LLMs in technical writing

The fundamentals of technical writing | VUT 2025

The Red Hat Customer Content Services team

About the authors:

Alexandra Nikandrova

Technical Writer for Red Hat

Former Devops

Jana Vrbkova

Technical Writer for Red Hat

Favorite AIs: Google AI on Pixel

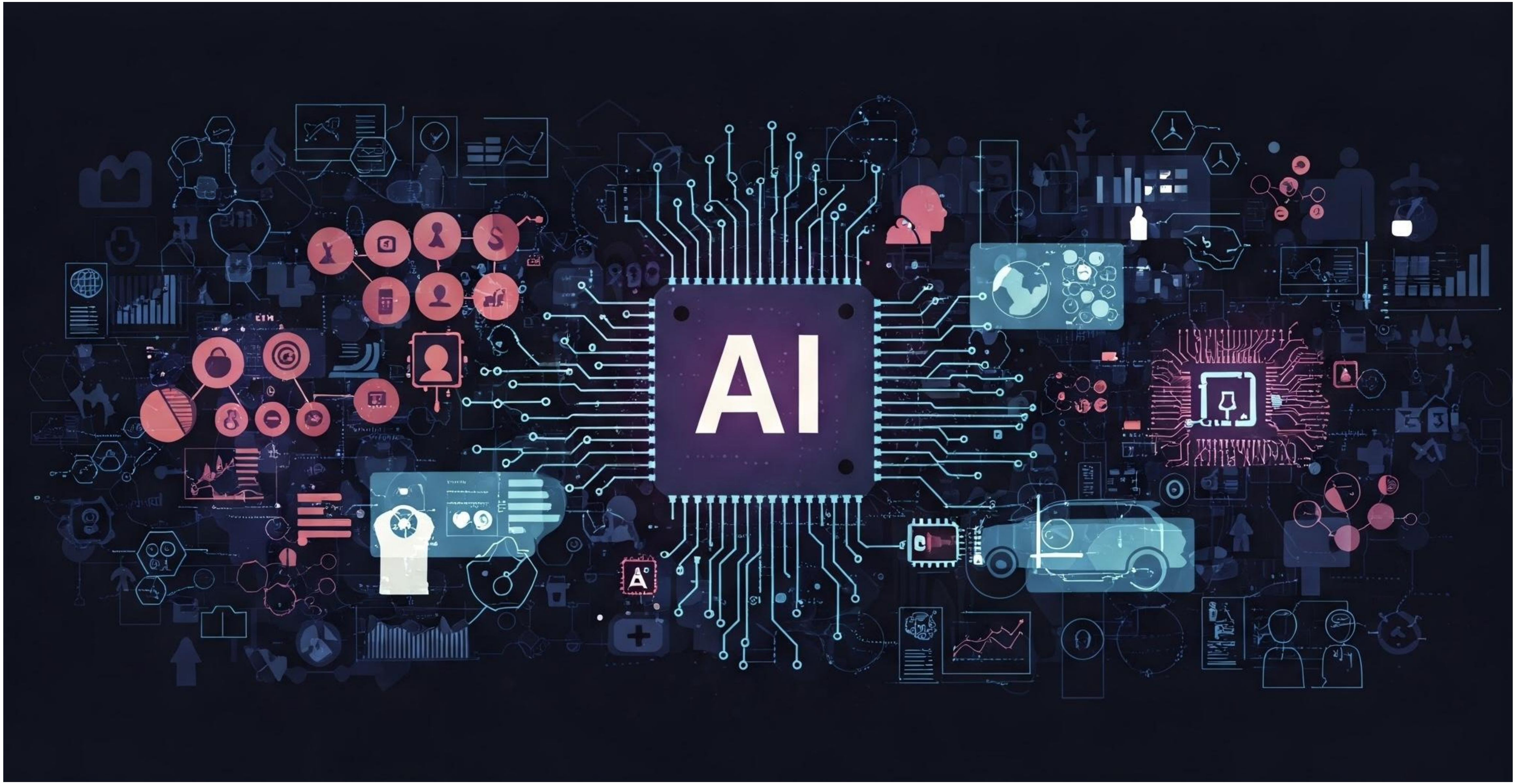
DeepL

Srikanth R

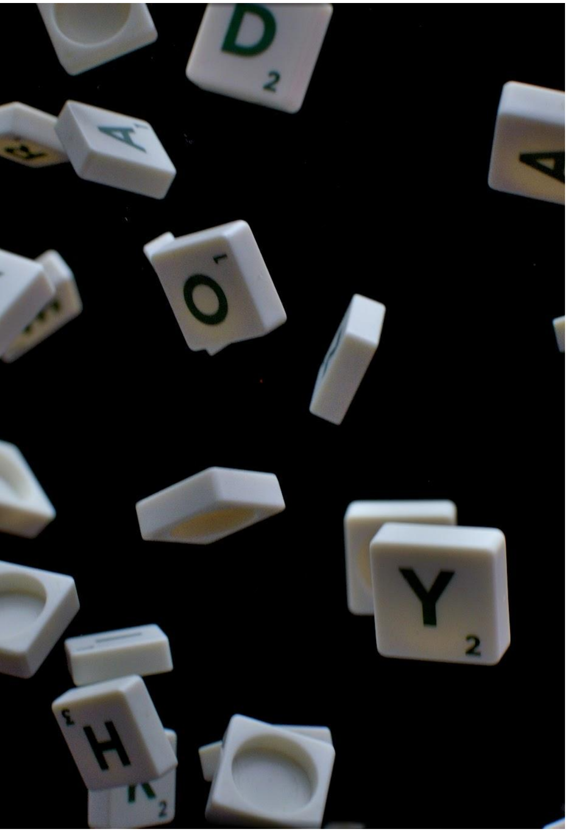
Technical Writer for Red Hat

What we'll discuss today

- What is LLM and SLM?
- Types of LLM
- Use cases of LLM in technical writing
- LLM vs automation
- Pros and cons of LLM
- Prompt engineering



Generated with gemini.google.com

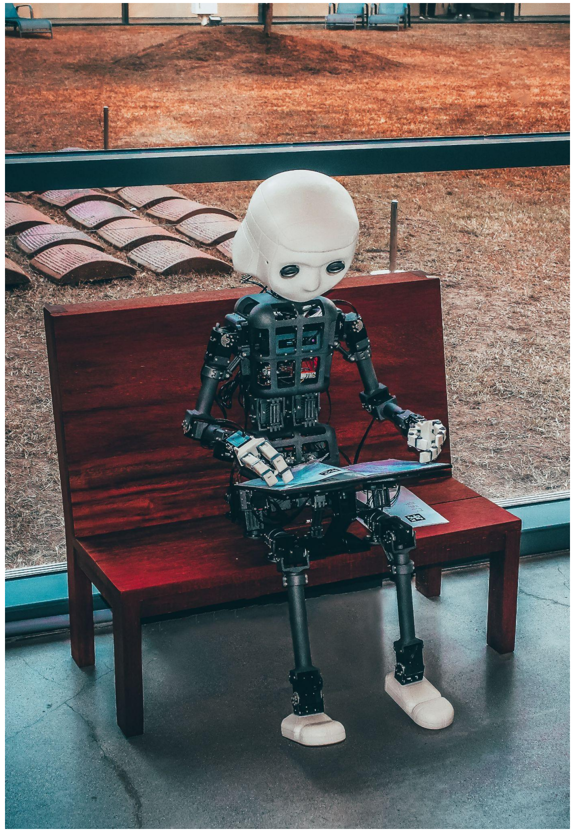


How do large language models work?

- LLMs have trillions of parameters.
- How LLMs work:
  - **Data**
  - **Architecture**
  - **Training**
- They can be **fine-tuned** for specific tasks.
- They iteratively **predict the next token** based on context.

Source: IBM YouTube

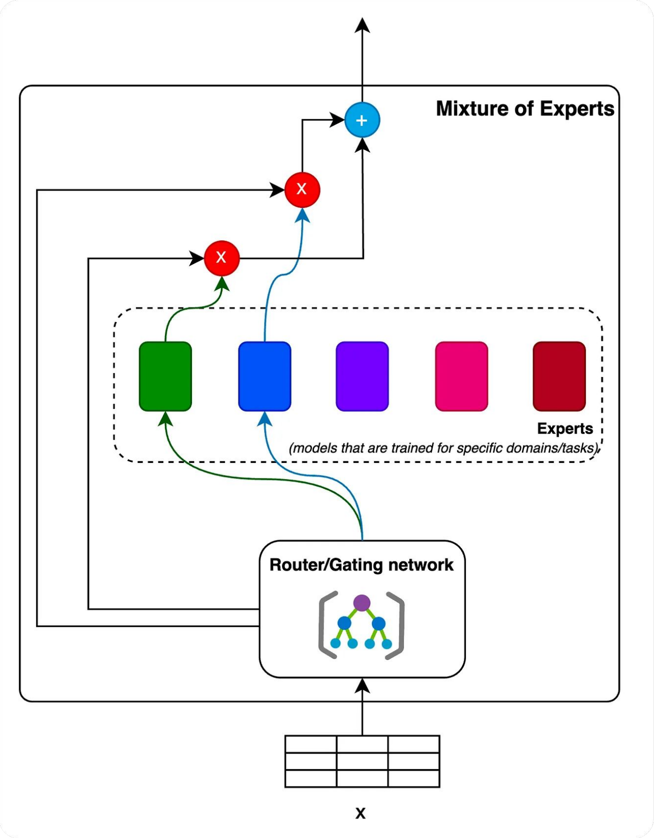
Types of LLMs



- Architecture (encoder, decoder, mixture of experts)
- Training methodology (general purpose LLM, fine-tuned)
- Modality (text-only, multimodal)
- Licencing (proprietary, open-source)

Source: https://unsplash.com/@santesson89

Types of LLMs



Encoder, decoder, mixture of experts

Encoder-only focus on understanding input data

Examples: BERT (Bidirectional Encoder Representations from Transformers)

Decoder-only focus on generating context-aware text

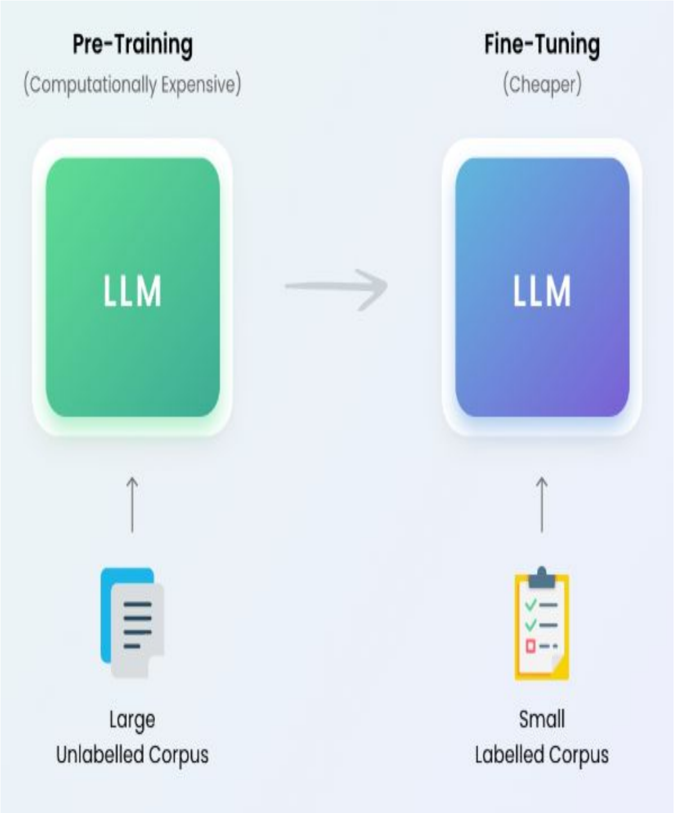
Examples: GPT, LLaMA, Gemma, Claude

Mixture of Experts (MoE) focus on scalability and efficiency of all the models involved

Examples: Qwen3 (30B A3B)

Source: Mixture of experts

Types of LLMs



General purpose vs Fine-tuned

General purpose LLMs are trained on a wide range of data—can handle most tasks but not optimized for one domain.

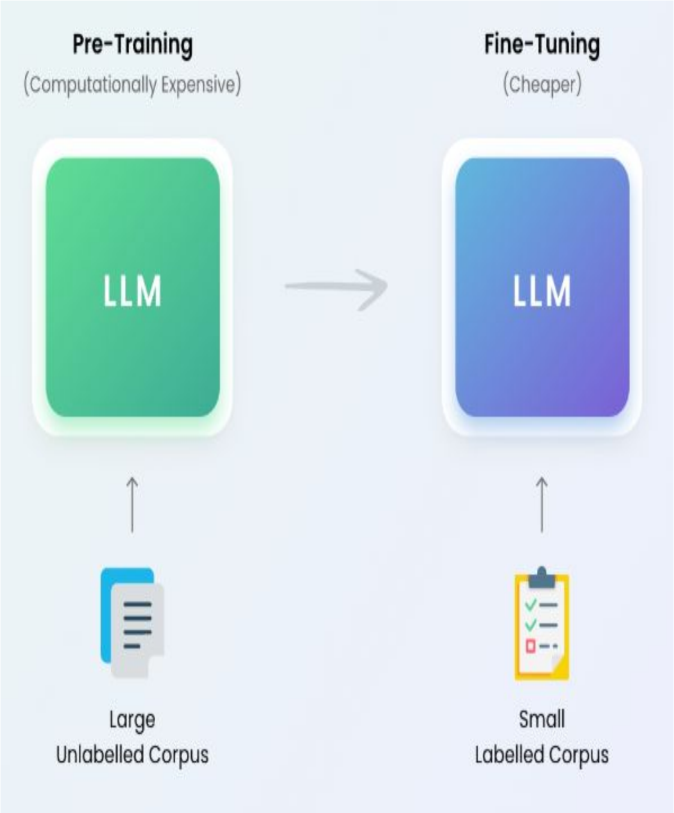
Examples: ChatGPT, Claude, Gemini

Fine tuned LLMs are Specialized on specific domains like legal, medical, or technical documentation.

Examples: Writer.com (custom LLM), CodeLlama (for code), BioGPT (for biomedical content)

Source: https://unsplash.com/@santesson89

Types of LLMs

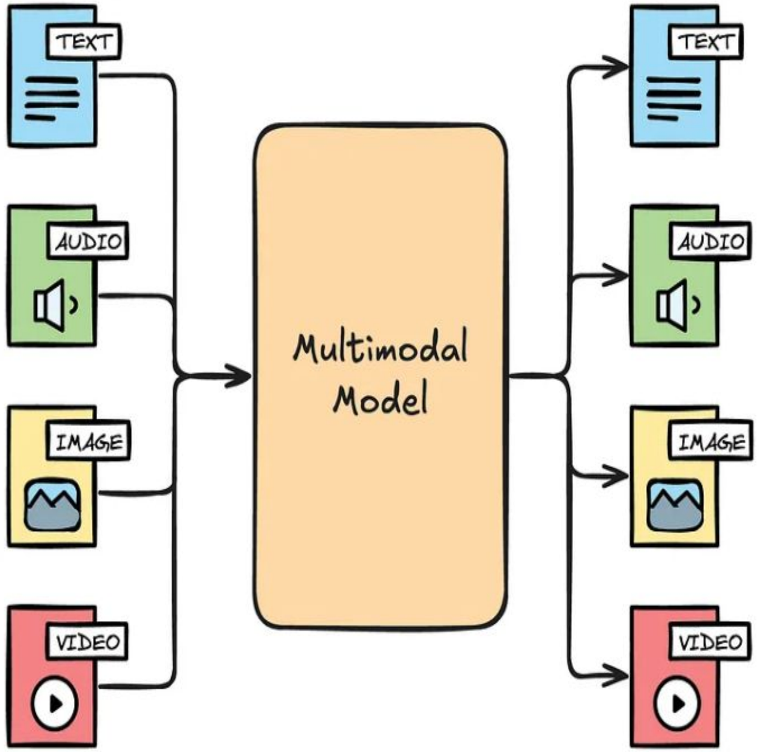


Retrieval-Augmented Generation (RAG)

General purpose model fine-tuned to your needs.

Source: https://unsplash.com/@santesson89

Types of LLMs



Text-only vs Multi-modal

Text-only LLMs are primarily designed to understand, analyze, and generate text-based content.

Examples: GPT-3, LLaMA, BLOOM.

Use Cases: Text generation, translation, summarization, question answering, chatbots.

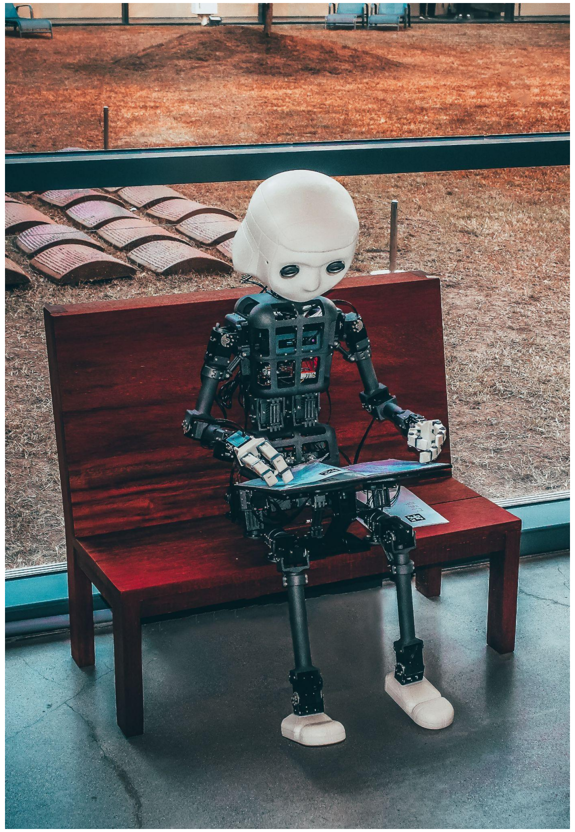
Multi-modal LLMs are capable of processing and generating content across multiple data types, such as text, images, audio, and video.

Examples: Flamingo (DeepMind), Kosmos-1 (Microsoft), LLaVA (open-source), PaLM-E (Google).

Use Cases: Computer Vision, Audio-Visual AI, Robotics, Content Creation, and so on.

Source: https://unsplash.com/@santesson89

Types of LLMs



Open-source vs Proprietary

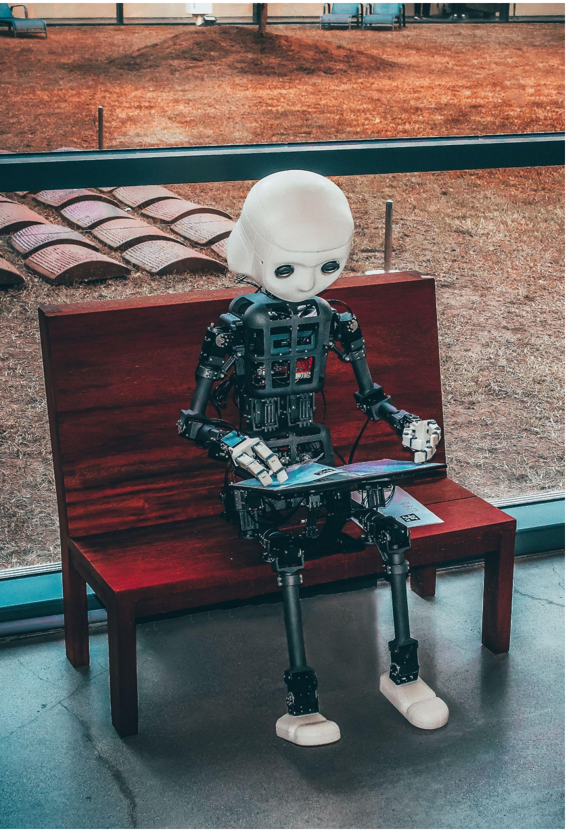
Open-source LLMs are publicly available, allowing for free use, modification, and distribution, enabling extensive customization.

Examples: Llama 3, Falcon, WizardLM, and so on.

Proprietary LLMs have limited access and customization options, often offered with strong performance and robust support.

Examples: OpenAIs GPT-4, Google's Gemini, Anthropic's Claude, and so on.

Source: https://unsplash.com/@santesson89



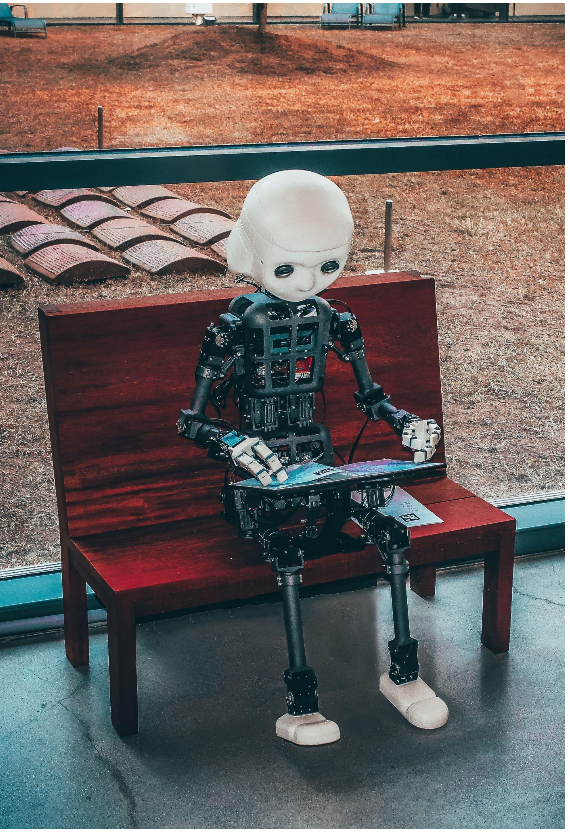
How do you use AI tools?

Source: https://unsplash.com/@santesson89

LLM use cases in TW

- Brainstorming: “Give me 5 headings for this paragraph”
- Research, explanations, summarisation: “How does SELinux work?”
- Missing prerequisites: “What do I need to compete this procedure”
- Description of code: “What does this yaml file do?”
- Code snippets or regex: “Create a script that…”
- Formatting: “Convert this csv into a markdown table”
- Ideas for visual elements (runwayml.com)

Source: unsplash.com/@rechaoktaviani



Automation vs AI

Source: https://unsplash.com/@santesson89

When to use automation

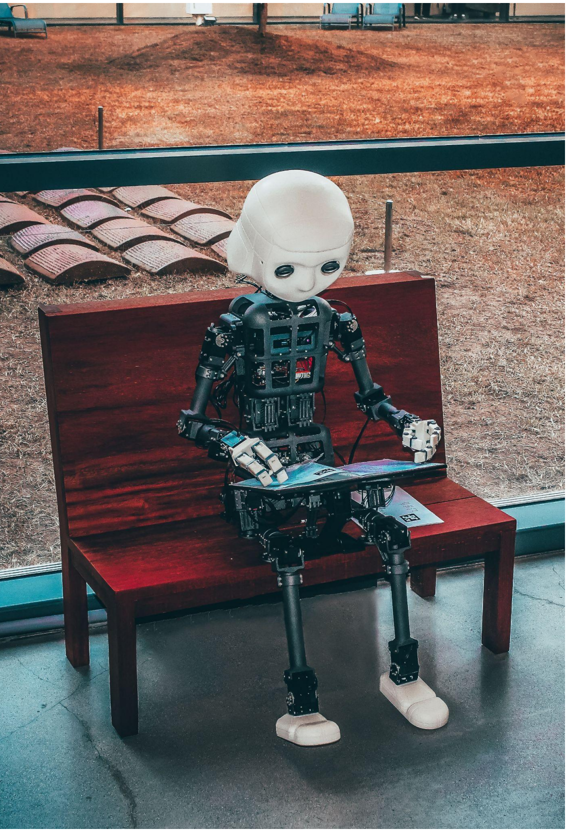
- Predictable, repetitive rules (processing files, system monitoring)
- Low complexity
- Speed
- Limited data (no data required)
- Low cost (generally cheaper to implement)

Source: unsplash.com/@rechaoktaviani

When to use AI

- Pattern recognition with no clear rules
- Processing natural language
- Modeling

Source: unsplash.com/@rechaoktaviani



Pros and cons

Source: https://unsplash.com/@santesson89

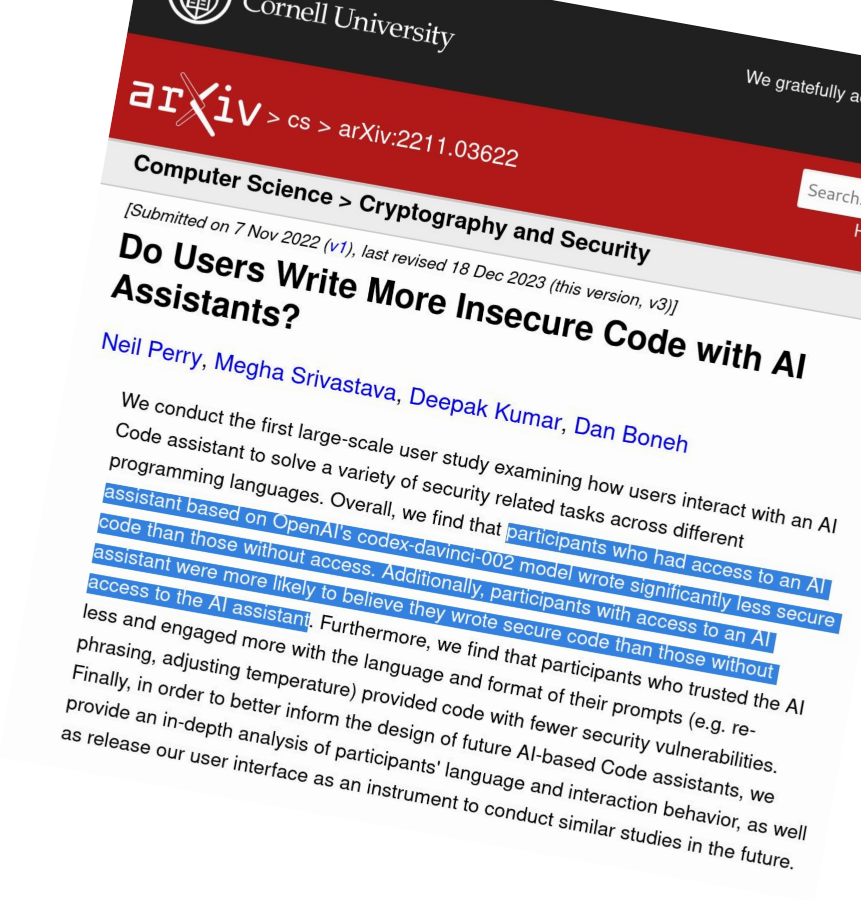
Pros of LLMs

- Enhanced efficiency
- Automation
- Reduced human error
- Reduced physical risks
- Improved customer experience
- Democratization of skills

https://www.consilium.europa.eu/en/policies/benefits-and-risks-of-ai/
https://www.ibm.com/think/topics/artificial-intelligence
https://www.forbes.com/advisor/in/business/software/advantages-of-ai/

Cons of LLMs

- No concept of truth
- Hallucinations
- Legal limitations
- Security and privacy concerns
- Environmental impact
- De-skilling



Cornell University

We gratefully a

arXiv > cs > arXiv:2211.03622

Search...

Computer Science > Cryptography and Security

[Submitted on 7 Nov 2022 (v1), last revised 18 Dec 2023 (this version, v3)]

Do Users Write More Insecure Code with AI Assistants?

Neil Perry, Megha Srivastava, Deepak Kumar, Dan Boneh

We conduct the first large-scale user study examining how users interact with an AI Code assistant to solve a variety of security related tasks across different programming languages. Overall, we find that participants who had access to an AI assistant based on OpenAI's codex-davinci-002 model wrote significantly less secure code than those without access. Additionally, participants with access to an AI assistant were more likely to believe they wrote secure code than those without access to the AI assistant. Furthermore, we find that participants who trusted the AI less and engaged more with the language and format of their prompts (e.g. re-phrasing, adjusting temperature) provided code with fewer security vulnerabilities. Finally, in order to better inform the design of future AI-based Code assistants, we provide an in-depth analysis of participants' language and interaction behavior, as well as release our user interface as an instrument to conduct similar studies in the future.

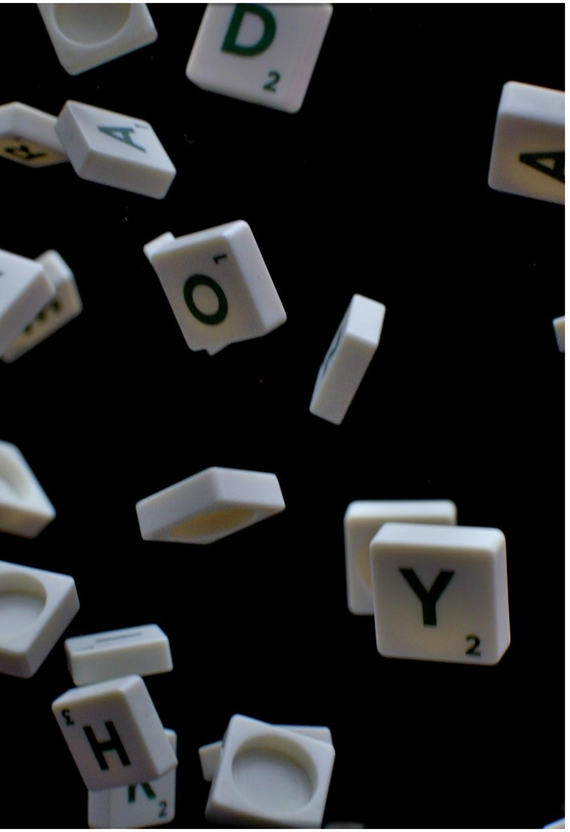
Sources: unsplash.com/@jannerboy62
https://arxiv.org/abs/2211.03622
https://dl.acm.org/doi/10.1145/3491101.3519665
https://www.technologyreview.com/2025/05/20/1116327/ai-energy-usage-climate-footprint-big-tech/
https://www.nature.com/articles/s41586-024-07566-y#Sec2
https://www.sciencedirect.com/science/article/abs/pii/S0007681324000272

Prompt Engineering

Source: gemini

Which text is written by a human TW?

A	B	C
Red Hat Enterprise Linux 9: GIMP 2.99.8 Technology Preview Available This release introduces GNU Image Manipulation Program (GIMP) version 2.99.8 as a Technology Preview in Red Hat Enterprise Linux 9. GIMP 2.99.8 delivers a comprehensive set of image editing and manipulation capabilities, enabling users to: ● **Perform basic image editing tasks:** Crop, resize, adjust color and brightness, apply filters, and more. ● **Create and manipulate artwork:** Draw vector graphics, paint with various brushes, and work with layers for complex compositions. ● **Automate repetitive tasks:** Utilize scripting capabilities to automate workflows and save time. **Important Note:** GIMP 2.99.8 is provided as a Technology Preview. This means it is still under development and may contain bugs or stability issues. It is not intended for production use. This Technology Preview allows users to explore the capabilities of GIMP 2.99.8 before its official introduction in Red Hat Enterprise Linux 9. Feedback from users will be crucial in shaping the final offering. **Installation:** GIMP 2.99.8 is available through the Technology Preview repository. You can install it using the following command: `sudo dnf install gimp-techpreview` **Disclaimer:** Red Hat does not provide support for Technology Preview features. Use them at your own risk and discretion.	**Feature: GNU Image Manipulation Program (GIMP) 2.99.8** **Description:** The GIMP package version 2.99.8 is now available as a Technology Preview in Red Hat Enterprise Linux 9. This pre-release version provides a glimpse into the upcoming GIMP 3 release. Users can explore and test this version using the RPM package for installation. **Key Points:** ● **Technology Preview Status:** ○ GIMP 2.99.8 is presented as a Technology Preview, showcasing features and enhancements that will be fully integrated into the official GIMP 3 release in future versions of RHEL 9. ● **Pre-Release Version:** ○ This package serves as a pre-release version, allowing users to experience and provide feedback on the latest developments in GIMP. **Recommended Action:** ● Users interested in previewing the upcoming features in GIMP 3 can install the GIMP 2.99.8 package. ● As a Technology Preview, this version is not intended for production use but offers users an opportunity to familiarize themselves with the upcoming GIMP release. **Note:** GIMP 2.99.8 is a forward-looking Technology Preview, giving users early access to the evolving capabilities of GIMP 3. Feedback from the community during this preview phase is valuable in refining and enhancing the user experience before the official release.	**GIMP available as a Technology Preview in RHEL 9** GNU Image Manipulation Program (GIMP) 2.99.8 is now available in RHEL 9 as a Technology Preview. The `gimp` package version 2.99.8 is a pre-release version with a set of improvements, but a limited set of features and no guarantee for stability. As soon as the official GIMP 3 is released, it will be introduced into RHEL 9 as an update of this pre-release version. In RHEL 9, you can install `gimp` easily as an RPM package.



How to write prompts for LLM?

- Start simple
- Instruct
- Specify and direct
- To do

Source: IBM YouTube

Takeaways

- LLMs are part of a foundational model
- Requires a lot of data and computing power to train
- Automation can be used cheaper and faster in some cases
- **AI cannot think**, only predict the next token. You need to have the following skills: user focus, minimalism, stylistics, applying a style guide

Source: unsplash.com/@rechaoktaviani

Exercise

gemini.google.com

Source: unsplash.com/@rechaoktaviani

- Access documentation
- Download a PDF file

Learn prompt engineering

[Prompt engineering guide]

Source: unsplash.com/@rechaoktaviani

EXERCISE: Now you try it!

Cause: Too restricted checks in openssl

Consequence: random tls connection failures

Workaround (if any): just reconnect

Result: handshake succeeds

gemini.google.com

bing.com/chat

chat.openai.com

Exercise: Tips for prompting

"Document how to manage and configure passkey authentication devices in the Identity Management (IdM) environment."

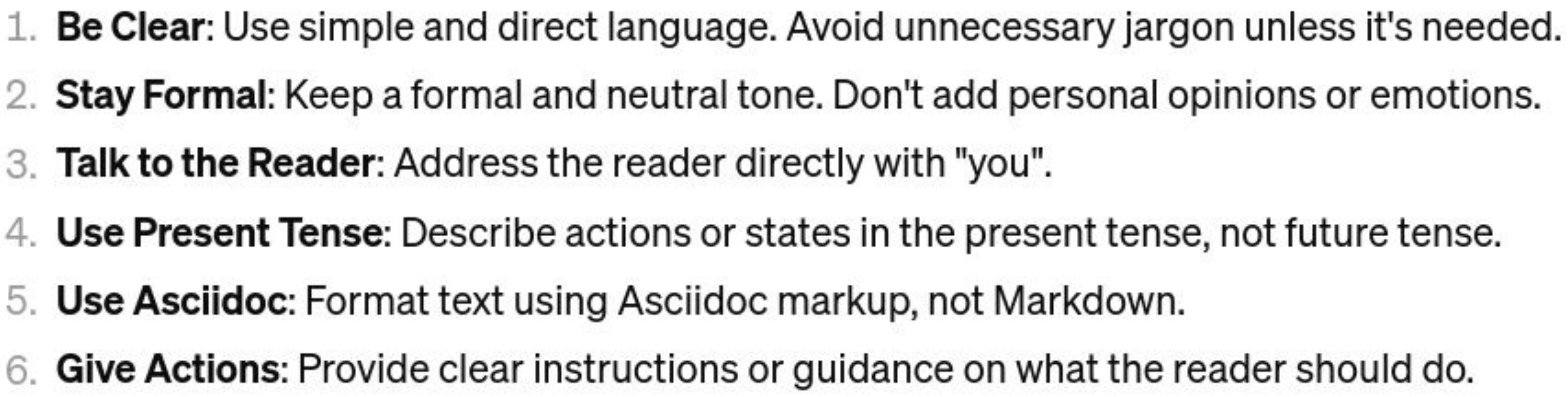
- Provide contextual details.
- Be clear and specific in your prompts.
- Focus on one aspect at the time (content format, structure, language).
- Work one section at a time (heading, abstract, procedural steps, verification…).
- Provide examples of the style or tone you're aiming for.
- Refine the prompts using different wording to get better results.
- Feed ChatGPT with content or reference material.

Exercise: Tips for prompting

Bonus: Custom instructions (system prompt)

Go to **Your profile** > **Customize ChatGPT** to customize the responses.

For example, create your own style guide:



1. **Be Clear:** Use simple and direct language. Avoid unnecessary jargon unless it's needed.
2. **Stay Formal:** Keep a formal and neutral tone. Don't add personal opinions or emotions.
3. **Talk to the Reader:** Address the reader directly with "you".
4. **Use Present Tense:** Describe actions or states in the present tense, not future tense.
5. **Use Asciidoc:** Format text using Asciidoc markup, not Markdown.
6. **Give Actions:** Provide clear instructions or guidance on what the reader should do.

Technical writing is not just writing

- Research of new topics
- Planning, content strategy, user/content journey
- Verification and testing
- User experience

Source: unsplash.com/@rechaoktaviani

Thank you