Few-Shot Encoding, **Prompt Chaining**, and **Chain of Thought Prompting**

Aspect	**Few-Shot Encoding**	**Prompt Chaining**	**Chain of Thought Prompting**
Definition	A technique where a prompt includes a few examples of inputs and outputs to guide the AI.	A method where tasks are broken into smaller steps, and the output of one prompt feeds into the next.	A prompting strategy that encourages the model to generate step-by-step reasoning to solve a problem.
Focus	Providing examples to establish a pattern for the AI to follow.	Breaking complex workflows into sequential sub-tasks.	Explicit reasoning to solve a single task or question.
Goal	To improve task understanding by giving contextual examples.	To complete complex, multi-step workflows or tasks modularly.	To enhance reasoning and accuracy for solving intricate tasks.
Structure	Examples are embedded in a single prompt.	Multiple prompts are used in a sequence, with interdependencies.	A single prompt, but the response involves reasoning steps.
Use Case	Pattern recognition tasks like classification, summarization, or translation.	Multi-stage workflows like data pipelines, content generation, or process automation.	Tasks involving logical reasoning, arithmetic, or multi-step problem-solving.
Complexity Handling	Handles tasks directly by teaching via examples.	Decomposes the complexity by isolating subtasks.	Handles complexity by explicitly reasoning step by step.
Key Feature	Few-shot examples act as "training data" within the prompt.	Sequential prompts allow modular processing.	Model explains its thought process in intermediate steps.
Output Dependency	Independent – the output is generated directly from the examples provided.	Dependent – output from one step serves as input for the next.	Output includes reasoning and final answer for the same task.
Example	Classifying emails as spam or not: "Email 1: Spam Email 2: Not Spam Email 3: ?"	1. Extract product features. 2. Summarize features. 3. Generate product description.	Solving: "What is 45 divided by 9?" "Step 1: Divide 45 by 9. Step 2: 45 ÷ 9 = 5. Final answer: 5."
Framework Compatibility	Simple to use in most LLMs without additional tooling.	Often implemented with frameworks like LangChain.	Naturally fits in LLM prompts, but also adaptable to frameworks.

Transparency	Low – reasoning is implicit.	Medium – reasoning may emerge across chained prompts.	High – explicit reasoning is integral to the output.
Ideal For	Tasks with clear examples or patterns.	Multi-step workflows with dependencies between stages.	Single tasks that benefit from detailed, step-by-step thinking.

Summary

- **Few-Shot Encoding** is best for teaching patterns through examples.
- **Prompt Chaining** handles complex workflows by modularizing tasks.
- **Chain of Thought Prompting** enhances logical reasoning by explaining intermediate steps explicitly.

Foundation Models and **External Models**

Aspect	**Foundation Model**	**External Model**
Definition	A large pre-trained AI model designed to be versatile and adaptable to a wide range of tasks.	A specialized model designed for specific tasks, often integrated into or used alongside other systems.
Scope	General-purpose and multi-domain.	Narrow, task-specific, or domain-specific.
Training Data	Trained on massive datasets spanning diverse domains (e.g., text, images, code).	Trained on smaller, specialized datasets tailored to specific tasks or domains.
Examples	GPT (OpenAI), BERT, DALL·E, Stable Diffusion.	A fraud detection model for banking, a recommendation system for e-commerce.
Capabilities	Broad; capable of tasks like text generation, translation, summarization, and more.	Limited to solving a specific problem or set of problems.
Customizability	Requires fine-tuning or prompt engineering for domain-specific tasks.	Often pre-configured for its intended task; may allow parameter adjustment.
Deployment	Used as a base for multiple applications or adapted for specific use cases.	Deployed directly for a specific application or problem.
Use Cases	General NLP tasks, image synthesis, multi-modal tasks, large-scale analytics.	Predictive maintenance, credit scoring, inventory optimization.

Flexibility	Highly flexible; can be adapted to various tasks with minimal changes.	Rigid; designed to solve specific problems with high precision.
Dependencies	Self-contained, providing broad functionality.	Often relies on external data sources or integration with broader systems.
Development Effort	High; requires extensive computational resources and data for training.	Moderate; tailored to specific needs, often built with domain expertise.
Performance	May require fine-tuning for optimal performance in specialized tasks.	Optimized for high performance in the targeted task or domain.
Scalability	Scales well across domains and tasks; resource-intensive.	Limited scalability outside its intended domain or task.
Cost	Expensive to train and deploy due to large-scale infrastructure.	Relatively cost-effective for task-specific applications.

Key Differences

- **Foundation Models** are broad, adaptable, and designed to serve as a base for many applications, while **External Models** are specialized and optimized for specific use cases.
- **Foundation Models** require fine-tuning or context-specific prompts to perform specialized tasks, whereas **External Models** are ready to solve a pre-defined problem out of the box.

Guided Agents and **Unguided Agents**

Aspect	**Guided Agents**	**Unguided Agents**
Definition	Agents that operate with explicit instructions, constraints, or predefined workflows.	Agents that operate autonomously, relying solely on their internal logic and environment exploration.
Level of Autonomy	Limited; their behavior is shaped and restricted by specific guidance or rules.	High; they make decisions independently without explicit external constraints.
Decision-Making	Deterministic or semi-deterministic based on pre-set instructions or defined boundaries.	Stochastic or dynamic, adapting based on their understanding of the environment.
Dependency on Guidance	Heavily dependent on user input, workflows, or constraints to complete tasks.	Operate independently, often learning from interactions or exploration.
Complexity of Tasks	Well-suited for structured or predefined tasks where rules and objectives are clear.	Suitable for open-ended or exploratory tasks where outcomes are not predefined.
Examples	- A customer support chatbot scripted to follow a specific	- Autonomous exploration agents in video games.

	decision tree. - Workflow automation bots that execute predefined processes.	- AI agents solving puzzles or conducting research without explicit step-by-step instructions.
Learning Style	Typically does not involve self-learning; follows provided guidelines.	Often involves self-learning or reinforcement learning to improve performance.
Error Handling	Handles errors within defined boundaries; may fail if a scenario falls outside guidance.	Adapts to unexpected situations by making decisions based on learned knowledge or context.
Use Cases	- Process automation (e.g., invoice processing, form filling). - Decision-support systems with clear objectives.	- Autonomous driving. - Research assistants exploring open-ended problems or generating creative outputs.
Flexibility	Limited; rigidly adheres to rules or predefined logic.	Highly flexible; adapts based on the situation and environment.

Summary:

- **Guided Agents**: Operate within a framework of predefined rules or instructions. They excel in predictable environments where objectives are clear and tasks require minimal deviation from set paths.
- **Unguided Agents**: Function autonomously, exploring possibilities and making decisions dynamically. They are ideal for complex, open-ended, or unpredictable environments where predefined rules may not suffice.

For details about the Generative AI workshop, visit:
https://www.nitinkapse.com/generative-ai

Register here to secure your spot:
https://forms.gle/PrzkmvYh5yvEWUKZ6