AWS Certified AI Practitioner (AIF-C01)

Domain 2: Fundamentals of Generative AI

Task Statement 2.1: Explain the Basic Concepts of Generative AI

OBJECTIVES:

1. Understand foundational generative AI concepts
2. Identify potential use cases for generative AI models
3. Describe the foundation model lifecycle

---

1. FOUNDATIONAL GENERATIVE AI CONCEPTS

These are the building blocks of modern generative AI systems.

Tokens

- **Definition**: Basic units of input/output for a model (words, subwords, or characters).
- **Example**:
  - The sentence `"Cats are great!"` might be tokenized into `["Cats", " are", " great", "!"]`
- **Why It Matters**: Model input/output size is measured in tokens (not words or characters).

Chunking

- **Definition**: Dividing large documents into manageable parts (chunks).
- **Use Case**: For models with token limits (e.g., summarizing large PDFs), documents are chunked.
- **Application**: Core step in Retrieval-Augmented Generation (RAG).

Embeddings

- **Definition**: Numerical representations of text or data capturing semantic meaning.
- **Used For**: Semantic search, clustering, similarity comparison.
- **AWS Example**: Amazon Titan Embeddings on Amazon Bedrock.

Vectors

- **Definition**: Arrays of numbers used to represent embeddings in vector space.
- **Use**: Power vector databases for similarity search.
- **Example**: “Dog” and “puppy” have similar vector representations.

◈ Prompt Engineering

- **Definition**: Designing effective inputs to elicit desired outputs from an LLM.
- **Types**:
  - Zero-shot: No examples
  - Few-shot: With examples
  - Chain-of-thought: Step-by-step reasoning
- **Best Practice**: Be explicit, use role-based prompts, control formatting.

◈ Transformer-Based LLMs

- **Definition**: Large models using self-attention for parallel sequence processing.
- **Examples**: GPT, BERT, Claude, LLaMA
- **Advantage**: Handle long-range dependencies and scale effectively.

◈ Foundation Models (FMs)

- **Definition**: Pre-trained on massive datasets and adaptable to many tasks (via fine-tuning or prompting).
- **Features**:
  - Multi-domain
  - General-purpose
- **AWS**: Access via Amazon Bedrock (Anthropic Claude, Meta Llama 2, etc.)

◈ Multi-Modal Models

- **Definition**: Models that understand/generate across modalities (text, images, audio, etc.)
- **Examples**:
  - GPT-4 (text + image)
  - DALL·E (text-to-image)
  - Gemini (text + code + image + audio)
- **Use Case**: Describe images, generate audio, caption videos.

◈ Diffusion Models

- **Definition**: Models trained to reverse a noise process and generate data.
- **Use Case**: Image/video/audio generation.

- **Example**: Stable Diffusion, DALL·E
- **How It Works**:
    1. Add noise to images during training
    2. Learn to remove noise step-by-step during generation

🚀 2. USE CASES FOR GENERATIVE AI

Generative AI powers a wide range of practical applications.

Use Case	Description	AWS Services
Image Generation	Generate images from text or modify images	Amazon Bedrock + Stability AI, DALL·E
Video Generation	Create synthetic video content	Third-party integrations + Bedrock
Audio Generation	Text-to-speech and music generation	Amazon Polly
Summarization	Condense long documents into key points	Amazon Bedrock, Comprehend
Chatbots	AI-powered conversational agents	Amazon Lex, Amazon Q, Bedrock
Translation	Convert text between languages	Amazon Translate
Code Generation	Generate or refactor code snippets	Amazon CodeWhisperer, Bedrock
Customer Support	Automate help desk interactions	Amazon Connect + Q, Lex, Bedrock
Semantic Search	Retrieve relevant content using meaning, not keywords	Amazon Kendra, OpenSearch, Bedrock
Recommendation Engines	Suggest content/products based on user data and embeddings	Amazon Personalize, Titan Embeddings

---

🔄 3. FOUNDATION MODEL LIFECYCLE

The lifecycle includes all stages from data to deployed generative AI systems.

---

1. **Data Selection**

- **Goal**: Collect high-quality, diverse, and relevant datasets.
- **Types**: Text, code, audio, video, images
- **Key Concepts**:

- Remove noise
- Balance representation
- Prevent bias

---

2. **Model Selection**

- **Options**:
  - Use an existing FM (e.g., Claude, LLaMA)
  - Train a model from scratch (costly)
- **Factors**:
  - Task (e.g., summarization, translation)
  - Modality (text, image, audio)
  - Model size vs cost

---

3. **Pre-Training**

- **Purpose**: Learn general knowledge from large unstructured datasets.
- **Requires**: Enormous compute (e.g., GPU clusters)
- **Duration**: Weeks to months
- **Output**: Foundation model

---

4. **Fine-Tuning**

- **Purpose**: Specialize FM to domain-specific tasks.
- **Techniques**:
  - Full fine-tuning
  - Parameter-efficient tuning (LoRA, adapters)
- **Example**: Customize LLM to legal or healthcare documents.

---

5. **Evaluation**

- **Metrics**:
  - **Performance**: Accuracy, BLEU, ROUGE, F1
  - **Robustness**: Out-of-distribution handling
  - **Bias/Fairness**: Safety and inclusivity
- **Methods**: Test sets, human evaluation, adversarial tests

---

6. **Deployment**

- **Options**:

  - Fully managed (Amazon Bedrock)
  - Custom hosting (Amazon SageMaker, ECS, EKS)

- **Key Factors**:
  - Latency
  - Scaling
  - Cost

---

7. **Feedback Loop**

- **Goal**: Improve model performance post-deployment.
- **Mechanisms**:
  - Human feedback (e.g., thumbs up/down)
  - RLHF (Reinforcement Learning from Human Feedback)
  - Monitoring user interactions

🛠️ Related AWS Services

Stage	AWS Tools
Data Collection	Amazon S3, AWS Data Exchange
Model Hosting	Amazon Bedrock, Amazon SageMaker, ECS, EKS
Fine-Tuning	Amazon SageMaker (JumpStart, Model Training Jobs)
Evaluation & Testing	SageMaker Model Monitor, CloudWatch
Prompt-Based Use	Amazon Bedrock, Amazon Q, Lex, Polly
Retrieval + RAG	Amazon Kendra, OpenSearch with vector search

✅ Study Tips

- **Understand concepts with real-world examples**: e.g., “Use embeddings in a movie recommendation engine.”
- **Practice prompt engineering**: Try prompts using Amazon Bedrock or similar tools.
- **Review AWS service capabilities**: Especially those that support GenAI (Bedrock, SageMaker, Lex).
- **Create diagrams**: Lifecycle and architecture visuals help with recall.