Module 1 Summary: Explore GenAI Universe

1. Introduction to Language Models (LMs)

Language Models (LMs) are AI systems trained to understand and generate human language. They are broadly classified based on:

Size: Small Language Models (SLMs) vs. Large Language Models (LLMs)

Modality: Unimodal (text) vs. Multimodal (text, image, audio)

Access Type: Open-source vs. Proprietary

2. LLMs vs. SLMs

Feature	LLMs (e.g., GPT-4, Claude 3)	SLMs
Parameters	Billions to Hundreds of Billions	Millions to Low Billions
Tasks	Multi-step reasoning, complex math problems	Simple tasks
Advantages	Performs well for complex tasks	Efficient and low resource usage
Limitations	High resource, time, and training cost	Cannot handle complex reasoning

3. Multimodal Capabilities

Multimodal LMs can handle inputs/outputs in different formats (text, images, etc.)

Examples: GPT-4-Vision, Gemini 1.5 Flash, Claude 3.5 Sonnet

Unimodal LMs deal with only text inputs (e.g., GPT-3.5, LLaMA 2.1)

4. Open Source vs. Proprietary Models

Open Source	Proprietary
Public architecture	Privately controlled
Promotes innovation & collaboration	Restricted customization
Examples: Meta's LLaMA 3, Mistral 7B	GPT-4, Claude 3, Gemini, Anthropic

5. Transformer Architecture

Transformers are the foundational architecture behind LLMs, known for:

Handling long-term dependencies in text

Enabling efficient parallel processing

Components:

Multi-head attention

Add & Norm layers

Feed-forward networks

Linear + Softmax layers for output

6. Fine-Tuning Challenges

Data availability: Lack of quality, domain-specific data

Resource intensive: Requires expensive hardware and high computing power

Bias amplification: Can replicate societal biases

Limited use-case generalization

7. Retrieval-Augmented Generation (RAG)

Combines information retrieval with generative capabilities of LLMs

Workflow:

1. Query → Retrieval from database/web/files

2. Documents + Query → Passed to LLM

3. LLM generates a response

Example Tool: Google NotebookLM (acts as a RAG engine)

8. Prompt Engineering

Crafting inputs effectively to get desired outputs from LLMs.

Key Characteristics:

Clarity

Specificity

Meaningful context

Goal orientation

9. Prompting Strategies

Strategy	Description
Zero-shot	No examples provided
Few-shot	Few examples included
Chain-of-thought	Step-by-step explanation
ReAct	Reasoning + action-based prompting
Self-consistency	Multiple answers → best selected

10. Types of Prompt Engineering

Used across domains:

Code generation: Meta's Code Llama

Finance: BloombergGPT

Medical/Legal: Google Med-PaLM

11. Quizzes & Assignments

Covered core concepts and use cases

Quizzes tested understanding of model types, prompting, architecture, and strategies

Shared reference: ChatGPT Quiz Answers

Assignment: Appears to involve applying prompting techniques in real-world scenarios (request help if unclear)