If you know all these, then you know most things in GenAl

Prompt Engineering

Prompt Techniques

- 1. Chain of Thought (CoT)
- 2. Few-Shot Chain of Thought (Few-Shot-

CoT)

- 3. ReAct (Reasoning and Acting)
- 4. Tree of Thoughts (ToT)
- 5. Self-Consistency
- 6. Hypothetical Document Embeddings (HyDE)
- 7. Least-to-Most Prompting
- 8. Graph Prompting
- 9. Recursive Prompting
- 10. Generated Knowledge
- 11. Automatic Reasoning and Tool-Use (ART)
- 12. Automatic Prompt Engineer (APE)

Transformer Architecture

- 1. Self-Attention Mechanism
- 2. Positional Encoding
- 3. Multi-Head Attention:
- 4. Encoder-Decoder Architecture
- 5. Layer Normalization and Feed-

Forward Layers

- 6. Pre-training and Fine-tuning
- 7. Scalability with Parallelization
- 8. Applications Across Domains

Cloud Support

Azure - Azure OpenAl, Azure Al Studio, Azure Machine Learning

AWS - Amazon Bedrock,

Amazon Sagemaker, Amazon Q

Google - Vertex Al

LLMOPs

- 1. Model Deployment and Scaling
- 2. Monitoring and Logging
- 3. Versioning and Model Lifecycle
- Management
- 4. Cost Optimization
- 5. Feedback Loops and CI
- 6. Compliance and Security
- 7. Latency and Throughput
- Optimization 8. A/B Testing

RAG

Vector DB - Pinecone, Weaviate, Qdrant,

Chroma, Milvus, Vespa, LanceDB

Embedding Models - OpenAl's

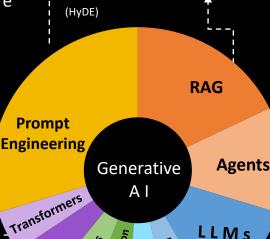
Embedding, Google's text-embedding-

004, Huggingface Opensource

Embedding Models

RAG Techniques

- 1. Basic RAG
- 2. Re-ranking RAG
- 3. Hybrid Search RAG
- 4. Multi-Index RAG
- 5. Query Expansion RAG
- 6. Adaptive RAG
- 7. Corrective RAG
- 8. Self Adaptive RAG
- 9. Hypothetical Document Embedding



ΑI

LLMs

Multimodal

Models etc.

Agents

industries.

LangGraph

Manages

Determines Tool

Usage

Coordinates

Provide

Results

Tools

LLM agents use large language models

to handle complex queries, combining

Multi-Agentic Frameworks - CrewAl,

Inform

Decisions

Memory

Utilises

Retrieves/Stores

@genieincodebottle

User

Reque

Agent

Planning

LLMs

Closed Source - OpenAl o1, GPT-

4o,GPT-4o Mini, Gemini 1.5/1 Pro,

Gemini 1.5 Flash-8B, Claude 3.5

DeepSeek-R1, DeepSeek distilled

models, Meta's Llama 3.2 1B, & 3B,

Llama 3.1 405B, Microsoft's Phi3.5-

Mini, Google's Gemma 2, Ministral 3B

& 8B, Huggingface based Opensource

Sonnet, Claude 3.5 Haiku etc.

Open LLMs - DeepSeek-v3,

general and specialized knowledge,

making them valuable across

Closed Source - GPT-4o, Gemini 1.5/1 Pro, Gemini 1.5 Flash-8B, Claude 3.5 Sonnet etc Open Source - Meta's Llama 3.2 11B, 90B,

Google's PaliGemma, LLaVA-1.5, Pixtral 12B,

QwenVL, DeepSeek Janus-Pro 7B & 1 B,

Huggingface based MultiModal

LLM Fine Tuning

- 1. Data Selection and Preprocessing
- 2. Transfer Learning and Domain Adaptation
- 3. Parameter-Efficient Fine-Tuning
- 4. Prompt Tuning and Instruction Fine-Tuning
- 5. Evaluation Metrics for Fine-Tuned Models
- 6. Safety and Bias Mitigation in Fine-Tuning
- 7. Hyperparameter Optimization

LLM Evaluation 1. BLEU (Bilingual Evaluation Understudy) 2. ROUGE (Recall-Oriented Understudy for Gisting Evaluation) 3. Perplexity 4. Exact Match (EM) Human Evaluation Scores

6. Bias and Fairness Metrics

LLM Frameworks (Across Layers) - LangChain/LlamaIndex

7. Toxicity Scores

If you know all these, then you know most things in GenAl

Vector DB - Pinecone, Weaviate, Qdrant,

Embedding, Google's text-embedding-

Chroma, Milvus, Vespa, LanceDB

Embedding Models - OpenAl's

004, Huggingface Opensource

Embedding Models

RAG Techniques

2. Re-ranking RAG

3. Hybrid Search RAG

5. Query Expansion RAG

4. Multi-Index RAG

6. Adaptive RAG

7. Corrective RAG

(HyDE)

8. Self Adaptive RAG

1. Basic RAG

RAG

Prompt Engineering

Prompt Techniques

- 1. Chain of Thought (CoT)
- 2. Few-Shot Chain of Thought (Few-Shot-

CoT)

- 3. ReAct (Reasoning and Acting)
- 4. Tree of Thoughts (ToT)
- Self-Consistency
- 6. Hypothetical Document Embeddings

(HyDE)

- 7. Least-to-Most Prompting
- 8. Graph Prompting
- 9. Recursive Prompting
- 10. Generated Knowledge
- 11. Automatic Reasoning and Tool-Use (ART)
- 12. Automatic Prompt Engineer (APE)

Transformer Architecture

- 1. Self-Attention Mechanism
- 2. Positional Encoding
- 3. Multi-Head Attention:
- 4. Encoder-Decoder Architecture
- 5. Layer Normalization and Feed-

Forward Layers

- 6. Pre-training and Fine-tuning
- 7. Scalability with Parallelization
- 8. Applications Across Domains

Prompt

LLM Evaluation

2. ROUGE (Recall-Oriented Understudy

1. BLEU (Bilingual Evaluation

for Gisting Evaluation)

4. Exact Match (EM)

7. Toxicity Scores

Human Evaluation Scores

6. Bias and Fairness Metrics

Understudy)

3. Perplexity

Engineering Generative Transformers

ΑI

9. Hypothetical Document Embedding

Agents

RAG

LLMs

Agents

LLM agents use large language models to handle complex queries, combining general and specialized knowledge, making them valuable across industries.

Multi-Agentic Frameworks - CrewAl,

LangGraph User Request Provide Inform Results Decisions Agent Core Manages Utilises Coordinates Planning Tools Memor **Determines Tool** Retrieves/Stores Usage

@genieincodebottle

LLMs

Closed Source - OpenAI o1, GPT-4o, GPT-4o Mini, Gemini 1.5/1 Pro, Gemini 1.5 Flash-8B, Claude 3.5 Sonnet, Claude 3.5 Haiku etc.

Open LLMs - DeepSeek-v3, DeepSeek-R1, DeepSeek distilled models, Meta's Llama 3.2 1B, & 3B, Llama 3.1 405B, Microsoft's Phi3.5-

Mini, Google's Gemma 2, Ministral 3B & 8B, Huggingface based Opensource

Models etc.

Multimodal

Closed Source - GPT-4o, Gemini 1.5/1 Pro, Gemini 1.5 Flash-8B, Claude 3.5 Sonnet etc Open Source - Meta's Llama 3.2 11B, 90B, Google's PaliGemma, LLaVA-1.5, Pixtral 12B, QwenVL, DeepSeek Janus-Pro 7B & 1 B,

Huggingface based MultiModal

LLM Fine Tuning

- 1. Data Selection and Preprocessing
- 2. Transfer Learning and Domain Adaptation
- 3. Parameter-Efficient Fine-Tuning
- 4. Prompt Tuning and Instruction Fine-Tuning
- 5. Evaluation Metrics for Fine-Tuned Models
- 6. Safety and Bias Mitigation in Fine-Tuning
- 7. Hyperparameter Optimization

Cloud Support

Azure - Azure OpenAI, Azure AI Studio, Azure Machine Learning

AWS - Amazon Bedrock,

Amazon Sagemaker, Amazon Q

Google - Vertex Al

LLMOPs

- 1. Model Deployment and Scaling
- 2. Monitoring and Logging
- 3. Versioning and Model Lifecycle

Management

- 4. Cost Optimization
- 5. Feedback Loops and CI
- 6. Compliance and Security
- 7. Latency and Throughput
- Optimization
- 8. A/B Testing

LLM Frameworks (Across Layers) - LangChain/LlamaIndex