RAG Pipeline: -

Making a RAG for all the Harry Potter Books.

Load in the pdfs and all the dependencies.

We load in the pdfs as text using PyPdf from Langchain.

Then we perform chunking on these pdfs. For chunking we use Recursive Character Text Splitter. This divides the documents into chunks.

(Max tokens – 1024, with an overlap – 200)
We then convert the documents into the embeddings using an Embedding model and store it in the **Chroma DB.**

LLM – Llama3 model (8B parameters)

Now we can perform retrieval. We pass a query, gets its embedding and find the most similar chunks from the Database and retrieve it.
This is then passed as suggested material to the LLM with the query as the prompt.

Then we do system prompting – telling the RAG to follow the suggestive material and answer the questions.
Once trained, we can use it to get our desired results.

Other RAG Resources:-

Chunking Strategies - https://www.mongodb.com/developer/products/atlas/choosing-chunking-strategy-rag/

Advanced RAGs - https://medium.com/decodingml/the-4-advanced-rag-algorithms-you-must-know-to-implement-5d0c7f1199d2