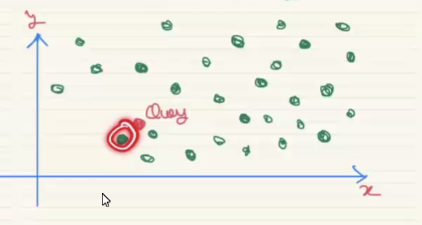
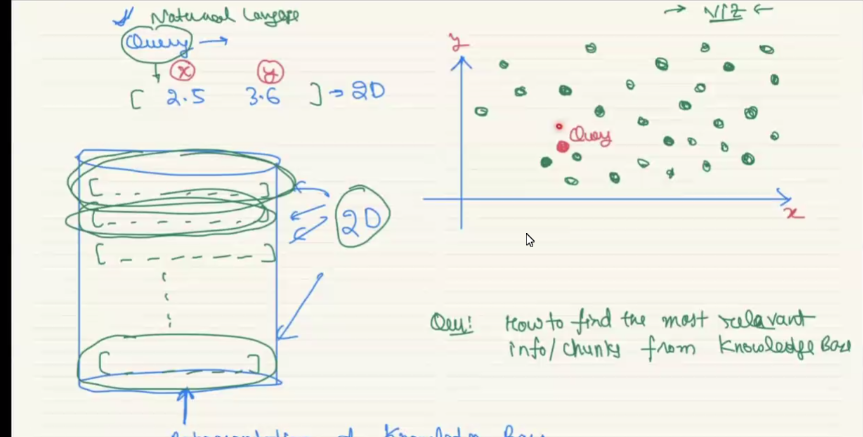
**PROCESS OF FINDING THE CORRECT CHUNK**

All the vectorize data will be represented as the below graph, our human input also will be converted to vectorized and pass to the knowledge base to find the correct chunk, that correct chunk can be found by the closest data from the query.

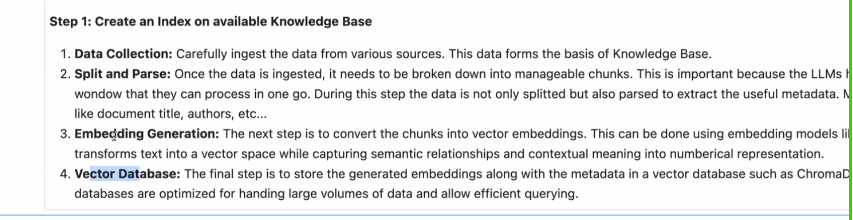


****

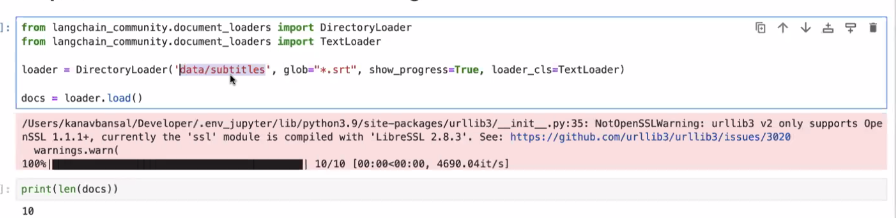
**CROMA DB:**

It is vector database used to vector representation of data, and it is open source, this is the DB used to retrieve the latest info, that is combined with model

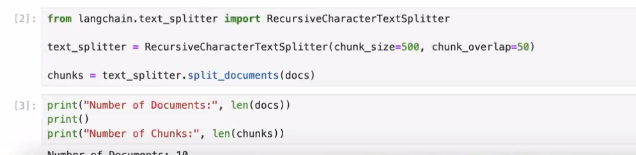
**STPES TO CREATE RETERIVER**



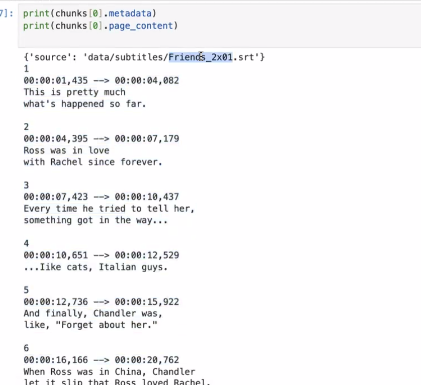
Knowledge are will be in different format, DirectoryLoader and TextLoader helps to load all the files with texts in one single variable in the below code.



Recursive character text splitter is the class helps to chunk it, Each chunk will have 500 character, second chunk are starting from after 50 overlapping character from first chunk, this is the very important one, overlapping text give raises to hold all file same text, so that’s second one is starting after some 50 text, this is the manual check, we have to experiment with adjusting the overlap value. The is how a RAG system uses, Below is the code which has this.

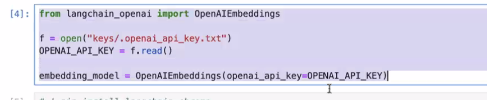




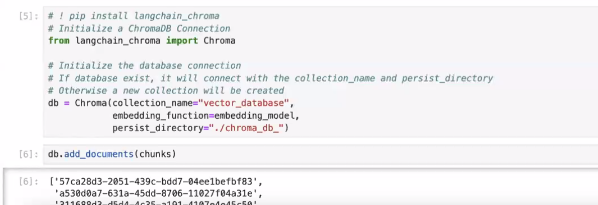


Before chunking the document it is important to clean the text files too.

Now all the chunks are converted into vector with the help of OpenAIEmbedding

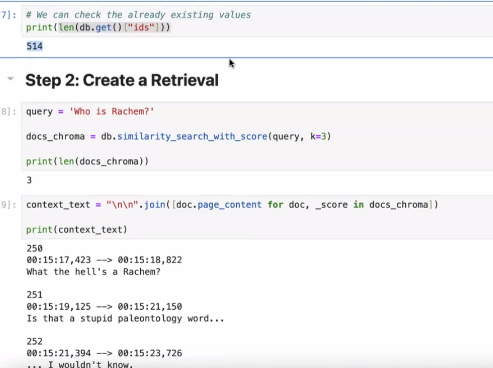


Following is the command helps to use chroma



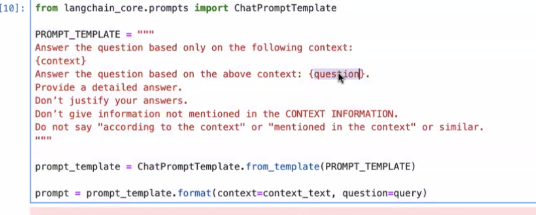
**RETERIVING THE DATA DROM DB**

Similarity search with score is the one that take user query convert it to vector and uses cosine similarity to find the correct answer for the user question.

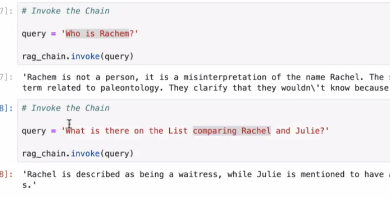
****

**FINAL STEP**

Embedding this db with prompt to take the latest info.

****

**Example query**

****

**PROJECT:** See how to use hugging face wishper model to get the input from user (Audio to text)