as a neural network) is itself an optimizer."
69. Garg, Shivam; Tsipras, Dimitris; Liang, Percy; Valiant, Gregory (2022). "What Can Transformers Learn In-Context? A Case Study of Simple Function Classes". arXiv:2208.01066 (https://arxiv.org/abs/2208.01066) [cs.CL (https://arxiv.org/archive/cs.CL)]. "Training a model to perform in-context learning can be viewed as an instance of the more general learning-to-learn or meta-learning paradigm"
70. Monge, Jim Clyde (2022-08-25). "Dall-E2 VS Stable Diffusion: Same Prompt, Different Results" (https://medium.com/mlearning-ai/dall-e2-vs-stable-diffusion-same-prompt-different-results-e795c84adc56). *MLearning.ai*. Retrieved 2022-08-31.
71. "Prompts" (https://docs.midjourney.com/docs/prompts). Retrieved 2023-08-14.
72. "Stable Diffusion prompt: a definitive guide" (https://stable-diffusion-art.com/prompt-guide/). 2023-05-14. Retrieved 2023-08-14.
73. Heikkilä, Melissa (2022-09-16). "This Artist Is Dominating AI-Generated Art and He's Not Happy About It" (https://www.technologyreview.com/2022/09/16/1059598/this-artist-is-dominating-ai-generated-art-and-hes-not-happy-about-it/). *MIT Technology Review*. Retrieved 2023-08-14.
74. Max Woolf (2022-11-28). "Stable Diffusion 2.0 and the Importance of Negative Prompts for Good Results" (https://minimaxir.com/2022/11/stable-diffusion-negative-prompt/). Retrieved 2023-08-14.
75. Gal, Rinon; Alaluf, Yuval; Atzmon, Yuval; Patashnik, Or; Bermano, Amit H.; Chechik, Gal; Cohen-Or, Daniel (2022). "An Image is Worth One Word: Personalizing Text-to-Image Generation using Textual Inversion". arXiv:2208.01618 (https://arxiv.org/abs/2208.01618) [cs.CV (https://arxiv.org/archive/cs.CV)]. "Using only 3-5 images of a user-provided concept, like an object or a style, we learn to represent it through new "words" in the embedding space of a frozen text-to-image model."
76. Kirillov, Alexander; Mintun, Eric; Ravi, Nikhila; Mao, Hanzi; Rolland, Chloe; Gustafson, Laura; Xiao, Tete; Whitehead, Spencer; Berg, Alexander C.; Lo, Wan-Yen; Dollár, Piotr; Girshick, Ross (2023-04-01). "Segment Anything". arXiv:2304.02643 (https://arxiv.org/abs/2304.02643) [cs.CV (https://arxiv.org/archive/cs.CV)].
77. Li, Xiang Lisa; Liang, Percy (2021). "Prefix-Tuning: Optimizing Continuous Prompts for Generation". *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. pp. 4582–4597. doi:10.18653/V1/2021.ACL-LONG.353 (https://doi.org/10.18653%2FV1%2F2021.ACL-LONG.353). S2CID 230433941 (https://api.semanticscholar.org/CorpusID:230433941). "In this paper, we propose prefix-tuning, a lightweight alternative to fine-tuning... Prefix-tuning draws inspiration from prompting"

78. Lester, Brian; Al-Rfou, Rami; Constant, Noah (2021). "The Power of Scale for Parameter-Efficient Prompt Tuning". *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. pp. 3045–3059. arXiv:2104.08691 (https://arxiv.org/abs/2104.08691). doi:10.18653/V1/2021.EMNLP-MAIN.243 (https://doi.org/10.18653%2FV1%2F2021.EMNLP-MAIN.243). S2CID 233296808 (https://api.semanticscholar.org/CorpusID:233296808). "In this work, we explore "prompt tuning," a simple yet effective mechanism for learning "soft prompts"...Unlike the discrete text prompts used by GPT-3, soft prompts are learned through back-propagation"
79. Sun, Simeng; Liu, Yang; Iter, Dan; Zhu, Chenguang; Iyyer, Mohit (2023). "How Does In-Context Learning Help Prompt Tuning?". arXiv:2302.11521 (https://arxiv.org/abs/2302.11521) [cs.CL (https://arxiv.org/archive/cs.CL)].
80. Shin, Taylor; Razeghi, Yasaman; Logan IV, Robert L.; Wallace, Eric; Singh, Sameer (November 2020). "AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts" (https://aclanthology.org/2020.emnlp-main.346). *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics. pp. 4222–4235. doi:10.18653/v1/2020.emnlp-main.346 (https://doi.org/10.18653%2Fv1%2F2020.emnlp-main.346). S2CID 226222232 (https://api.semanticscholar.org/CorpusID:226222232).
81. Willison, Simon (12 September 2022). "Prompt injection attacks against GPT-3" (http://simonwillison.net/2022/Sep/12/prompt-injection/). *simonwillison.net*. Retrieved 2023-02-09.
82. Papp, Donald (2022-09-17). "What's Old Is New Again: GPT-3 Prompt Injection Attack Affects AI" (https://hackaday.com/2022/09/16/whats-old-is-new-again-gpt-3-prompt-injection-attack-affects-ai/). *Hackaday*. Retrieved 2023-02-09.
83. Vigliarolo, Brandon (19 September 2022). "GPT-3 'prompt injection' attack causes bot bad manners" (https://www.theregister.com/2022/09/19/in_brief_security/). *www.theregister.com*. Retrieved 2023-02-09.

Retrieved from "https://en.wikipedia.org/w/index.php?title=Prompt_engineering&oldid=1259992777"