Introduction to RAG

Large Language Models are powerful reasoning machines, but they know nothing about your data.

Retrieval-Augmented Generation (RAG) is a popular technique for making your data available to an LLM. This gives you the reasoning power of LLMs, but with knowledge of your data.

Instead of just asking a question of an LLM, RAG first retrieves prior context from a memory system. This context is passed with the question to the LLM to provide knowledge of your data for solving your unique problem.

Implementing RAG requires the **indexing** of your data, the **querying** of your data prior to LLM submission, and the **evaluation** of the LLM response. RAGStack provides tools for each of these steps, and enables advanced RAG techniques for improving responses and reducing LLM round-trips.

Read on to learn more about each step of RAG and how to implement it with RAGStack.

Components of the RAG process

Index Data for Retrieval
Whatever type of data you have, from unstructured text to compressed video, is converted into a structured format that allows it to be easily retrieved. This conversion is performed by segmenting or “chunking” the input data into many pieces, and then embedding each chunk as a vector in a vector database. A vector is a series of numbers that represents the characteristics of each chunk of input data. These vectors are then indexed for searching.

Query Data
When you ask a question of your data, this index is queried for relevant information - semantically similar vectors, for example - and the results of the index query are passed along with your question to an LLM. LLMs possess amazing powers of reasoning and deduction, but they know nothing about your data. By including the results of the index query, you supply valuable contextual information for the LLM to reason through and return an answer to your question.

Evaluate RAG Performance
Receiving an answer back from the LLM marks the end of the RAG process, but not the end of the story. Each step of the process from chunking strategy to answer quality can be tested, evaluated, and optimized for your production systems.

Each of RAG’s components is covered in detail in the following sections. We recommend you start with Index Data for Retrieval.

1. Index Data for Retrieval
2. Query Data
3. Evaluate RAG Performance

When you're confident in your RAG understanding, take your RAG application to the next level and check out some advanced RAG techniques in Advanced RAG Techniques.