## 🧱 **Unit 1 – Foundations of Prompt Engineering**

**Current focus:**  
What is Prompt Engineering, history, elements of a well-designed prompt, open/closed prompts, clarity/specificity.

**Analysis:**  
This unit is solid and foundational, but it can be made more contemporary and conceptually deeper. Currently, it lacks grounding in *how* LLMs interpret tokens and context — which helps students truly understand why prompt phrasing matters.

**Recommendations:**

* 🔁 **Update:** Add a short section *“How LLMs interpret prompts (tokenization, attention, and context windows)”* — not mathematical, but conceptual (helps connect prompt design to LLM behavior).  
  *Justification:* Improves conceptual foundation and aligns with industry’s need for model-aware prompting.
* 🔼 **Add:** Introduce “Prompt Taxonomy” — zero-shot, few-shot, chain-of-thought, role prompting, system vs user prompts.  
  *Justification:* Students should know different prompting paradigms; this is now standard in prompt engineering courses (OpenAI, DeepLearning.AI).
* 🔁 **Update:** Merge “Elements of a well-designed prompt” and “Open vs. closed prompts” into a single lecture titled *“Prompt Clarity and Goal Framing”* for smoother flow.
* 🔽 **Remove:** “Historical background and evolution” (can be shortened to a brief 10-min context note).  
  *Justification:* History adds little practical value and consumes time that can be used for modern applications.

## 🧰 **Unit 2 – AI-Assisted Coding (GitHub Copilot & GPT-3)**

**Current focus:**  
Installing Copilot, using GPT-3, limitations, examples, text generation.

**Analysis:**  
This unit reflects 2022–2023 trends. In 2025, Copilot and GPT-4-Turbo/Claude 3.5 are standard. It would be better to make this *tool-agnostic* and conceptually focus on *AI-assisted problem-solving and reasoning via prompts.*

**Recommendations:**

* 🔁 **Update:** Rename to *“AI-Assisted Problem Solving and Coding”* — include Copilot, Replit Ghostwriter, and ChatGPT Code Interpreter.  
  *Justification:* Future-proofs the syllabus beyond a single vendor.
* 🔼 **Add:** “Prompting for debugging, code explanation, and refactoring.”  
  *Justification:* Teaches students to use LLMs responsibly in coding, not just for autocompletion.
* 🔽 **Remove:** “Installing and setting up Copilot” (too mechanical, and often outdated quickly). Replace with a lab demo.  
  *Justification:* Tool setup is not a learning outcome.
* 🔁 **Update:** Replace “What is GPT-3?” with “Understanding GPT-4/Claude-3 and transformer basics.”  
  *Justification:* Keeps the course current and gives conceptual grounding.

## 🌐 **Unit 3 – Exploring AI Tools (Bard, Bing, Vision, Whisper)**

**Current focus:**  
Overview of Bard, Bing, Google Vision, Whisper (voice-to-text), Falcon, ARC.

**Analysis:**  
While variety is good, this unit risks being too *tool-catalog-like* without deep learning value. Students should instead learn *how to evaluate and compare* tools for tasks.

**Recommendations:**

* 🔁 **Reframe:** “Evaluating AI Tools and Multimodal Models.” Focus on **text, image, and voice prompts** through examples (ChatGPT vision, Gemini, Whisper, Claude Opus).  
  *Justification:* Reflects multimodal trend and makes learning transferable.
* 🔼 **Add:** “Prompting for vision-language tasks (image captioning, diagram explanation)” using open tools.  
  *Justification:* Introduces multimodality simply — trending in 2025.
* 🔽 **Remove:** “Tencent ARC capabilities” and “Falcon” (too obscure for B.E.).  
  *Justification:* Low industry exposure, limited educational return.
* 🔁 **Update:** Replace “Bard and Bing” with “Gemini and Copilot Chat” — both represent 2024-2025 mainstream.
* 🔼 **Add (Ethical mini-module):** Responsible tool usage — misinformation, copyright, hallucinations.  
  *Justification:* Fulfills your “awareness of ethics and safety” outcome.

## 🧠 **Unit 4 – Advanced Prompting Concepts**

**Current focus:**  
Meta-prompting, chain-of-thought reasoning, prompt optimization, ReAct, chunking, delimiters, role prompting.

**Analysis:**  
Excellent inclusion — this is where students start thinking like engineers.  
However, the list mixes *advanced system concepts* (chunking, ReAct) with *basic prompt logic*. We should separate *core prompt reasoning* from *pipeline engineering*.

**Recommendations:**

* 🔁 **Restructure:**
  + Lecture 1–3: Meta-prompting, Chain-of-Thought (CoT), Step-by-Step reasoning.
  + Lecture 4–6: Role prompting, reasoning templates.
  + Lecture 7–9: Evaluation and optimization.
* 🔼 **Add:** “Prompt Evaluation Frameworks” — introduce BLEU/ROUGE intuitively, or use OpenAI’s Evals for demonstration.  
  *Justification:* Directly connects to your CO-4 “Evaluate and improve prompts.”
* 🔽 **Remove:** “Overcoming token limits and chunking” — move to optional reading.  
  *Justification:* Too advanced (memory management is graduate-level concept).
* 🔼 **Add:** “Prompt patterns” — e.g., Persona, Few-shot, Reframing, Debate prompts.  
  *Justification:* Practical, creative, and widely used in industry.

## 🧭 **Unit 5 – Ethics, Feedback, and Continuous Improvement**

**Current focus:**  
Ethical considerations, evaluation, feedback incorporation.

**Analysis:**  
Good closing unit, but could be richer. Students should connect prompting to *communication design*, *bias handling*, and *evaluation loops*.

**Recommendations:**

* 🔼 **Add:** “Human-in-the-loop prompt refinement.”  
  *Justification:* Connects ethical awareness to real-world feedback cycles.
* 🔼 **Add:** “Prompt safety and hallucination management.”  
  *Justification:* Industry expects awareness of model unreliability.
* 🔁 **Update:** Rename unit to *“Ethics, Evaluation, and Continuous Prompt Refinement.”*
* 🔼 **Add (Mini project):** Design a prompt-based application (e.g., chatbot tutor or idea generator) evaluated by clarity, safety, and creativity.  
  *Justification:* Concludes the course with hands-on synthesis — aligns with your “design and prototype” outcome.

## 🎓 **Additional Global Recommendations**

| Category | Recommendation | Justification |
| --- | --- | --- |
| 🔄 **Integration** | Introduce one **“Prompt Lab Notebook”** per student — where they log experiments, results, and improvements. | Reinforces lifelong learning and reflection. |
| 🧩 **Tools** | Use open/free platforms (ChatGPT free tier, Gemini, HuggingFace) instead of locked enterprise tools. | Ensures accessibility and durability. |
| 🧭 **Industry Readiness** | Include 1–2 lectures on “Careers and Tools in Generative AI Ecosystem” | Bridges academic to career relevance. |
| ⚖️ **Assessment** | Reduce Level 1–2 (Remember/Understand) weighting from 50% → 40%, increase Level 3–4 (Apply/Analyze) to 50–60%. | Promotes hands-on skill-building. |
| 🧱 **Pedagogical Balance** | Use case-based labs (creative writing, summarization, code generation, data analysis). | Encourages applied learning. |

## 🏁 **Summary of Major Changes**

| Type | Theme | Change |
| --- | --- | --- |
| 🔁 Update | Unit 2 | Replace GPT-3 focus with GPT-4, Claude, Gemini |
| 🔼 Add | Unit 1 | How LLMs interpret prompts; prompt taxonomy |
| 🔼 Add | Unit 3 | Multimodal prompting (vision, voice) |
| 🔼 Add | Unit 4 | Prompt patterns, evaluation methods |
| 🔼 Add | Unit 5 | Human-in-loop & hallucination management |
| 🔽 Remove | Unit 2 | Tool installation steps |
| 🔽 Remove | Unit 3 | Obscure tools (Falcon, ARC) |
| 🔽 Remove | Unit 4 | Token/chunking section (too advanced) |