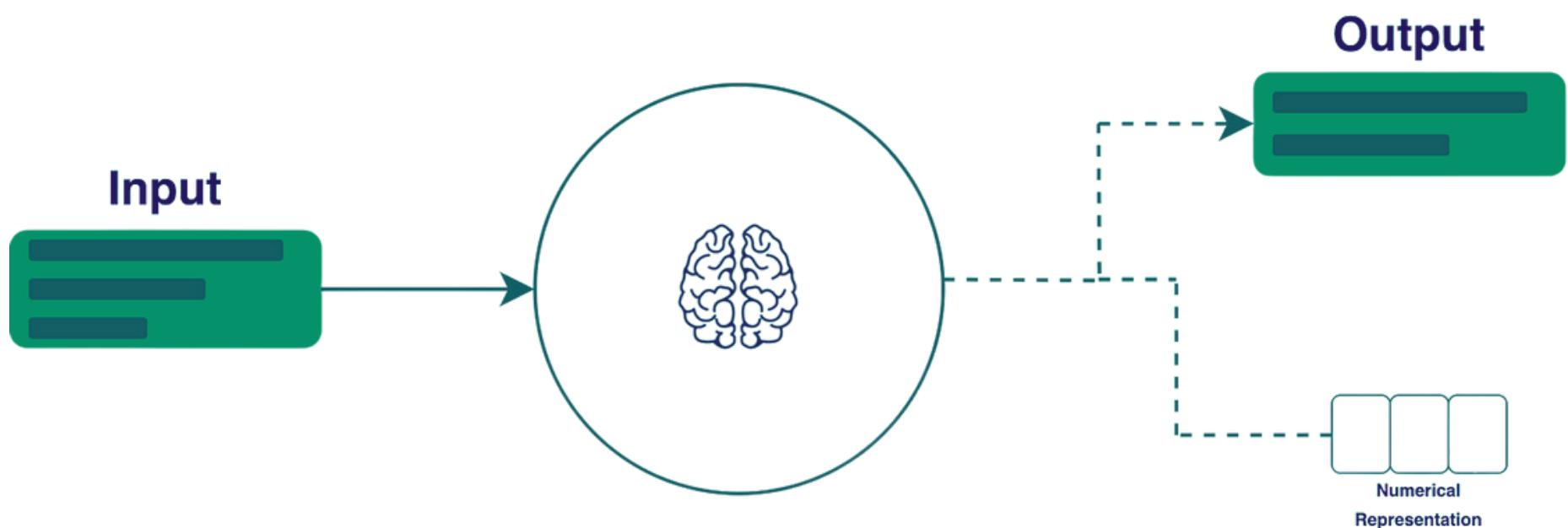


MASTERING LLM'S AND GENERATIVE AI





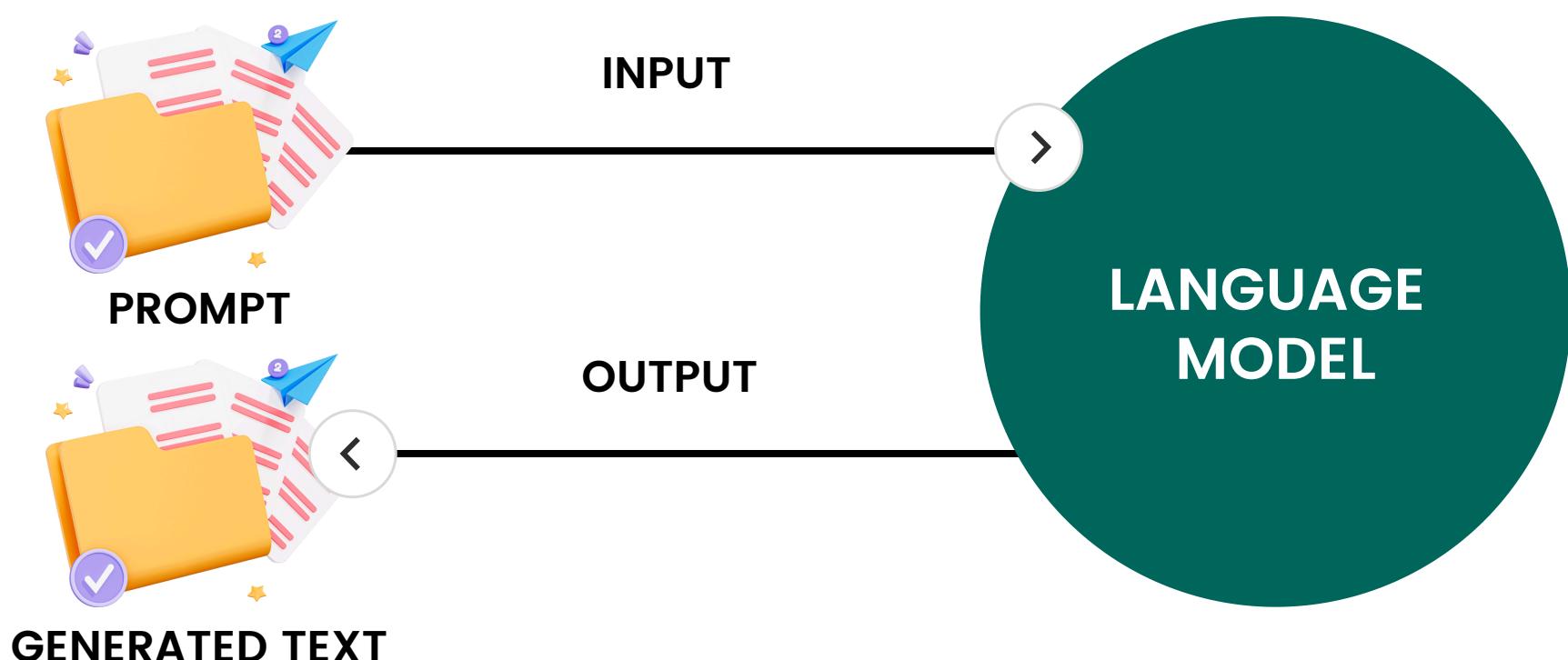
LLM Basics



- **Basics:** Understand key terms, uses, issues, and frameworks.
- **Data:** Know the training data and potential biases.
- **Scale:** Be aware of LLM size and training costs.
- **Training:** Understand differences between LLM and ML training.
- **Purpose:** Define clear objectives (chatbot, Q&A, image generator).



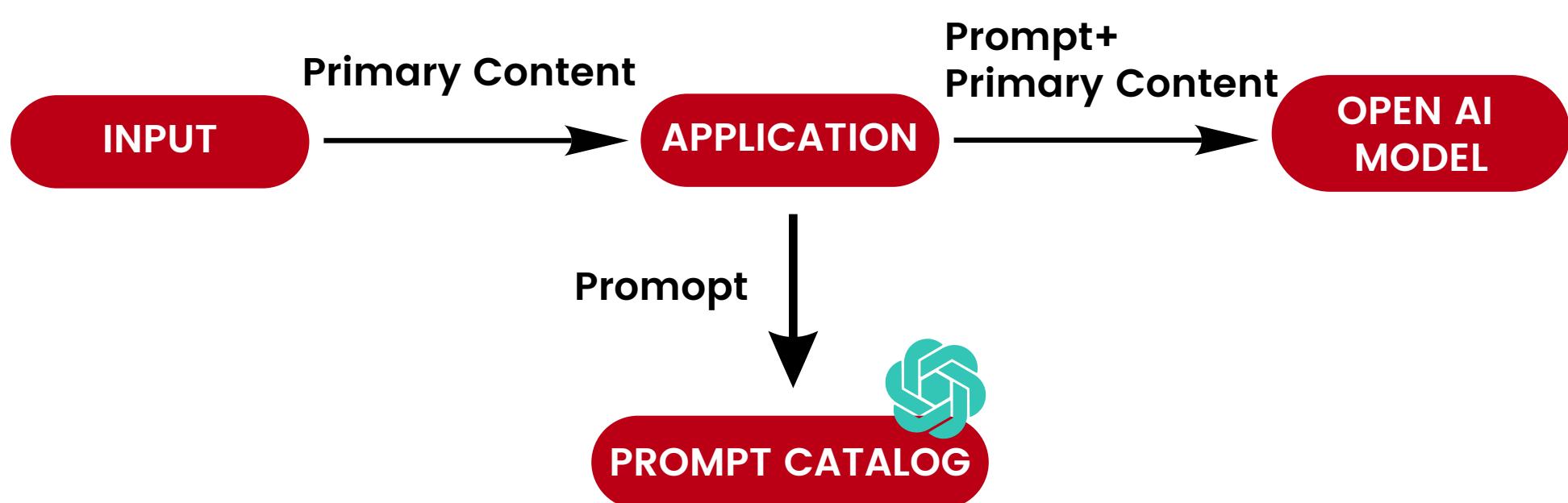
Prompt Engineering



- **What is it?** Designing precise inputs for LLMs to get specific results.
- **Example:** Instead of "Create a Twitter post," use "Create a snappy Twitter post for millennials."
- **Fine-tuning:** Adjust prompts to achieve the exact output you need.
- **Growing Field:** Prompt engineering has become a specialized job.
- **The Art of AI:** It's all about mastering the nuances of communication with AI.



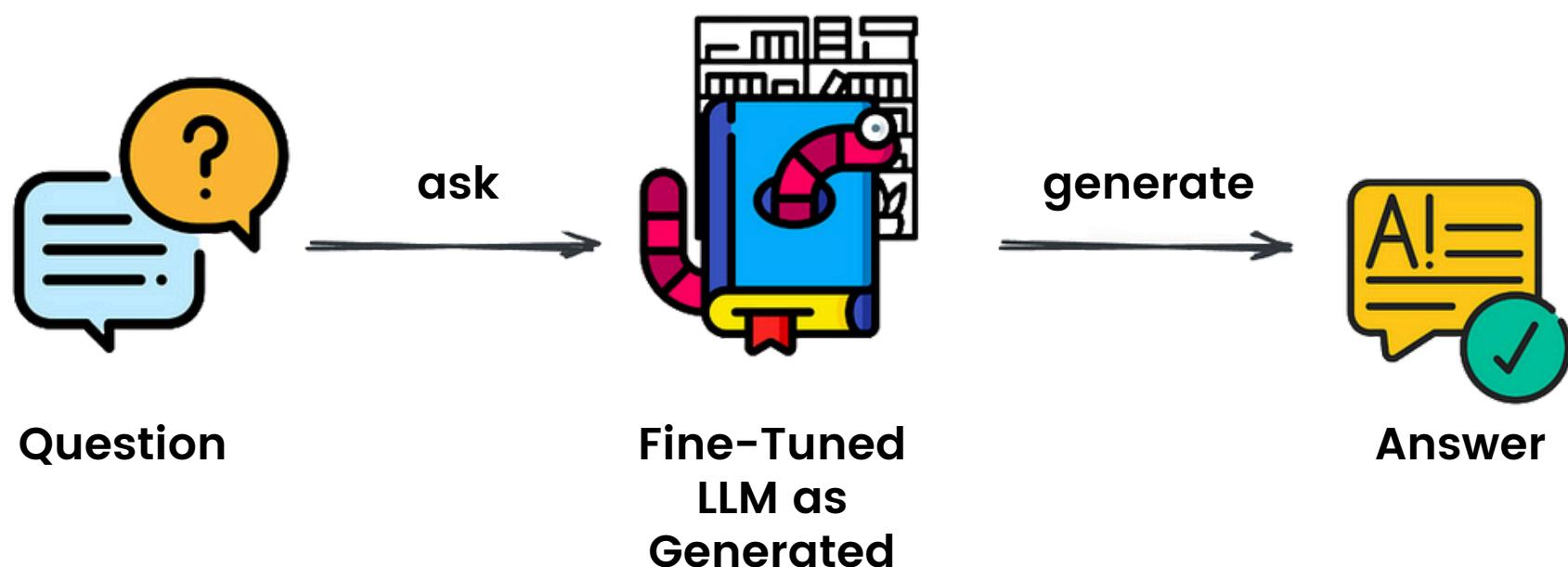
Prompt Engineering with OpenAI



- **Stay Updated:** Use the latest OpenAI API version and relevant plugins.
- **Follow Guidelines:** Ensure you adhere to any specific instructions from tools like Microsoft's Azure OpenAI GPT models.
- **Career Essential:** Knowing prompt engineering with OpenAI is crucial for your career in LLMs and generative AI.



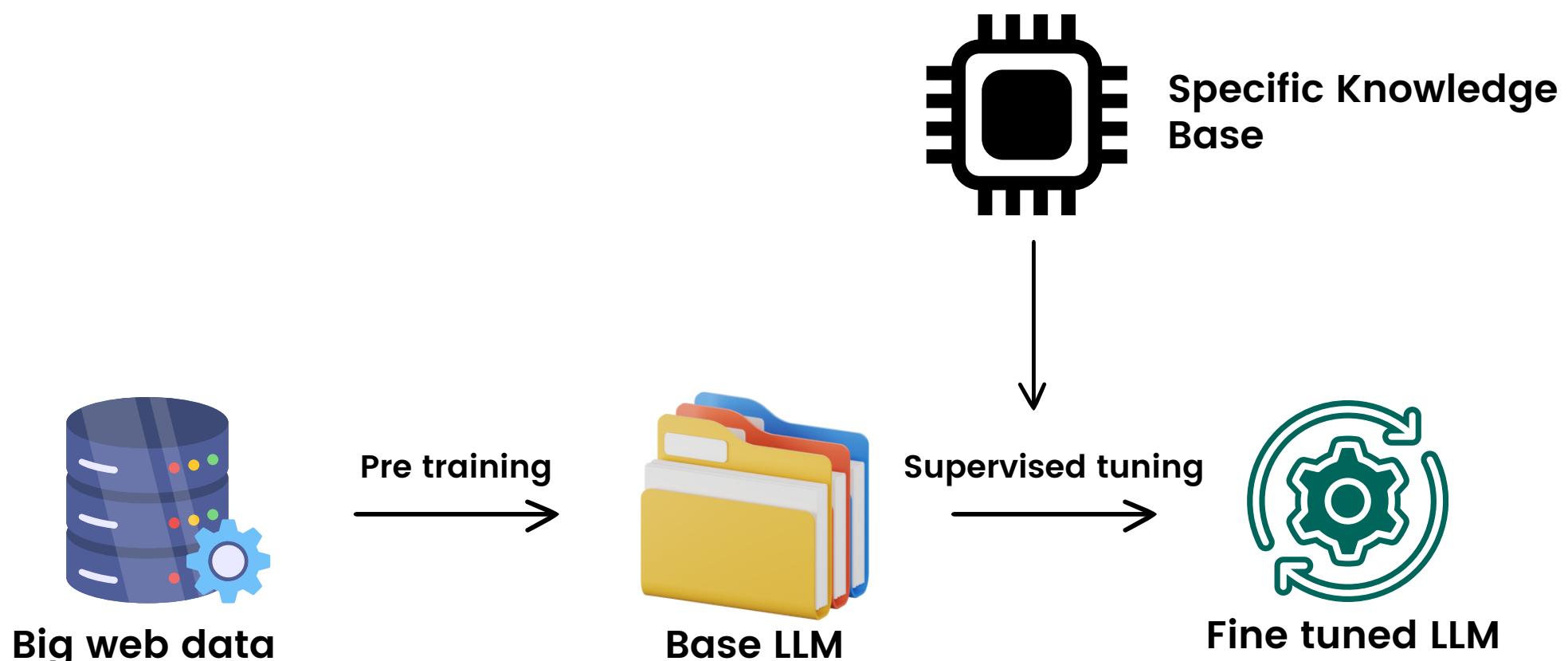
Question-Answering



- **Stay Updated:** Use the latest OpenAI API version and relevant plugins.
- **Follow Guidelines:** Ensure you adhere to any specific instructions from tools like Microsoft's Azure OpenAI GPT models.
- **Career Essential:** Knowing prompt engineering with OpenAI is crucial for your career in LLMs and generative AI.



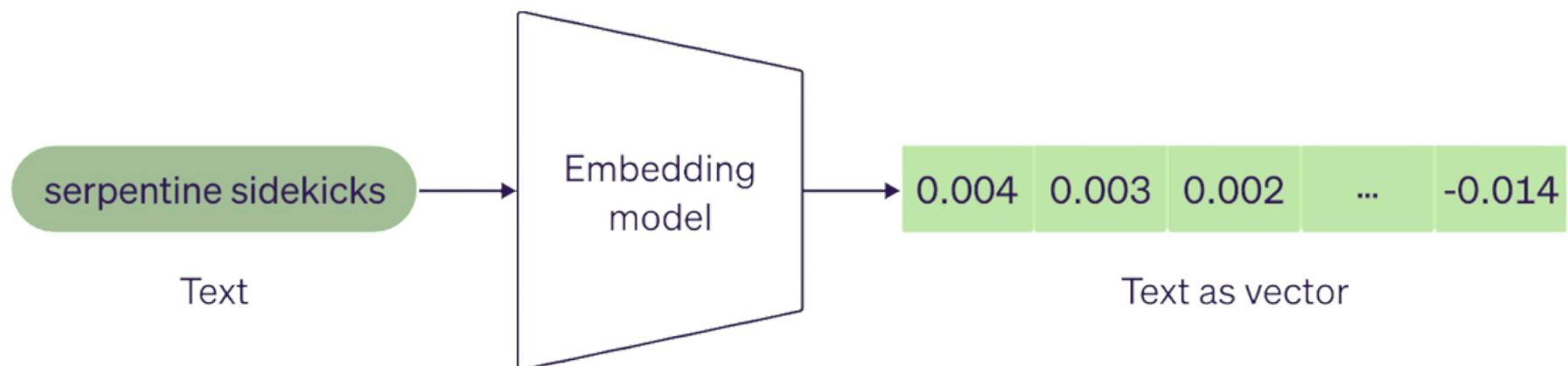
Fine-Tuning



- Enhance LLM abilities in text generation, translation, summarization, and QA.
- Customize for specific applications like chatbots or medical systems.
- Various methods for fine-tuning, including supervised learning.
- Continuous refinement ensures adaptability to evolving tasks and domains.
- Leverage labeled datasets to train LLMs efficiently and effectively.



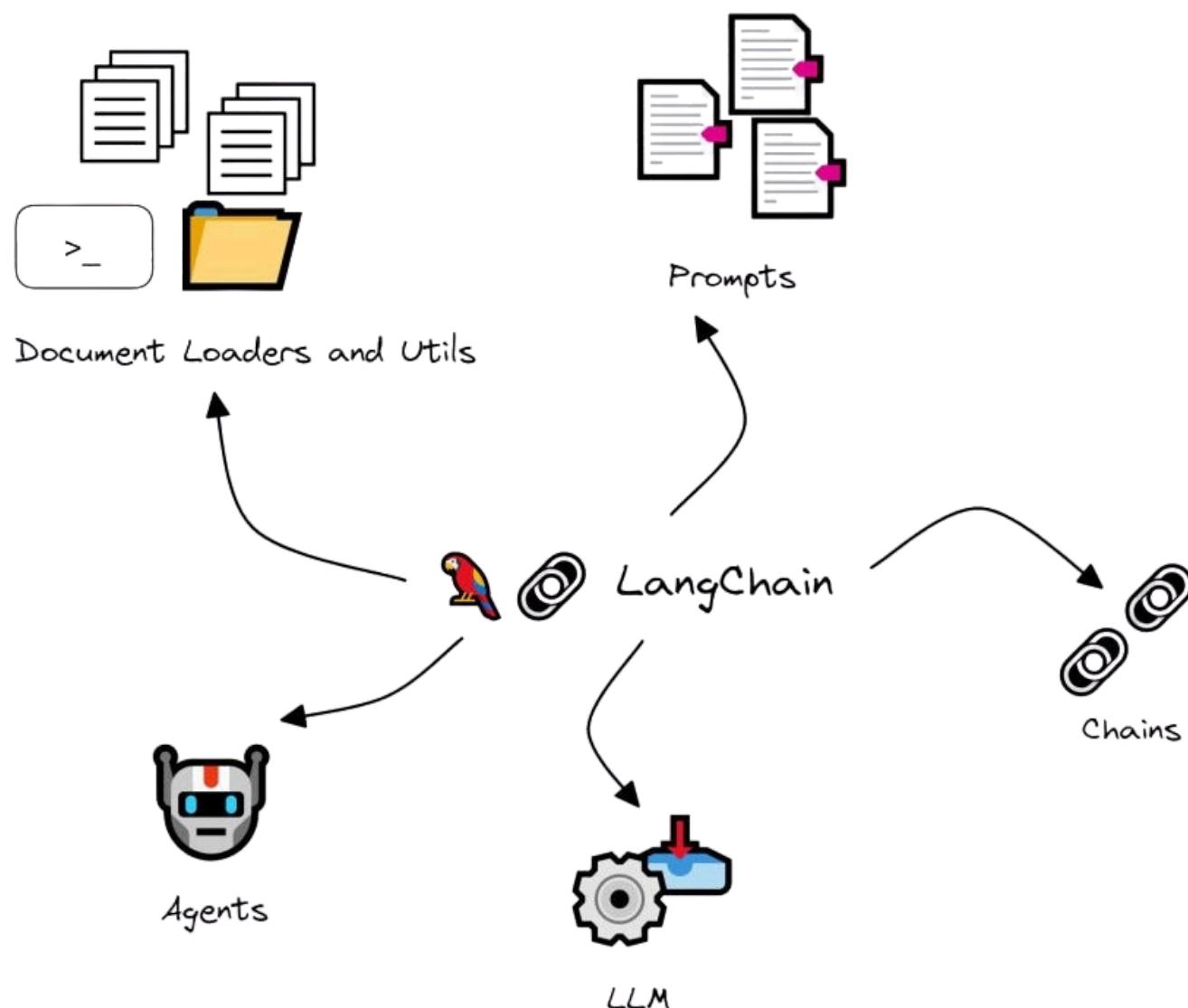
Embedding Models



- **Embedding models map natural language to vectors for downstream LLMs.**
- **Fine-tune pipelines with multiple models to capture data nuances effectively.**
- **LLMs benefit from pre-trained word embeddings, enhancing semantic understanding.**
- **Embedding models lay the groundwork for coherent and contextually relevant text generation.**



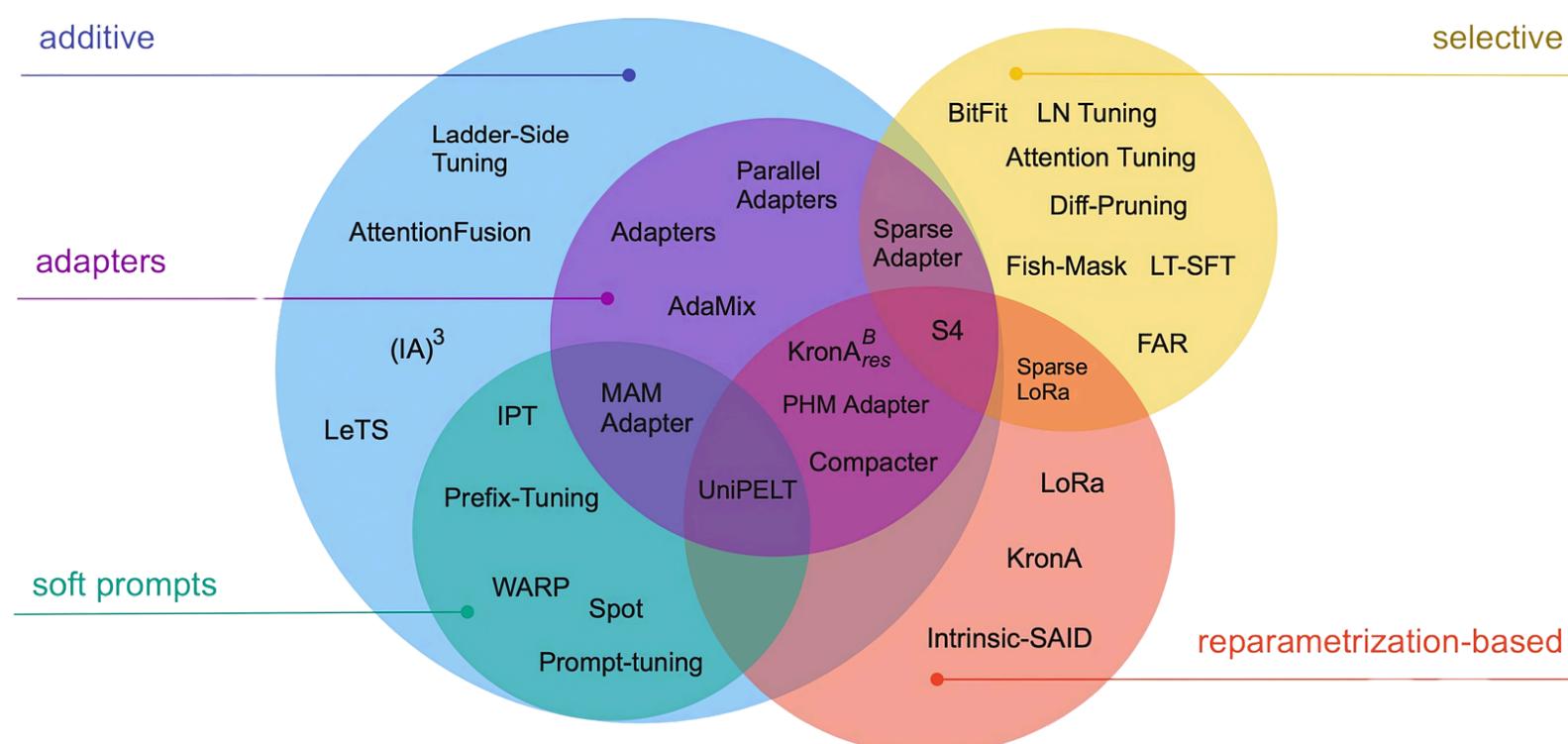
LangChain



- **Chain multiple models for complex tasks like classification, text, and code generation.**
- **Integrate with diverse systems for tasks like API calls, data science, and querying.**
- **Use Agents to interact with external systems, executing actions guided by LLMs.**
- **Agents empower LLMs to select and execute actions using a range of tools.**



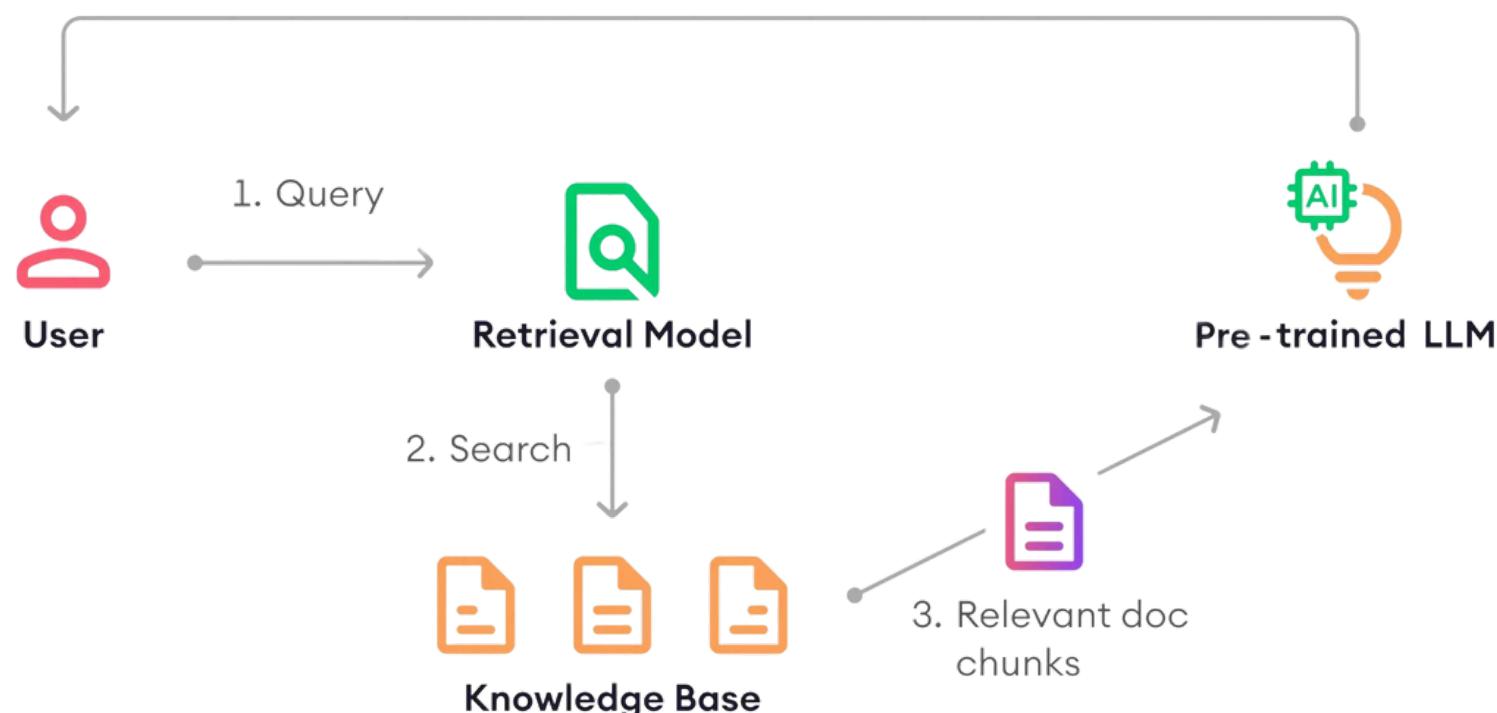
Parameter Efficiency/Tuning



- Adapt large language models like GPT or BERT for specific tasks with minimal parameter overhead.
- Add compact task-specific "adapters" to pre-trained models instead of fine-tuning the entire model.
- Reduce computational and memory requirements, making fine-tuning more feasible.



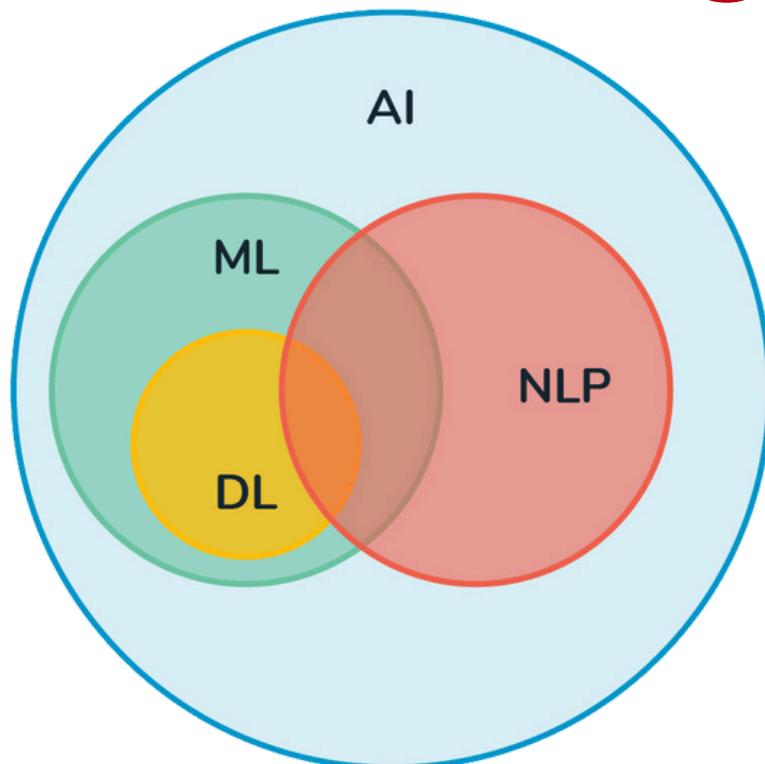
RAG



- **Retrieval-based model:** Retrieves relevant documents from a knowledge base based on input text.
- **Linking:** Connects retrieved documents with input text.
- **Generative model:** Uses linked input and documents to generate output text.
- **Integration:** Considers both input text and retrieved documents for enhanced text generation.



Natural Language Processing



- **Foundation:** LLMs and generative AI are rooted in NLP principles.
- **Training Data:** Massive datasets of text and code drive LLMs, leveraging NLP techniques.
- **Meaning Understanding:** NLP aids LLMs in comprehending data semantics and generating text.
- **Effectiveness:** Strong grasp of NLP is essential for leveraging LLMs efficiently.