LLM & GPT Academy

Your Step-by-Step Guide to Language Models,
Prompt Engineering, and Building Custom AI Agents

Level 1

📝 Course Syllabus: Understanding and Building with LLMs/GPTs

Target Audience

- Learners new to large language models (LLMs), prompt engineering, and custom GPTs.
- Educators, students, hobbyists, and professionals wanting practical skills.

Course Goals

- Explain the foundations of LLMs and GPT models in simple language.
- Teach hands-on skills: prompt engineering, evaluation, and responsible use.
- Guide learners through building and customizing their own GPT agents.
- Equip learners to use LLMs ethically and effectively.

Syllabus Outline

Unit 1: Introduction to LLMs and GPTs

- 1.1 What is a Language Model?
- 1.2 Brief History of LLMs (GPT-1 to GPT-4+)
- 1.3 Key Concepts (Tokens, Parameters, Context, Training)
- 1.4 Capabilities and Limitations

Unit 2: Inside a GPT

- 2.1 How Do LLMs Work? (Simplified Architecture)
- 2.2 Common LLM Tasks: Text Generation, Summarization, Q&A, Coding
- 2.3 Understanding Prompts and Responses
- 2.4 What Makes a Good Prompt?

Unit 3: Prompt Engineering

- 3.1 Principles of Prompt Engineering
- 3.2 Writing Effective Prompts: Do's and Don'ts
- 3.3 Prompt Patterns (Instruction, Chain-of-Thought, Role Play, etc.)
- 3.4 Troubleshooting Bad Outputs

Unit 4: Evaluating LLMs

- 4.1 Benchmarking and Metrics (Accuracy, BLEU, ROUGE, etc.)
- 4.2 Human-in-the-Loop Evaluation
- 4.3 Safety and Reliability Testing

Unit 5: Building a Custom GPT

- 5.1 What is a Custom GPT? Use Cases
- 5.2 Ingredients for a Custom GPT (Manifest, Main File, Data)
- 5.3 Step-by-Step: Creating Your Own GPT
- 5.4 Practical Example: Educational GPT Tutor

Unit 6: Advanced Topics and Responsible AI

- 6.1 Bias, Safety, and Ethical Use
- 6.2 Explainability and Transparency
- 6.3 Security Risks and Mitigations
- 6.4 Keeping Up with LLM Advances

Unit 7: Hands-On Projects and Exercises

- 7.1 Guided Project: Your First Custom GPT
- 7.2 Prompt Engineering Challenge
- 7.3 LLM Benchmarking Exercise
- 7.4 Ethical Dilemma Scenarios

Appendices

- A. LLM Glossary
- B. Further Reading and Resources
- C. Sample Prompts and Templates
- D. Troubleshooting and FAQ

Unit 1: Introduction to LLMs and GPTs

1.1 What is a Language Model?

A **language model** is a computer program trained to predict and generate language—like guessing the next word in a sentence or writing a paragraph based on a prompt.

- **Analogy:**
  Imagine playing a game where you complete each other's sentences. A language model has "read" billions of sentences and learned the patterns, so it's great at this game!

- **Why is this useful?**
  It lets computers answer questions, write code, summarize text, and even hold conversations.

1.2 Brief History of LLMs (GPT-1 to GPT-4+)

- **Early Days:**
  Before 2018, most language models were smaller and only worked on specific tasks.

- **GPT-1 (2018):**
  OpenAI's first "Generative Pretrained Transformer." It used a new architecture (transformer) and could do many language tasks with one model.

- **GPT-2 (2019):**
  Much bigger. Could generate coherent paragraphs, stories, and even some code.

- **GPT-3 (2020):**
  A giant leap: 175 billion parameters. Great at answering questions, writing essays, and creative tasks.

- **GPT-4 (2023):**
  Even more powerful, more reliable, and better at understanding complex instructions and images.

- **Today:**
  There are many LLMs (not just from OpenAI): Google Gemini, Anthropic Claude, Meta Llama, Mistral, etc. The field evolves rapidly!

1.3 Key Concepts

- **Tokens:**
  LLMs don't "see" text as whole words or sentences. They split everything into small pieces called *tokens* (can be words, parts of words, or even punctuation).

- **Parameters:**
  The "memory" of the model—think of them like dials or settings in the brain. More parameters usually mean more knowledge, but also need more compute.

- **Context Window:**
  How much text the model can "see" at once. Larger context windows allow models to handle longer conversations or documents.

- **Training:**
  The process of teaching the model by feeding it huge amounts of text and letting it learn the patterns.

1.4 Capabilities and Limitations

What LLMs Do Well:

- Answer questions
- Summarize articles
- Write stories, code, emails
- Translate between languages
- Analyze data, patterns

But They Also Have Limits:

- **No real-world understanding:** They don't "think" or "know" like humans—just pattern match.
- **Hallucination:** Sometimes they make up facts that sound plausible.
- **Bias:** Models may reflect biases in their training data.
- **Context limitations:** May forget information from earlier in a conversation if it's too long.

Quick Quiz & Reflection

1. **What is a "token" in an LLM?**
2. **What does "GPT" stand for?**
3. **Can LLMs think or understand like humans? Why or why not?**
4. **What are some strengths and weaknesses of LLMs?**

Take a minute to answer these questions to check your understanding!

Hands-On Activity: Play the Prediction Game

With a partner or by yourself, try this:

- Write the start of a sentence (e.g., "The cat sat on the...").
- Try to guess the next word as an LLM might.
- See how many reasonable completions you can come up with.
- Now, type your prompt into ChatGPT or another LLM—how does it complete your sentence?

Summary

- Language models predict and generate text based on patterns.
- LLMs like GPT-4 are huge, powerful, but not magical—they have strengths and weaknesses.
- Understanding the basics is the first step to using and building with LLMs!

Unit 2: Inside a GPT

2.1 How Do LLMs Work? (Simplified Architecture)

An LLM like GPT is based on a model called a **transformer**.
Think of it like a super-advanced "autocomplete" on steroids.

- **How does it predict text?**
    - The model "looks" at the words (tokens) before the blank, then tries to guess what comes next—over and over, to generate text.
- **Key pieces of the transformer:**
    - **Attention:**
      Like reading a paragraph and deciding which words are most important to the meaning.
    - **Layers:**
      Multiple stages where the model refines its understanding, each time making a better guess.
    - **Parameters:**
      These are like millions (or billions) of knobs tuned during training to get the best predictions.
- **Training:**
    - The model is trained by reading huge amounts of text, over and over, learning the patterns by adjusting its parameters.

Analogy:
Imagine teaching a student to finish your sentences. The more examples they see, the better they get—even without understanding the world as you do.

2.2 Common LLM Tasks

LLMs are incredibly versatile. Here are some things they do best:

- **Text Generation:**
  Writing stories, blog posts, emails, or code.

- **Summarization:**
  Making long texts shorter and more readable.
- **Question Answering:**
  Responding to user queries, like a search engine but more conversational.
- **Translation:**
  Turning text from one language into another.
- **Code Assistance:**
  Writing, explaining, or fixing code snippets.

2.3 Understanding Prompts and Responses

A **prompt** is whatever you give the model as input—your question, instruction, or context.

- **Prompt Example:**
  "Explain what a token is in language models."

The **response** is what the model generates based on the prompt.

- **Prompt Engineering:**
  Crafting better prompts leads to better results! Small changes to your prompt can make a big difference.

2.4 What Makes a Good Prompt?

Good prompts are:

- **Clear:**
  State exactly what you want.
- **Specific:**
  Give details or examples.
- **Concise:**
  Don't overload the model with unnecessary info.

Compare:

- Weak prompt:
  "Explain LLMs."

- Strong prompt:
  "In simple language, explain how a large language model like GPT predicts the next word in a sentence. Give a real-world analogy."

Prompt Engineering Tip:
Try asking for an example, analogy, or step-by-step answer if you want more detail or clarity.

Quick Quiz

1. **What does a transformer "pay attention" to when generating text?**
2. **Name three tasks LLMs are commonly used for.**
3. **Why is prompt clarity important?**
4. **What is "prompt engineering"?**

Hands-On Activity: Prompt Comparison

1. Write a prompt asking about LLMs in two ways—one vague, one detailed.
2. Enter both into an LLM (like ChatGPT).
3. Compare the answers. Which was better? Why?

Summary

- LLMs work by predicting text, using transformers with layers and attention.
- They perform many tasks—writing, summarizing, translating, coding, and more.
- A well-crafted prompt is the key to getting great results.

Unit 3: Prompt Engineering

3.1 Principles of Prompt Engineering

Prompt engineering is the art and science of crafting inputs (prompts) to get the best results from a language model.

- **Why is it important?**
  The same model can give wildly different outputs based on how you ask. Clear, well-structured prompts can dramatically improve accuracy and usefulness.

3.2 Writing Effective Prompts: Do's and Don'ts

Do's:

- **Be specific:**
  Instead of "Tell me about dogs," try "List 3 unique facts about golden retrievers suitable for a school project."
- **Give context:**
  Tell the model who it is ("You are a helpful math tutor…") or set the scene.
- **Ask for format:**
  Request bullet points, tables, code blocks, step-by-step answers, etc.
- **Set constraints:**
  Specify word limits, style (e.g., "in simple language"), or audience.

Don'ts:

- **Avoid vagueness:**
  "Help me with homework" is too broad.
- **Don't overload:**
  Keep prompts focused—too many tasks in one prompt can confuse the model.
- **Don't forget boundaries:**
  If you want only facts (no opinions), say so!

3.3 Prompt Patterns

Different patterns help you get the output you want. Here are some common ones:

- **Instruction:**
  "Summarize the following article in 3 bullet points: [text]"

- **Chain-of-Thought:**
  "Explain your reasoning step by step before giving the final answer."

- **Role Play:**
  "You are an expert Python programmer. Explain what a function does."

- **Examples (Few-Shot):**
  "Rewrite the sentence to be more formal.
  Example: 'Hey, what's up?' → 'Good afternoon, how are you?'
  Sentence: 'Gotta go, see ya!'"

3.4 Troubleshooting Bad Outputs

Common Problems:

- **Too vague/generic:**
  Output isn't useful or is repetitive.

- **Hallucination:**
  The model invents facts or misquotes.

- **Off-topic:**
  The answer drifts away from your question.

Fixes:

- **Clarify your request:**
  Add more details or constraints.

- **Ask for sources:**
  "List your sources" or "cite references" can improve reliability.

- **Break it down:**
  Split complex requests into smaller steps or prompts.

Quick Quiz

1. **Why is it helpful to ask for a specific format in your prompt?**
2. **What is a "chain-of-thought" prompt?**
3. **Give an example of a "role play" prompt.**
4. **What should you do if the model gives a vague answer?**

Hands-On Activity: Prompt Remix

1. Write a weak prompt (e.g., "Tell me about AI").
2. Make it better using the do's and patterns above.
3. Try both with a real LLM—compare the responses.

Summary

- Prompt engineering is crucial for getting good results from LLMs.
- Use clear, specific, and well-structured prompts.
- Practice different prompt patterns and troubleshoot when needed.

Unit 4: Evaluating LLMs

4.1 Benchmarking and Metrics

Evaluating an LLM means measuring how well it performs at different tasks. There are two main ways to do this:

- **Automatic (quantitative) metrics**
- **Human (qualitative) evaluation**

Common Metrics

- **Accuracy:**
  Does the model give the right answer (e.g., in a quiz or test set)?
- **BLEU / ROUGE:**
  Used for translation or summarization tasks; compares the model's output to "gold standard" answers.
- **F1 Score:**
  Measures balance between precision (how many selected items are relevant) and recall (how many relevant items are selected).
- **Perplexity:**
  How "surprised" the model is by the correct answer. Lower is better.

Example:
For summarization, you might use ROUGE to compare the LLM's summary to a human-written one.

4.2 Human-in-the-Loop Evaluation

Not all LLM tasks have simple "right or wrong" answers, so we need human judgment!

- **Human Evaluation:**
  People rate model outputs for relevance, helpfulness, correctness, or creativity.
- **Rubrics:**
  Clear criteria make evaluations fair and consistent (e.g., "Clarity: 1-5 stars").

- **Pairwise Comparison:**
  Humans pick the better of two answers, often used to tune LLMs (as in RLHF—Reinforcement Learning from Human Feedback).

4.3 Safety and Reliability Testing

LLMs are powerful but can make mistakes or even cause harm if not tested carefully.

- **Adversarial Testing:**
  Try "tricky" prompts to see if the model gives unsafe, biased, or nonsensical answers.
- **Bias Detection:**
  Check if the model's answers are skewed or unfair in certain contexts.
- **Red Teaming:**
  Security and safety experts probe the model for vulnerabilities or misuse potential.

Best Practice:
Always review important outputs manually, especially in sensitive or high-stakes settings.

Quick Quiz

1. **What is "perplexity" in LLM evaluation?**
2. **Why is human evaluation important?**
3. **Name one way to test an LLM's safety.**
4. **What does "bias" mean in the context of LLMs?**

Hands-On Activity: Evaluate the Evaluator

1. Pick a simple task (e.g., summarize a news article).
2. Try it with an LLM.
3. Rate the output on a scale of 1–5 for:
   - Clarity
   - Factual accuracy
   - Usefulness

4. (Optional) Compare with a friend or classmate—do you agree on the ratings?

Summary

- LLMs are evaluated using a mix of automated metrics and human judgment.
- Safety, reliability, and bias checks are critical.
- Clear evaluation makes LLMs safer and more useful in the real world.

Unit 5: Building a Custom GPT

5.1 What is a Custom GPT? Use Cases

A **custom GPT** is a version of a language model that's tailored for specific needs, topics, or tasks.
You can adjust its personality, knowledge, and abilities—turning it into a tutor, assistant, creative partner, or specialist.

Example Use Cases:

- Math tutor for students
- Code reviewer for developers
- Legal research assistant
- Storytelling or creative writing partner
- Customer service bot
- Custom tools for businesses

5.2 Ingredients for a Custom GPT

Every custom GPT has a few core "ingredients":

Ingredient	Description
Manifest file	Blueprint describing the agent's name, abilities, APIs, modules
Main Python file	Core logic: how the agent processes input and generates responses
Knowledge/data	Documents, FAQs, examples, or datasets your agent uses
Dependencies	Required Python libraries/tools
Instructions	Guidance for tone, safety, and behavior

5.3 Step-by-Step: Creating Your Own GPT

Step 1: Define Your Agent's Mission

- What problem will your GPT solve?
- Who is it for (students, coders, researchers, etc.)?
- What kind of personality or style should it have?

Step 2: Write the Manifest File

This is your agent's "recipe."
It specifies the name, main script, capabilities, modules, and resources.

Example snippet:

```
json
{
  "agent_name": "MathTutorGPT",
  "description": "A friendly math homework helper.",
  "entry_point": "mathtutor.py",
  "modules": ["Memory", "QuizGenerator"],
  "dependencies": ["numpy"]
}
```

Step 3: Implement the Main Python File

- Write the class with the agent's logic and response rules.
- Connect it to your modules and data.
- Make sure it follows the tone and behavior you want.

Example skeleton:

```
python
class MathTutorGPT:
    def answer(self, question):
        # Process input and return a helpful answer
        return "Let's solve this step by step…"
```

Step 4: Add Knowledge and Data

- Upload documents, examples, or datasets your agent can draw from.
- Reference these in your agent's logic or instructions.

Step 5: Deploy and Test

- Launch the agent using your platform (e.g., GPT Store, API, local server).
- Test with real prompts—fix any issues, improve instructions.

5.4 Practical Example: Educational GPT Tutor

Suppose you want to build an LLM educator (like Angela):

- **Manifest:**
  States modules like "Glossary," "Quiz Engine," "Ethics Explainer."
- **Main File:**
  Handles Q&A, lessons, quizzes, and project guidance.
- **Data:**
  Uploads a PDF (like this textbook!) and links to other resources.
- **Behavior:**
  Friendly, clear, always checks for understanding.

Quick Quiz

1. **What are the main ingredients for a custom GPT?**
2. **Why is a manifest file important?**
3. **What might you include as extra data for a "history tutor" GPT?**
4. **What's the benefit of customizing the agent's personality?**

Hands-On Activity: Plan Your Own GPT

1. **Choose a topic or task** (e.g., language learning, recipe assistant).

2. **Write a one-sentence mission** (“This GPT helps users learn Spanish with quizzes and conversation.”)
3. **List 2–3 features or modules** you want your agent to have.
4. **Brainstorm what kind of data or resources** your agent should include.

Summary

- Building a custom GPT is like assembling a recipe: define the purpose, gather the parts, and test your creation.
- You can tailor GPTs for nearly any domain or audience.
- Practice building and testing to truly master the process!

Unit 6: Advanced Topics and Responsible AI

6.1 Bias, Safety, and Ethical Use

Bias in LLMs

- **What is bias?**
  LLMs are trained on huge text datasets from the internet, which can contain stereotypes, cultural assumptions, or outdated viewpoints. If not carefully managed, models may repeat or amplify these biases.
- **Example:**
  A model might generate stereotyped descriptions of certain professions or nationalities.
- **Mitigation:**
  Developers use filtering, balanced training data, and testing for fairness.

Safety Concerns

- **Misinformation:**
  LLMs can "hallucinate" (make up) facts that sound convincing.
- **Sensitive content:**
  LLMs might generate harmful, violent, or offensive language if prompted carelessly.
- **Misuse:**
  Bad actors could use LLMs for spam, scams, or to write malware.
- **Mitigation:**
  Developers add guardrails—such as content filters, moderation tools, and human oversight.

Ethical Use

- Always double-check important outputs, especially for factual or sensitive topics.
- Don't use LLMs to deceive, impersonate, or plagiarize.
- Disclose when content is AI-generated, especially in academic or professional settings.

6.2 Explainability and Transparency

Explainability means making it clear how and why an LLM produced a particular response.

- **Why is this important?**
  Users need to trust and understand AI systems—especially in education, healthcare, or law.
- **How can LLMs be made more transparent?**
  - Show reasoning steps ("chain-of-thought")
  - Cite sources or provide references
  - Let users inspect which parts of a prompt influenced the answer

6.3 Security Risks and Mitigations

Potential Risks:

- **Prompt Injection:**
  A user can try to trick the model into ignoring its instructions and outputting unintended text.
- **Data Leakage:**
  Sensitive info in training data could leak into responses.
- **Model Hacking:**
  Attackers might probe for vulnerabilities or attempt to "jailbreak" a model's safety rules.

Mitigations:

- Limit or sanitize user input.
- Use up-to-date models with robust safety mechanisms.
- Monitor usage and update safety policies regularly.

6.4 Keeping Up with LLM Advances

The field of LLMs changes fast!
Here are some tips for staying current:

- **Follow trusted sources:**
  Blogs (OpenAI, Anthropic, Google AI), academic preprint servers (arXiv), and news from reputable AI researchers.
- **Online courses and communities:**
  MOOCs (Coursera, edX), forums (Hugging Face, AI Stack Exchange), and GitHub repos.
- **Experiment:**
  Try out new models, tools, and datasets as they become available.

Quick Quiz

1. **What is one way bias can appear in LLM outputs?**
2. **Why is explainability important for AI in sensitive domains?**
3. **What is "prompt injection"?**
4. **List one source you can follow to keep up with LLM developments.**

Hands-On Activity: Spot the Pitfalls

1. Enter a prompt about a sensitive or controversial topic into an LLM.
2. Analyze the output: Is there bias? Is it factual? Is it safe for all audiences?
3. How would you improve the prompt or the LLM's instructions to avoid problems?

Summary

- Responsible AI use is essential—watch for bias, misinformation, and ethical concerns.
- Explainability and transparency build trust in AI systems.
- Stay curious and keep learning—LLMs evolve rapidly, and best practices change!

Unit 7: Hands-On Projects and Exercises

7.1 Guided Project: Your First Custom GPT

Project Goal:

Build a simple custom GPT tailored to a topic of your choice (e.g., travel advisor, coding tutor, history expert).

Step-by-Step:

1. **Choose your topic and audience.**
2. **Draft your GPT's "mission statement."**
3. **Outline your desired features/modules.**
   (Examples: Quiz generator, glossary, summarizer)
4. **Write a sample manifest file.**
5. **Plan or pseudocode your main logic.**
6. **List the types of data/resources your agent will use.**
7. **Test sample prompts and expected responses.**

Reflection:
What challenges did you face? What would you improve?

7.2 Prompt Engineering Challenge

Objective:
Refine weak prompts into effective, specific ones.

Exercise:

- Take 3 generic prompts (e.g., "Tell me about AI," "Explain history," "Help me with code").
- Rewrite each for clarity, audience, and format.
- Try both versions with a real LLM—compare responses.

- Reflect: What improved, and why?

7.3 LLM Benchmarking Exercise

Goal:
Practice evaluating LLM outputs.

Steps:

1. Pick a task (e.g., summarize a short news story, solve a math problem).
2. Run it through two different LLMs or GPT configurations if possible.
3. Use a simple rubric:
    - **Clarity:** 1–5
    - **Accuracy:** 1–5
    - **Relevance:** 1–5
4. Discuss or note which model performed better, and in what way.

7.4 Ethical Dilemma Scenarios

Goal:
Practice identifying and handling bias, safety, and ethical challenges.

Scenarios:

- A model generates an answer that contains subtle stereotypes.
  How do you address it?
- A user tries to get the GPT to write inappropriate or harmful content.
  How should the agent respond?
- A prompt requests private or sensitive information.
  What are best practices?

Activity:
For each, write how you'd adjust the prompt, agent instructions, or output to be safer and more ethical.

7.5 Extra Practice: Glossary, FAQ, and Troubleshooting

- **Build a mini-glossary** for your agent: 5–10 key LLM/GPT terms with clear definitions.
- **Write a short FAQ:** Anticipate 3–5 common user questions and prepare clear answers.
- **Identify a prompt that "breaks" your agent:** Try to fix it by updating your prompt engineering or instructions.

Summary and Next Steps

- Hands-on projects are the best way to solidify LLM and GPT skills.
- Continue to experiment—try new prompts, build new agents, and challenge yourself with real-world scenarios.
- Stay connected to the community and keep your knowledge up-to-date.

Course Completion

Congratulations!
You've learned:

- How LLMs and GPTs work
- The principles of prompt engineering
- How to evaluate, build, and use custom GPTs responsibly
- Advanced topics in safety, ethics, and explainability

What's Next?

- Build and share your custom GPTs with others.
- Help teach these skills in your community or classroom.
- Stay curious—AI is always evolving!

Appendices

Appendix A: LLM/GPT Glossary

Term	Definition
LLM	Large Language Model: an AI trained on vast amounts of text to generate, summarize, and answer questions in natural language.
GPT	Generative Pretrained Transformer: a family of LLMs developed by OpenAI,
Token	The smallest chunk of text LLMs process; could be a word, part of a word, or
Prompt	The text or instruction given to an LLM to generate a response.
Prompt Engineering	The practice of crafting inputs to maximize LLM performance and useful outputs.
Manifest	A configuration file that describes a custom GPT's features, modules, and
Parameters	The numerical values (weights) learned during training, which determine model
Context	The amount of text an LLM can "see" at one time when generating responses.
Bias	Systematic errors or prejudices in LLM outputs, often inherited from training
Hallucinatio	When an LLM generates plausible but false or invented information.
Chain-of-	A prompt pattern where the LLM is asked to "think out loud" and show its
Fine-Tuning	Training a pre-existing model further on specific data to specialize it.
RLHF	Reinforcement Learning from Human Feedback: technique for improving model
Explainabili	Making it clear how and why an LLM generated its answer.
Adversarial	Intentionally probing a model with tricky or hostile prompts to find weaknesses.

Appendix B: Further Reading and Resources

- **OpenAI Blog:**
  https://openai.com/blog
- **Hugging Face LLM Course:**
  https://huggingface.co/learn/nlp-course/chapter1/1
- **AI Alignment Fundamentals:**
  https://alignmentfundamentals.com/

- **Stanford CS25: Transformers United**
  https://web.stanford.edu/class/cs25/
- **arXiv Preprint Server (for latest AI research):**
  https://arxiv.org/

Appendix C: Sample Prompts and Templates

General Q&A

- "Explain how transformers work in simple terms."
- "Summarize this article for a high school audience: [paste article]."

Prompt Engineering Patterns

- **Instruction:**
  "List 5 ways teachers can use GPT models in the classroom."
- **Chain-of-Thought:**
  "Solve this math problem and show your reasoning step by step: 12 × 14."
- **Role Play:**
  "You are a professional career advisor. Give me 3 tips for my first job interview."
- **Few-Shot Example:**
  "Correct the grammar in these sentences.
  Example: 'She go to school.' → 'She goes to school.'
  Sentence: 'He eat breakfast every morning.'"

Appendix D: Troubleshooting and FAQ

Common Problems

- **The LLM makes up facts ("hallucinates"):**
  - Ask for sources.
  - Try rewording the prompt to specify only factual information.
- **The response is too vague or generic:**
  - Add more detail or constraints to your prompt.

- **The model won't answer a sensitive question:**
    - Rephrase the question, or review safety guidelines for your application.
- **I get an error or timeout:**
    - Try a shorter prompt, or check your internet/platform connection.

Frequently Asked Questions

Q: Can LLMs replace human experts?
A: No. LLMs are powerful assistants, but human judgment is essential—especially for important decisions.

Q: How do I make my GPT remember information across sessions?
A: Most sandboxes don't allow persistent memory for privacy and safety, but you can upload files or give context in each session.

Q: What's the best way to learn prompt engineering?
A: Practice! Try lots of prompts, compare results, and learn from examples in this book and online communities.

Q: How do I keep my GPT safe and ethical?
A: Use clear instructions, review outputs for bias or errors, and never use LLMs to deceive or harm.

End of Appendices