RAG Overview — Page 1

What is RAG?
Retrieval-Augmented Generation (RAG) is an AI technique that improves the accuracy of responses by combining two steps:
1. Retrieving relevant information from external documents.
2. Using a Large Language Model (LLM) to generate an answer based on that information.

This ensures answers are grounded in real data instead of relying only on the model's memory.

RAG Overview — Page 2

Why Do We Need RAG?
LLMs often hallucinate or provide outdated answers. RAG solves this by:
• Connecting the model to updated knowledge.
• Providing context-specific answers.
• Reducing hallucinations.
• Allowing domain-specific customization.

RAG is widely used in enterprise knowledge bots and internal search assistants.

RAG Overview — Page 3

Basic RAG Pipeline
The typical RAG pipeline has four major steps:
1. **Document Loading** – Load PDF, CSV, text, etc.
2. **Chunking** – Break documents into smaller pieces.
3. **Embedding + Vector Storage** – Convert chunks into vectors and store in a database.
4. **Retrieval + Generation** – Retrieve relevant chunks and feed them to an LLM for answering.

This pipeline forms the foundation of all practical RAG systems.

RAG Overview — Page 4

Where is RAG Used?
RAG is used in many real-world applications:
• Customer support chatbots
• Search assistants
• Internal knowledge bots for employees
• Financial research assistants
• Legal and policy summarization tools
• Product FAQ automation

Any domain with documents benefits from RAG.

RAG Overview — Page 5

Advantages & Limitations
Advantages:
• Accurate and grounded answers
• Easy to update knowledge (no retraining)
• Domain customization

Limitations:
• Retrieval quality affects output
• Poor chunking leads to poor answers
• Storage and compute add some overhead

Conclusion: RAG is the simplest and most practical way to build real-world AI systems that rely on factual data.