RAG Pipeline Implementation and Results

1. Introduction

This document presents the design choices and results of a Retrieval-Augmented Generation (RAG) pipeline built in Google Colab using LlamaIndex and Gemini. The pipeline was designed to process a PDF certificate, retrieve relevant information from the document, and generate answers to questions about its quality control methods and storage conditions.

The RAG pipeline included the following main steps:

1. PDF parsing
2. Document chunking
3. Embedding generation
4. Vector-based retrieval
5. Response generation

2. Short Explanation of Design Choices

2.1 Embedding Model

Embedding model: sentence-transformers/all-MiniLM-L6-v2 (HuggingFace)

I selected sentence-transformers/all-MiniLM-L6-v2 as the embedding model for three practical reasons. First, it runs entirely locally, meaning there are no external API calls or per-request costs, and there is no need to send pharmaceutical compliance data to a third-party embedding API.

Second, it is fast and lightweight enough to run efficiently on a free Google Colab instance.

Finally, this is the model that has been used consistently in previous stages of the externship, particularly Project 5 Step 1. Using the same model keeps the results comparable across my RAG experiments.

2.2 Chunking Strategy

Chunking strategy: Overlapping chunking (chunk_size=300, chunk_overlap=50)

I chose overlapping chunking rather than fixed-length chunking without overlap because this document contains label/value pairs that can sit right at natural break points. For example, "Date of Manufacture:" may be followed immediately by "20240315".

A hard cut with no overlap could separate a field's label from its value across two different chunks, making that information more difficult to retrieve when the system receives a query about it.

The 50-token overlap helps protect against this specific failure mode while adding only a small amount of storage overhead. This is a reasonable tradeoff for a short three-page document.

2.3 Retrieval Method

Retrieval method: Pure vector (semantic) retrieval

I selected pure vector-based semantic retrieval instead of keyword-only or hybrid retrieval because the required prompts ask conceptual questions such as "what test methods" and "what storage conditions", rather than asking for an exact phrase.

Semantic retrieval can match a query such as "quality control" to a relevant document section such as "Product Release Criteria", even when there is not an exact word-for-word match.

Hybrid retrieval, which combines vector and BM25 keyword retrieval, was explored in the previous task and is a stronger choice for longer and more varied documents. However, for this specific three-page, single-topic SDF, pure vector retrieval is simpler and sufficient. The document is small enough that recall is not a major concern.

3. Query Results

Prompt 1: What test methods were used for quality control?

The test methods and criteria used for quality control include:

- Flow Rate Test: Executed per SOP-QC-042.
- Autoclave (Pump tubing): Tested at 121°C for more than 15 minutes.
- Gamma Irradiation (Inlets): Measured at 25.0 – 40.0 kGy.
- Pressure Integrity: Tested at a maximum of 0.5 bar with a 5-minute hold.
- Visual Inspection: Inspected for visible defects.
- Package Integrity: Inspected to ensure it is sealed with no damage.
- Bio-safety: Evaluated using the United States Pharmacopeia (USP) <88> Biological Reactivity Test Class VI.

Prompt 2: What are the storage conditions specified in the certificate?

The recommended storage temperature for the flow kits is specified as > +5 °C. Extended storage below +5 °C could lead to brittleness or cracking of the plastic connectors.

Additionally, the operating temperature for the kits is +2 °C to +40 °C. Allowing the kits to acclimate to a warmer temperature before use reduces the risk of damage during setup and handling.