Key Concepts in Modern AI Applications

Prompt

A **prompt** is the instruction, question, or text you give to a large language model (LLM) to generate a response. It serves as the starting point of interaction between humans and AI. The quality and clarity of a prompt strongly influence the output: a well-crafted prompt can produce accurate and relevant answers, while a vague prompt may lead to irrelevant or confusing results. Prompts are the primary way we communicate with AI models to achieve specific tasks, such as summarizing text, answering questions, or generating content.

Related Topics:

- **Prompt Engineering:** The practice of designing effective prompts to get the best results from an AI model.
- **Few-shot / Zero-shot prompting:** Techniques where the model is shown a few examples (few-shot) or no examples (zero-shot) to perform a task.

Large Language Model (LLM)

A **Large Language Model** is an AI system trained on vast amounts of text to understand and generate human-like language. LLMs can perform tasks such as translation, summarization, question-answering, and text generation. They do not inherently "know" real-time data or specific private knowledge but can provide responses based on patterns learned during training. Examples include GPT-4, Claude, and LLaMA. LLMs are the core engine behind AI applications that process natural language.

Related Topics:

- **Transformer Architecture:** The deep learning structure behind most LLMs that allows understanding context and relationships in text.
- **Tokenization:** How text is split into smaller pieces (tokens) for the model to process efficiently.

Agent

An **agent** in AI is a system that uses an LLM to decide what actions to take in order to complete a task. Instead of only generating text, an agent can choose to perform actions like searching a database, calling an API, or running a calculator tool. Agents help automate complex workflows by combining reasoning, tool usage, and AI output. In essence, agents act like intelligent assistants that figure out **what to do next** rather than only responding to prompts.

Related Topics:

- **Tool Use in AI:** Integration of external APIs or functions that an agent can call to complete tasks.
- **Multi-step Reasoning:** Agents can plan sequences of steps to solve complex problems instead of single-turn answers.

Embedding

An **embedding** is a numeric representation of text, words, or documents that captures their meaning in a format a computer can process. Embeddings allow AI systems to compare texts based on meaning rather than exact words. For example, the words “cat” and “kitten” would have similar embeddings because they have similar meanings. Embeddings are essential for search, recommendation systems, and retrieval tasks where semantic understanding is required.

Related Topics:

- **Vector Search / Vector Database:** Databases optimized to search embeddings based on similarity, such as Pinecone, FAISS, or Milvus.
- **Semantic Search:** Searching for information based on meaning rather than exact keyword matches.

Fine-tuning

Fine-tuning is the process of adapting a pre-trained LLM to a specific task or dataset. While LLMs are trained on general text data, fine-tuning customizes the model to understand domain-specific language or provide more accurate responses for a particular application. Fine-tuning allows AI developers to make models more relevant, accurate, and effective for specialized use cases, such as legal documents, medical records, or company-specific knowledge.

Related Topics:

- **LoRA (Low-Rank Adaptation):** A method for efficient fine-tuning that updates only a small part of the model.
- **Custom Knowledge Models:** Fine-tuned models that incorporate a company’s or domain-specific data for better accuracy.

Retrieval-Augmented Generation (RAG)

RAG is a method that combines retrieval of external information with text generation by an LLM. Instead of relying solely on the model’s pre-trained knowledge, RAG retrieves relevant documents or data from a database and provides them as context for the LLM to generate more accurate and informed responses. This approach is especially useful when answering questions about private or recent data that the LLM would not otherwise know.

Related Topics:

- **Vector Store + Retriever:** The combination of embedding storage and retrieval mechanism that powers RAG.
- **Context Window:** The limit of how much information the LLM can process at once, which RAG helps optimize.