Technology Topic Summary

intel ai

How to Implement Retrieval-Augmented Generation

Deploy RAG to quickly and cost-effectively customize and launch large language model (LLM) applications tailored to your business or customers.

Implementing RAG at a Glance

RAG basics:

- LLMs need fresh and specialized data to provide more reliable and relevant responses.
- RAG connects LLMs to proprietary local databases while keeping data safe.
- RAG allows frequent data updates without fine-tuning.
- RAG helps launch customized AI apps faster and cheaper.

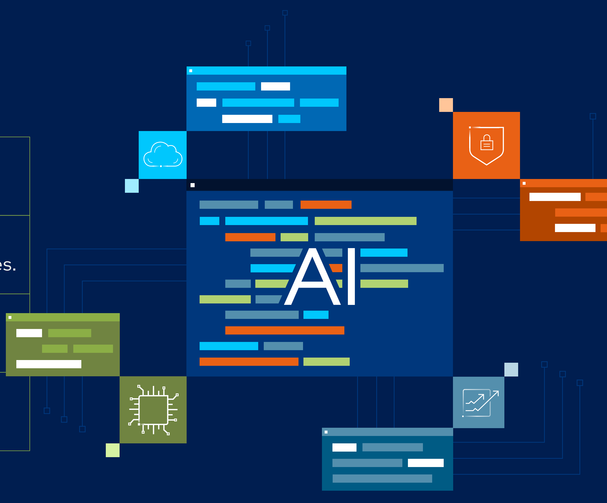
Implementing RAG:

- The right hardware makes RAG faster and more secure.
- Integrated RAG tools accelerate development and performance.
- Clean data leads to more-accurate and faster responses.
- Optimizing search and generation steps boosts RAG efficiency.

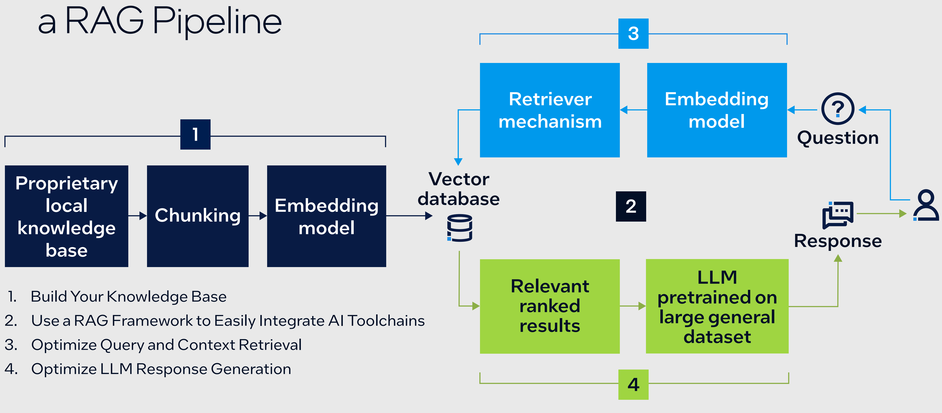
Selecting RAG

RAG is an ideal approach when:

- You want to use proprietary data to provide customized responses.
- Your LLM needs frequently updated materials to provide accurate responses.
- You want flexibility to switch between LLMs.
- You don't have time or budget for fine-tuning LLMs on fresh data.



Implementing a RAG Pipeline



1. Build Your Knowledge Base
2. Use a RAG Framework to Easily Integrate AI Toolchains
3. Optimize Query and Context Retrieval
4. Optimize LLM Response Generation

intel ai

Build and optimize your RAG pipeline

Dive into the details at intel.com/ai