Retrieval Augmented Generation (RAG) for Alzheimer's Disease Research

Contents

Introduction

In the realm of medical research, where breakthroughs and discoveries occur at a rapid pace, staying updated with the latest developments is paramount for researchers. This Retrieval Augmented Generation (RAG) application is designed to keep medical researchers, particularly in Alzheimer's disease, updated with the latest advancements in the field. By combining Large Language Models (LLM) with specialized information retrieval mechanisms, it efficiently delivers relevant information from scholarly articles and research publications. This report outlines the methodology, architecture, outcomes, and future improvements of the system.

Knowledge base preparation

Dataset selection:

In selecting the dataset for Alzheimer's disease research, a curated collection of recent articles was compiled. These articles cover a range of topics, including novel treatment strategies, advancements in early diagnosis, and insights into disease pathology. By incorporating diverse perspectives and findings from reputable sources such as academic institutions and medical research organizations, the dataset aims to provide a comprehensive understanding of current developments in Alzheimer's research.

Data pre-processing and chunking:

In preparing the knowledge base for the RAG, the datasets were scraped from articles' webpages and then pre-processed by removing tabs and newline characters. These cleaned datasets were then combined into a single document and subsequently chunked by sentences. Chunking by sentences was chosen to enhance retrieval and processing efficiency, as handling smaller units of information allows for faster responses and increases the likelihood of retrieving highly relevant chunks that address the user's query.

Embedding:

After segmenting the dataset into manageable chunks, the next crucial step is embedding, where each chunk of text is transformed into a numerical representation capturing its semantic meaning. Hugging Face is an open-source platform that provides a wide range of pre-trained models and embedding models. For this application, understanding the context and meaning of sentence is important. Hence, sentence level embedding is opted, for which "BAAI/bge-small-en-v1.5” pretrained embedding model from the Beijing Academy of Artificial Intelligence (BAAI) is used. This choice was made considering several factors. Firstly, the model's specialization in English text makes it suitable for processing the dataset, primarily comprising English-language articles on Alzheimer's research. Additionally, "small" model ensures a balance between computational efficiency and performance, making it optimal for resource-constrained environments. Leveraging the "BAAI/bge-small-en-v1.5" embedding model aims to facilitate the transformation of segmented text into meaningful numerical representations, advancing the capabilities of the RAG system.

Vector database:

After embedding the data, a vector database is created and indexed with embeddings and ChromaDB is chosen as database solution for its tailored advantages that closely align with the requirements of the RAG system. ChromaDB stands out for its specialized focus on efficiently storing and retrieving high-dimensional vector data, particularly text embeddings. Moreover, its capability to handle large-scale vector datasets, offering scalability and performance optimizations ensures that the RAG system can scale effectively as the size of the dataset grows. Additionally, it is user friendly and can be implemented easily.

Frameworks and models utilized in RAG development

Following components serve as the cornerstone of the RAG's development, enabling seamless integration with external data sources, efficient local deployment, and sophisticated language processing capabilities.

LlamaIndex

As a foundation to this RAG application, LlamaIndex, a comprehensive framework for creating RAG, is used to connect LLM with external data sources, which in this case is articles on Alzheimer’s disease. It offers a range of tools for data ingestion, querying, and indexing and include connectors for various data sources and formats. LlamaIndex focuses on building dedicated search and retrieval RAG applications and is also suitable for processing large amounts of data, hence is an ideal framework for this RAG application.

Ollama

Ollama, an open-source framework tailored for running LLMs on local devices is utilised for accessing LLM in local environment. It is known for its simplicity, efficiency, and speed. It

provides an ideal environment for deploying and managing language models in a local setting and has over 100 curated LLM’s and include some notable language models like Phi-3 and Llama3. Ollama's seamless synchronization with the Hugging Face Hub further enhances accessibility, granting access to a diverse range of LLMs with ease and offers a user-friendly environment to work with LLM’s.
Ollama is installed locally, and the specified Phi-3 model is pulled before running the python code for RAG development.

LLM

Phi-3, which is a lightweight open AI model developed by Microsoft is used for the RAG application and it is most capable and cost-effective small language model that outperforms same size or large size models. Phi-3's compact size makes it an ideal fit for this RAG application, especially considering its ability to operate seamlessly in compute-limited environments. Moreover, Phi-3 has longer context window (128K tokens), thus enabling to handle large text content ranging from documents to web pages. These capabilities, coupled with Phi-3's robust logic and reasoning capabilities, position it as the perfect candidate for this RAG application.

Prompt template

For querying the LLM effectively, a structured prompt template is developed that guides the interaction between users and the system. The template consists of following sections for context-based question answering.

Context information:
By embedding context information within the prompt, provides necessary background information for generating response that are accurate and contextually relevant.
Question prompt:
The question prompt specifies the user query, and it directs the system to generate a response tailored to the query, enhancing precision and relevance.
System message:
This guides users to align queries with application domain for coherent responses.
User message:
Users can pose questions relevant to the domain of the RAG, and for non-domain queries, the RAG responds indicating it does not have an answer due to the question being out of context. This approach upholds the integrity of the application's aim, which is to provide

information on research related to Alzheimer's disease. By encouraging users to stay within the scope of the provided context, the RAG fosters a focused and productive interaction experience.

Query pipeline

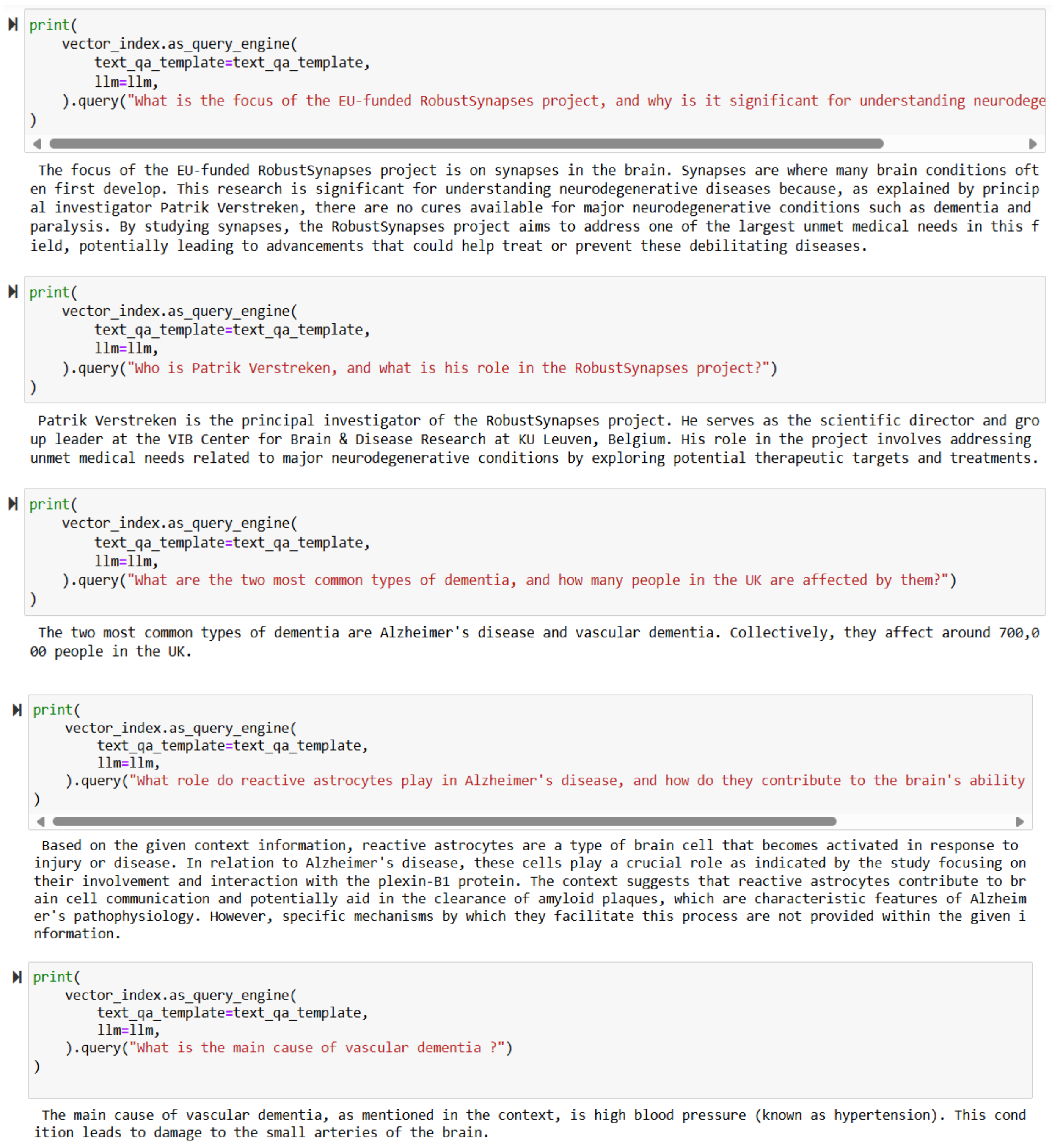
A query pipeline is implemented for seamless communication between the user and the system. It leverages the prompt template and Phi-3 model, to generate responses tailored to users' queries. The pipeline processes the user query, extracting relevant information and the context from the input/query and then retrieve information related to the query from the vector database enriching the context for generating responses. The system presents the user query along with the retrieved information using the prompt template and utilising the LLM, the pipeline generates a response tailored to the user query and the provided context.

Testing RAG performance

To test the RAG's effectiveness, a series of test queries were devised. Initially, the specified LLM was queried with an irrelevant question, "Where is Kerala ?" to gauge its ability to provide accurate responses unrelated to the RAG's focus. Subsequently, after integrating the specified LLM into the RAG, it was presented with the following questions, all related to articles on Alzheimer's disease research:

1. What is the focus of the EU-funded RobustSynapses project, and why is it significant ?
2. Who is Patrik Verstreken, and what role does he play in the RobustSynapses project ?
3. What are the two most common types of dementia, and how many people in the UK are affected by them ?
4. What role do reactive astrocytes play in Alzheimer's disease, and how do they contribute to the brain's ability to clear amyloid plaques ?
5. What is the main cause of vascular dementia ?

The RAG system successfully provided accurate responses derived from the articles, showcasing its ability to comprehend and respond to context-specific queries effectively.



```
print(
    vector_index.as_query_engine(
        text_qa_template=text_qa_template,
        llm=llm,
    ).query("What is the focus of the EU-funded RobustSynapses project, and why is it significant for understanding neurodege
)
```

```
 The focus of the EU-funded RobustSynapses project is on synapses in the brain. Synapses are where many brain conditions oft
en first develop. This research is significant for understanding neurodegenerative diseases because, as explained by princip
al investigator Patrik Verstreken, there are no cures available for major neurodegenerative conditions such as dementia and
paralysis. By studying synapses, the RobustSynapses project aims to address one of the largest unmet medical needs in this f
ield, potentially leading to advancements that could help treat or prevent these debilitating diseases.
```

```
print(
    vector_index.as_query_engine(
        text_qa_template=text_qa_template,
        llm=llm,
    ).query("Who is Patrik Verstreken, and what is his role in the RobustSynapses project?")
)
```

```
 Patrik Verstreken is the principal investigator of the RobustSynapses project. He serves as the scientific director and gro
up leader at the VIB Center for Brain & Disease Research at KU Leuven, Belgium. His role in the project involves addressing
unmet medical needs related to major neurodegenerative conditions by exploring potential therapeutic targets and treatments.
```

```
print(
    vector_index.as_query_engine(
        text_qa_template=text_qa_template,
        llm=llm,
    ).query("What are the two most common types of dementia, and how many people in the UK are affected by them?")
)
```

```
 The two most common types of dementia are Alzheimer's disease and vascular dementia. Collectively, they affect around 700,0
00 people in the UK.
```

```
print(
    vector_index.as_query_engine(
        text_qa_template=text_qa_template,
        llm=llm,
    ).query("What role do reactive astrocytes play in Alzheimer's disease, and how do they contribute to the brain's ability
)
```

```
 Based on the given context information, reactive astrocytes are a type of brain cell that becomes activated in response to
injury or disease. In relation to Alzheimer's disease, these cells play a crucial role as indicated by the study focusing on
their involvement and interaction with the plexin-B1 protein. The context suggests that reactive astrocytes contribute to br
ain cell communication and potentially aid in the clearance of amyloid plaques, which are characteristic features of Alzheim
er's pathophysiology. However, specific mechanisms by which they facilitate this process are not provided within the given i
nformation.
```

```
print(
    vector_index.as_query_engine(
        text_qa_template=text_qa_template,
        llm=llm,
    ).query("What is the main cause of vascular dementia ?")
)
```

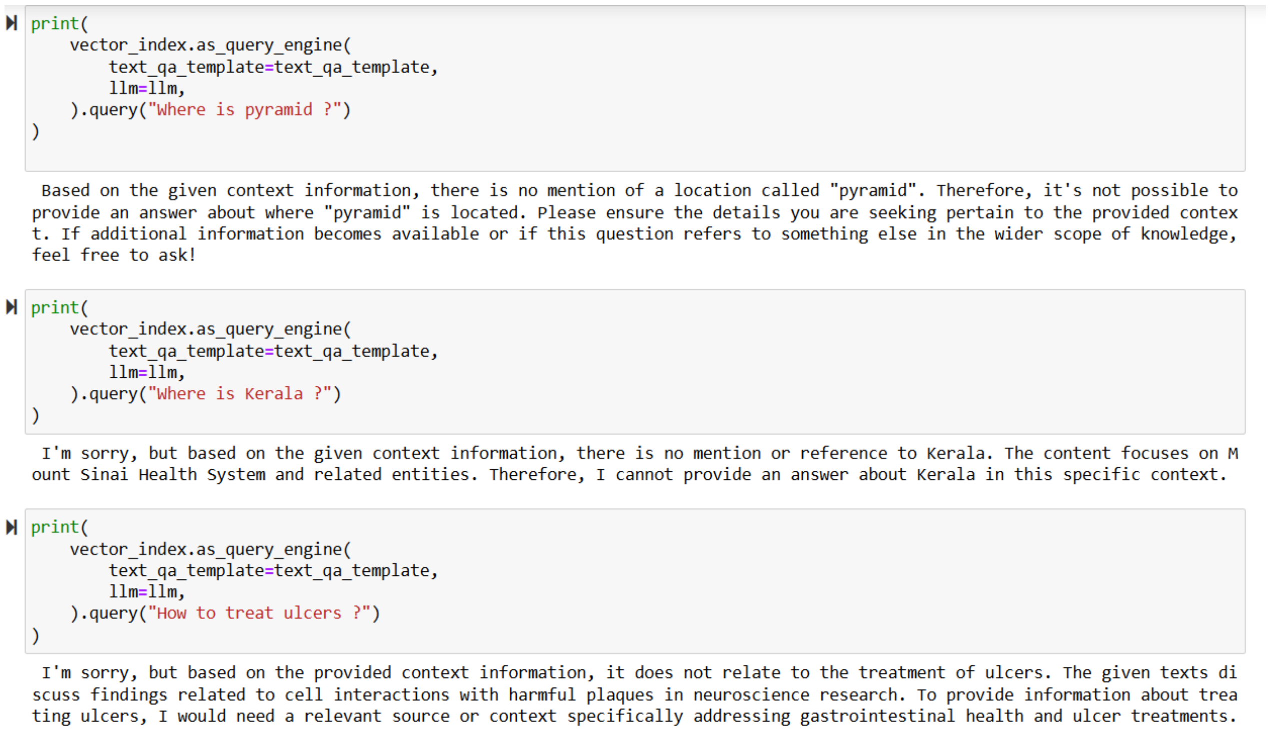
```
 The main cause of vascular dementia, as mentioned in the context, is high blood pressure (known as hypertension). This cond
ition leads to damage to the small arteries of the brain.
```

Figure 1: RAG answering domain specific queries

Furthermore, to assess the system's ability to discern and refrain from answering out-of-context questions, additional following queries unrelated to Alzheimer's disease research were posed.

1. Where is pyramid ?
2. Where is Kerala ?
3. How to treat ulcers ?

These questions aimed to test the system's adherence to the defined scope of providing information solely on Alzheimer's disease research articles. The RAG system appropriately refrained from answering and indicated these are out-of-context questions, thereby maintaining the integrity and relevance of its responses within the defined domain.



```
print(
    vector_index.as_query_engine(
        text_qa_template=text_qa_template,
        llm=llm,
    ).query("Where is pyramid ?")
)
```

```
 Based on the given context information, there is no mention of a location called "pyramid". Therefore, it's not possible to
provide an answer about where "pyramid" is located. Please ensure the details you are seeking pertain to the provided contex
t. If additional information becomes available or if this question refers to something else in the wider scope of knowledge,
feel free to ask!
```

```
print(
    vector_index.as_query_engine(
        text_qa_template=text_qa_template,
        llm=llm,
    ).query("Where is Kerala ?")
)
```

```
 I'm sorry, but based on the given context information, there is no mention or reference to Kerala. The content focuses on M
ount Sinai Health System and related entities. Therefore, I cannot provide an answer about Kerala in this specific context.
```

```
print(
    vector_index.as_query_engine(
        text_qa_template=text_qa_template,
        llm=llm,
    ).query("How to treat ulcers ?")
)
```

```
 I'm sorry, but based on the provided context information, it does not relate to the treatment of ulcers. The given texts di
scuss findings related to cell interactions with harmful plaques in neuroscience research. To provide information about trea
ting ulcers, I would need a relevant source or context specifically addressing gastrointestinal health and ulcer treatments.
```

Figure 2: RAG answering non-domain queries

Conclusion and future improvements

This report presents the development of a RAG application focusing on Alzheimer's field helping researchers to seek information on research happening in the field.
Some of the improvements that could be made to this RAG application include integrating multimodal capabilities, such as processing images and audio, thus helping to significantly enrich the user experience. By incorporating image recognition, speech-to-text, and other AI modalities, the application can provide more comprehensive responses and cater to diverse user preferences.
Moreover, to keep the RAG up to date, knowledge base can be expanded with the latest research articles, datasets, and domain-specific information, which can be achieved through periodic data scraping, integration with external knowledge sources to provide users with the most accurate information.

Additionally, optimizing the model architecture of the RAG system can further enhance its performance and efficiency. Experimenting with different transformer variants, attention mechanisms can help fine-tune the model to better suit the application's requirements and improve its capabilities.

References

European Commission (2022) *Brain study opens door to potential new disease treatments.* Research and Innovation website, success stories. Available at: Brain study opens door to potential new disease treatments - Research and Innovation (ec.europa.eu)

University of Manchester (2022) *Blood vessel breakthrough*. Faculty of Biology, Medicine and Health website, stories. Available at: Blood vessel breakthrough - The University of Manchester (manchester.ac.uk)

Mount Sinai Health System (2024) *Altering cellular interactions around amyloid plaques may offer novel Alzheimer's treatment strategies.* Mount Sinai Newsroom website. Available at: Altering cellular interactions around amyloid plaques may offer novel Alzheimer's treatment strategies - Mount Sinai Health System (mountsinai.org)

Medical Xpress (2024) *New door opens for earlier diagnosis and potential treatment.* Medical Xpress website, News. Available at: New door to earlier diagnosis, potential treatment - MedicalXpress (medicalxpress.com)

Llama Meta (2024) *Introducing LlamaIndex: A comprehensive framework for RAG applications.* Llama Meta website, Integration Guides. Available at: LlamaIndex integration guides - Meta (meta.com)

Microsoft Azure (2022) *Introducing Phi-3: Redefining what's possible with SLMs.* Microsoft Azure Blog. Available at: Introducing Phi-3: Redefining what's possible with SLMs - Microsoft Azure Blog (microsoft.com)

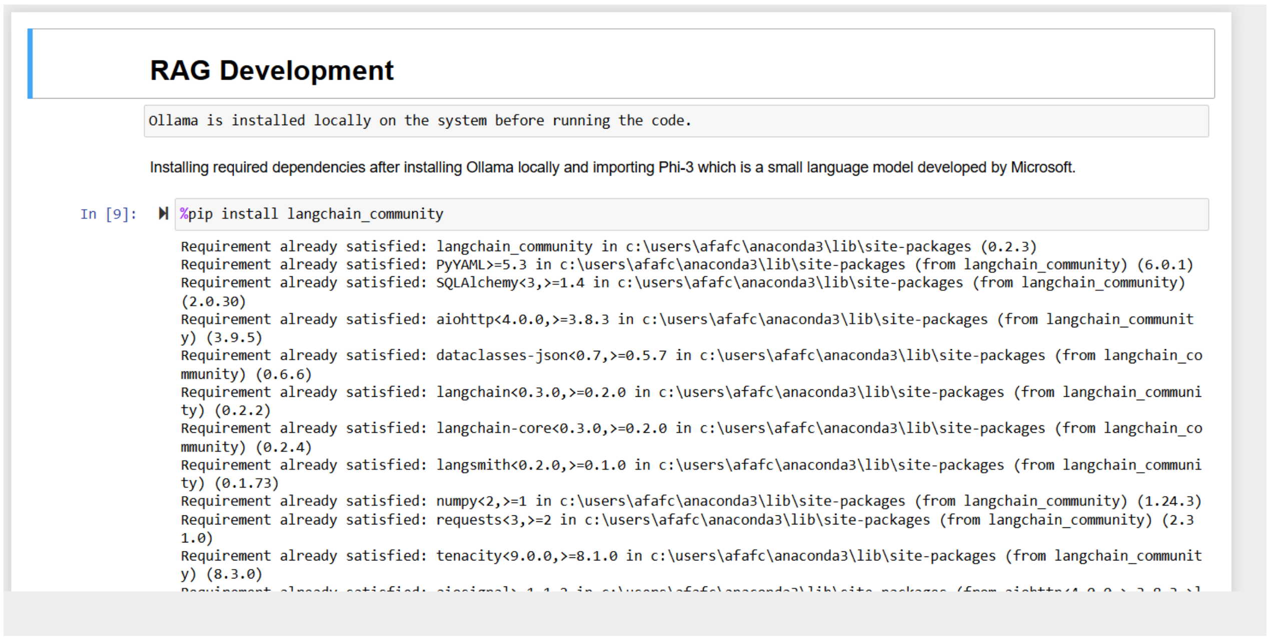
PyImageSearch (2024) An inside look: *Exploring Ollama for on-device AI.* PyImageSearch website, Blog. Available at: Inside look: Exploring Ollama for on-device AI - PyImageSearch (pyimagesearch.com)

Superlinked (2024) *Evaluation of RAG retrieval chunking methods.* VectorHub Articles website. Available at: Evaluation of RAG retrieval chunking methods - Superlinked (superlinked.com)

Amanah Luwalia (2024) *Enhancing semantic search with BAAI embedding model and ontology enrichment using vector database.* Medium website, Blog. Available at:

https://medium.com/@amanahluwalia/enhancing-semantic-search-with-baai-embedding-model-and-ontology-enrichment-using-vector-database-b660e900292c (medium.com)

Appendix



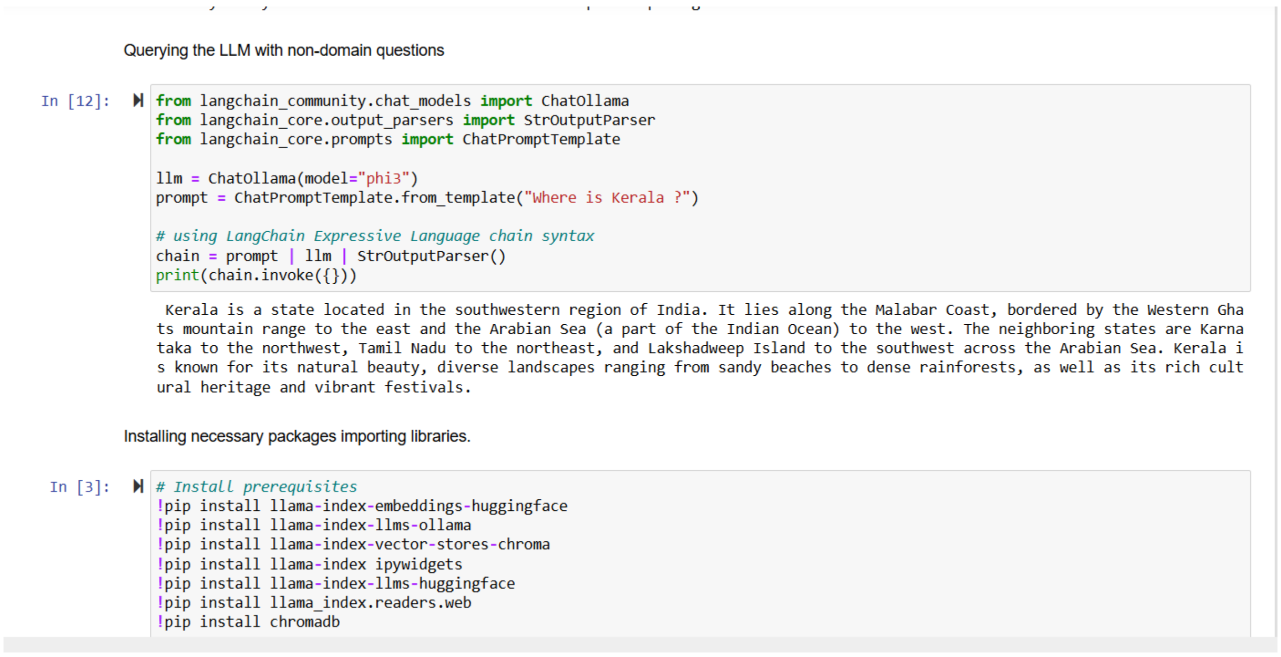
RAG Development

```
Ollama is installed locally on the system before running the code.
```

Installing required dependencies after installing Ollama locally and importing Phi-3 which is a small language model developed by Microsoft.

In [9]:
```
%pip install langchain_community
```

```
Requirement already satisfied: langchain_community in c:\users\afafc\anaconda3\lib\site-packages (0.2.3)
Requirement already satisfied: PyYAML>=5.3 in c:\users\afafc\anaconda3\lib\site-packages (from langchain_community) (6.0.1)
Requirement already satisfied: SQLAlchemy<3,>=1.4 in c:\users\afafc\anaconda3\lib\site-packages (from langchain_community)
(2.0.30)
Requirement already satisfied: aiohttp<4.0.0,>=3.8.3 in c:\users\afafc\anaconda3\lib\site-packages (from langchain_communit
y) (3.9.5)
Requirement already satisfied: dataclasses-json<0.7,>=0.5.7 in c:\users\afafc\anaconda3\lib\site-packages (from langchain_co
mmunity) (0.6.6)
Requirement already satisfied: langchain<0.3.0,>=0.2.0 in c:\users\afafc\anaconda3\lib\site-packages (from langchain_communi
ty) (0.2.2)
Requirement already satisfied: langchain-core<0.3.0,>=0.2.0 in c:\users\afafc\anaconda3\lib\site-packages (from langchain_co
mmunity) (0.2.4)
Requirement already satisfied: langsmith<0.2.0,>=0.1.0 in c:\users\afafc\anaconda3\lib\site-packages (from langchain_communi
ty) (0.1.73)
Requirement already satisfied: numpy<2,>=1 in c:\users\afafc\anaconda3\lib\site-packages (from langchain_community) (1.24.3)
Requirement already satisfied: requests<3,>=2 in c:\users\afafc\anaconda3\lib\site-packages (from langchain_community) (2.3
1.0)
Requirement already satisfied: tenacity<9.0.0,>=8.1.0 in c:\users\afafc\anaconda3\lib\site-packages (from langchain_communit
y) (8.3.0)
```



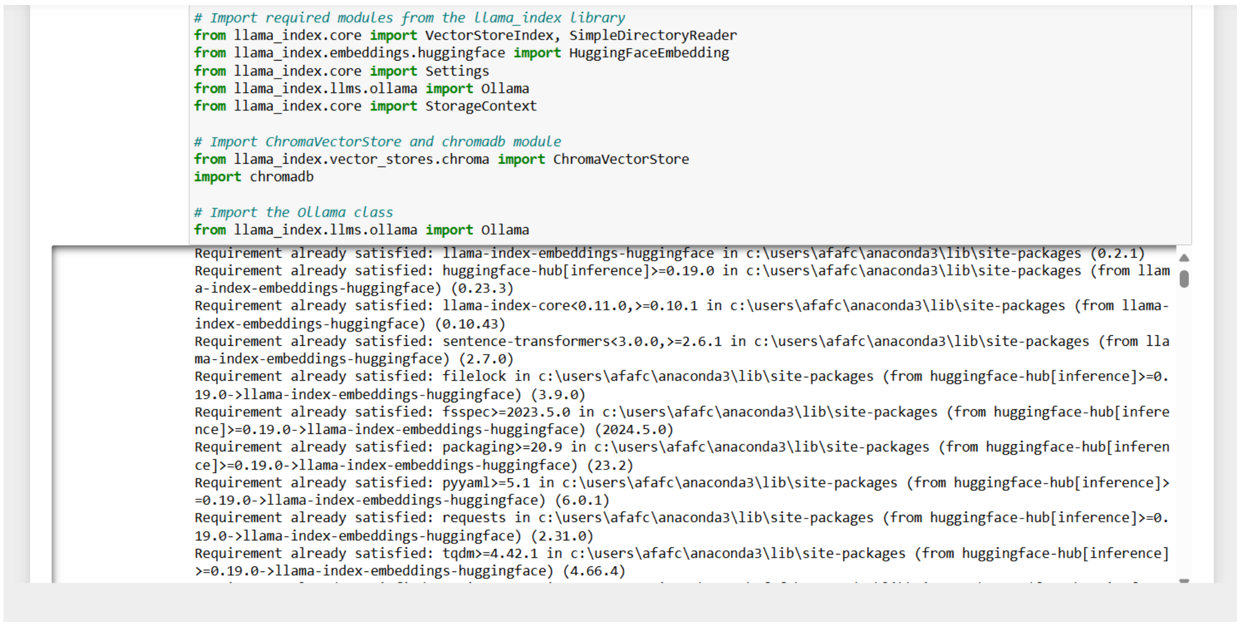
Querying the LLM with non-domain questions

In [12]:
```
from langchain_community.chat_models import ChatOllama
from langchain_core.output_parsers import StrOutputParser
from langchain_core.prompts import ChatPromptTemplate

llm = ChatOllama(model="phi3")
prompt = ChatPromptTemplate.from_template("Where is Kerala ?")

# using LangChain Expressive Language chain syntax
chain = prompt | llm | StrOutputParser()
print(chain.invoke({}))
```

```
 Kerala is a state located in the southwestern region of India. It lies along the Malabar Coast, bordered by the Western Gha
ts mountain range to the east and the Arabian Sea (a part of the Indian Ocean) to the west. The neighboring states are Karna
taka to the northwest, Tamil Nadu to the northeast, and Lakshadweep Island to the southwest across the Arabian Sea. Kerala i
s known for its natural beauty, diverse landscapes ranging from sandy beaches to dense rainforests, as well as its rich cult
ural heritage and vibrant festivals.
```

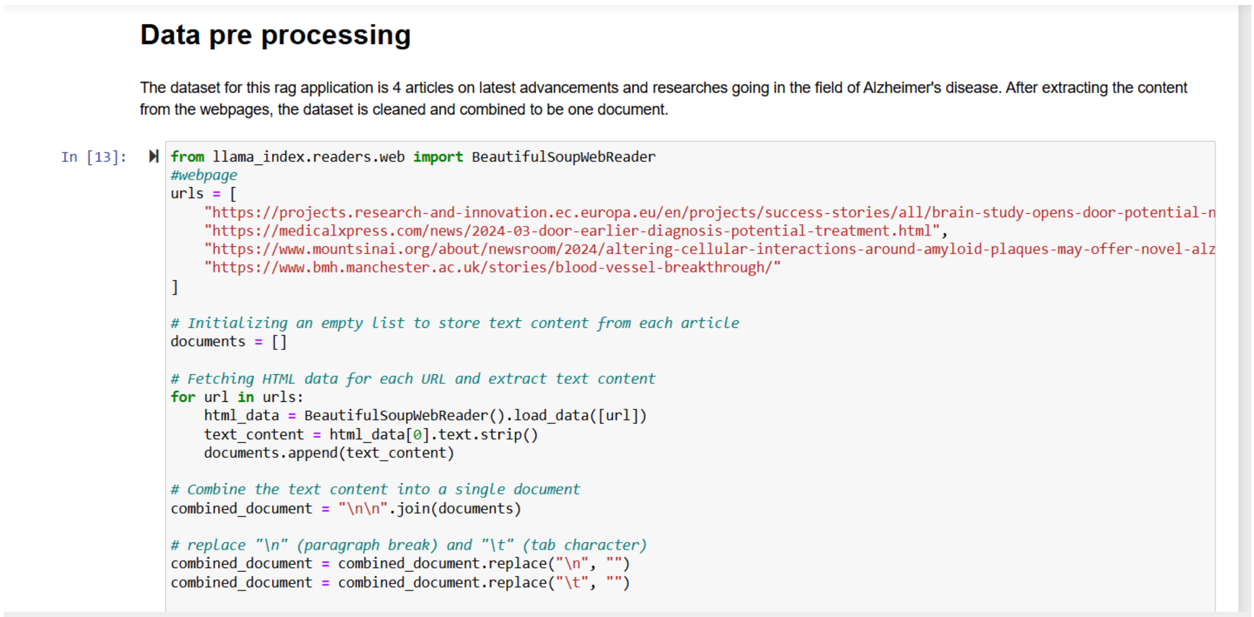
Installing necessary packages importing libraries.

In [3]:
```
# Install prerequisites
!pip install llama-index-embeddings-huggingface
!pip install llama-index-llms-ollama
!pip install llama-index-vector-stores-chroma
!pip install llama-index ipywidgets
!pip install llama-index-llms-huggingface
!pip install llama_index.readers.web
!pip install chromadb
```



```
# Import required modules from the llama_index library
from llama_index.core import VectorStoreIndex, SimpleDirectoryReader
from llama_index.embeddings.huggingface import HuggingFaceEmbedding
from llama_index.core import Settings
from llama_index.llms.ollama import Ollama
from llama_index.core import StorageContext

# Import ChromaVectorStore and chromadb module
from llama_index.vector_stores.chroma import ChromaVectorStore
import chromadb

# Import the Ollama class
from llama_index.llms.ollama import Ollama
```

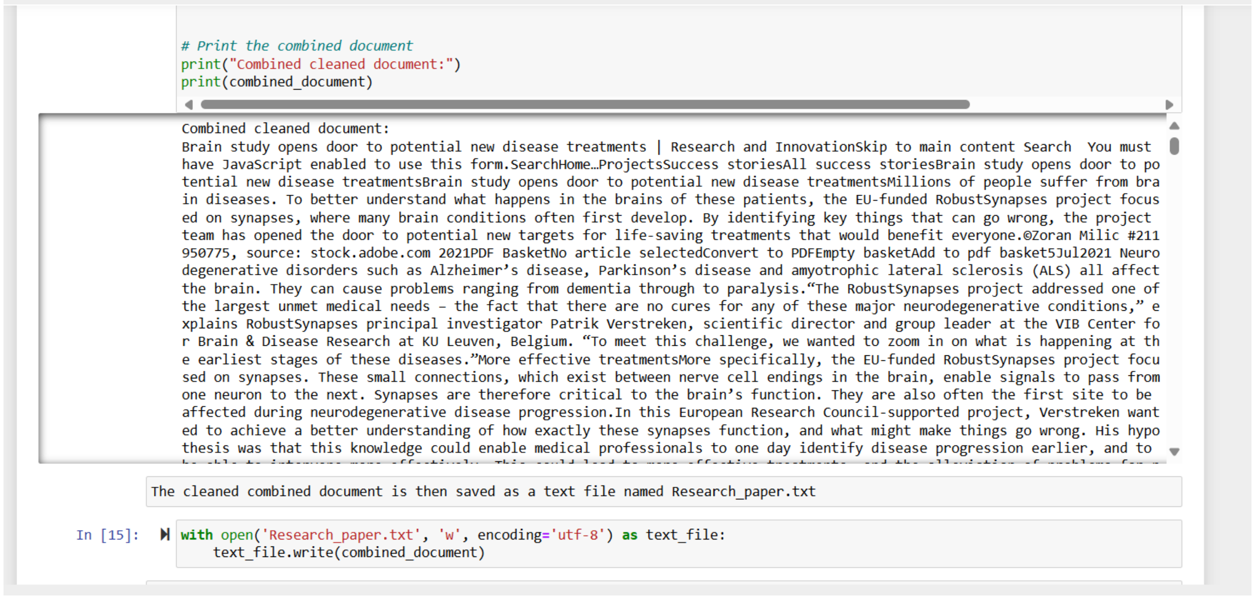
```
Requirement already satisfied: llama-index-embeddings-huggingface in c:\users\afafc\anaconda3\lib\site-packages (0.2.1)
Requirement already satisfied: huggingface-hub[inference]>=0.19.0 in c:\users\afafc\anaconda3\lib\site-packages (from llam
a-index-embeddings-huggingface) (0.23.3)
Requirement already satisfied: llama-index-core<0.11.0,>=0.10.1 in c:\users\afafc\anaconda3\lib\site-packages (from llama-
index-embeddings-huggingface) (0.10.43)
Requirement already satisfied: sentence-transformers<3.0.0,>=2.6.1 in c:\users\afafc\anaconda3\lib\site-packages (from lla
ma-index-embeddings-huggingface) (2.7.0)
Requirement already satisfied: filelock in c:\users\afafc\anaconda3\lib\site-packages (from huggingface-hub[inference]>=0.
19.0->llama-index-embeddings-huggingface) (3.9.0)
Requirement already satisfied: fsspec>=2023.5.0 in c:\users\afafc\anaconda3\lib\site-packages (from huggingface-hub[infere
nce]>=0.19.0->llama-index-embeddings-huggingface) (2024.5.0)
Requirement already satisfied: packaging>=20.9 in c:\users\afafc\anaconda3\lib\site-packages (from huggingface-hub[inferen
ce]>=0.19.0->llama-index-embeddings-huggingface) (23.2)
Requirement already satisfied: pyyaml>=5.1 in c:\users\afafc\anaconda3\lib\site-packages (from huggingface-hub[inference]>
=0.19.0->llama-index-embeddings-huggingface) (6.0.1)
Requirement already satisfied: requests in c:\users\afafc\anaconda3\lib\site-packages (from huggingface-hub[inference]>=0.
19.0->llama-index-embeddings-huggingface) (2.31.0)
Requirement already satisfied: tqdm>=4.42.1 in c:\users\afafc\anaconda3\lib\site-packages (from huggingface-hub[inference]
>=0.19.0->llama-index-embeddings-huggingface) (4.66.4)
```



Data pre processing

The dataset for this rag application is 4 articles on latest advancements and researches going in the field of Alzheimer's disease. After extracting the content from the webpages, the dataset is cleaned and combined to be one document.

```
In [13]:  from llama_index.readers.web import BeautifulSoupWebReader
          #webpage
          urls = [
              "https://projects.research-and-innovation.ec.europa.eu/en/projects/success-stories/all/brain-study-opens-door-potential-n
              "https://medicalxpress.com/news/2024-03-door-earlier-diagnosis-potential-treatment.html",
              "https://www.mountsinai.org/about/newsroom/2024/altering-cellular-interactions-around-amyloid-plaques-may-offer-novel-alz
              "https://www.bmh.manchester.ac.uk/stories/blood-vessel-breakthrough/"
          ]

          # Initializing an empty list to store text content from each article
          documents = []

          # Fetching HTML data for each URL and extract text content
          for url in urls:
              html_data = BeautifulSoupWebReader().load_data([url])
              text_content = html_data[0].text.strip()
              documents.append(text_content)

          # Combine the text content into a single document
          combined_document = "\n\n".join(documents)

          # replace "\n" (paragraph break) and "\t" (tab character)
          combined_document = combined_document.replace("\n", "")
          combined_document = combined_document.replace("\t", "")
```

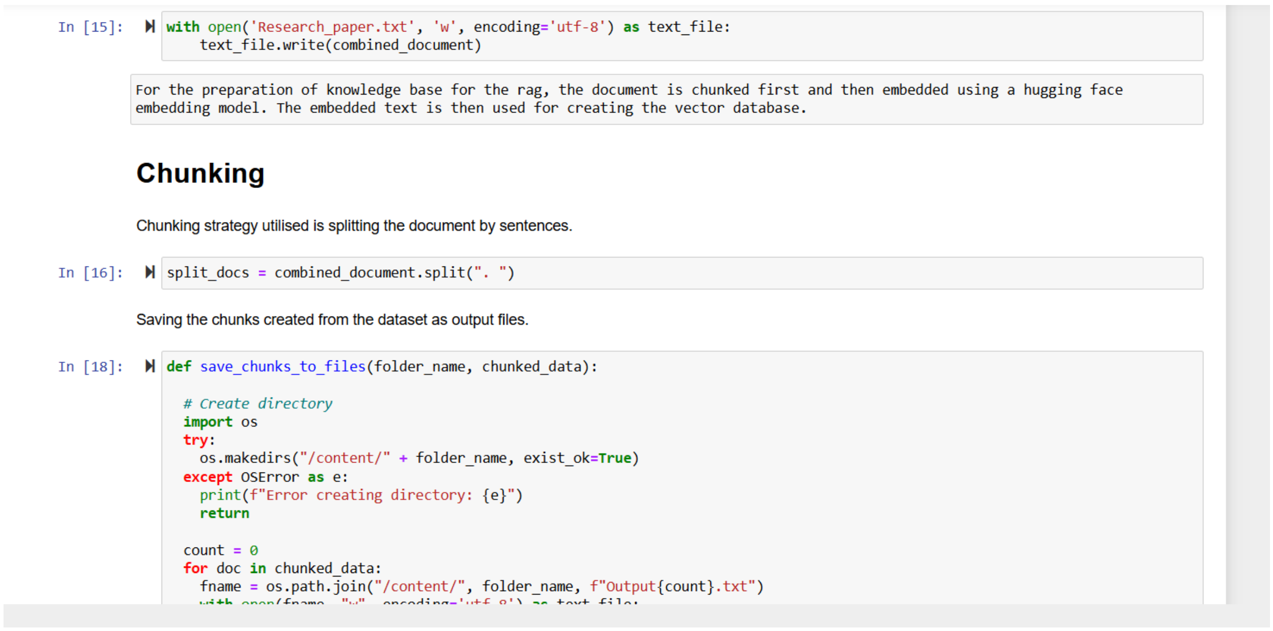


```
# Print the combined document
print("Combined cleaned document:")
print(combined_document)
```

```
Combined cleaned document:
Brain study opens door to potential new disease treatments | Research and InnovationSkip to main content Search  You must
have JavaScript enabled to use this form.SearchHome…ProjectsSuccess storiesAll success storiesBrain study opens door to po
tential new disease treatmentsBrain study opens door to potential new disease treatmentsMillions of people suffer from bra
in diseases. To better understand what happens in the brains of these patients, the EU-funded RobustSynapses project focus
ed on synapses, where many brain conditions often first develop. By identifying key things that can go wrong, the project
team has opened the door to potential new targets for life-saving treatments that would benefit everyone.©Zoran Milic #211
950775, source: stock.adobe.com 2021PDF BasketNo article selectedConvert to PDFEmpty basketAdd to pdf basket5Jul2021 Neuro
degenerative disorders such as Alzheimer's disease, Parkinson's disease and amyotrophic lateral sclerosis (ALS) all affect
the brain. They can cause problems ranging from dementia through to paralysis."The RobustSynapses project addressed one of
the largest unmet medical needs – the fact that there are no cures for any of these major neurodegenerative conditions," e
xplains RobustSynapses principal investigator Patrik Verstreken, scientific director and group leader at the VIB Center fo
r Brain & Disease Research at KU Leuven, Belgium. "To meet this challenge, we wanted to zoom in on what is happening at th
e earliest stages of these diseases."More effective treatmentsMore specifically, the EU-funded RobustSynapses project focu
sed on synapses. These small connections, which exist between nerve cell endings in the brain, enable signals to pass from
one neuron to the next. Synapses are therefore critical to the brain's function. They are also often the first site to be
affected during neurodegenerative disease progression.In this European Research Council-supported project, Verstreken want
ed to achieve a better understanding of how exactly these synapses function, and what might make things go wrong. His hypo
thesis was that this knowledge could enable medical professionals to one day identify disease progression earlier, and to
```

```
The cleaned combined document is then saved as a text file named Research_paper.txt
```

```
In [15]:  with open('Research_paper.txt', 'w', encoding='utf-8') as text_file:
              text_file.write(combined_document)
```



```
In [15]:  with open('Research_paper.txt', 'w', encoding='utf-8') as text_file:
              text_file.write(combined_document)
```

```
For the preparation of knowledge base for the rag, the document is chunked first and then embedded using a hugging face
embedding model. The embedded text is then used for creating the vector database.
```

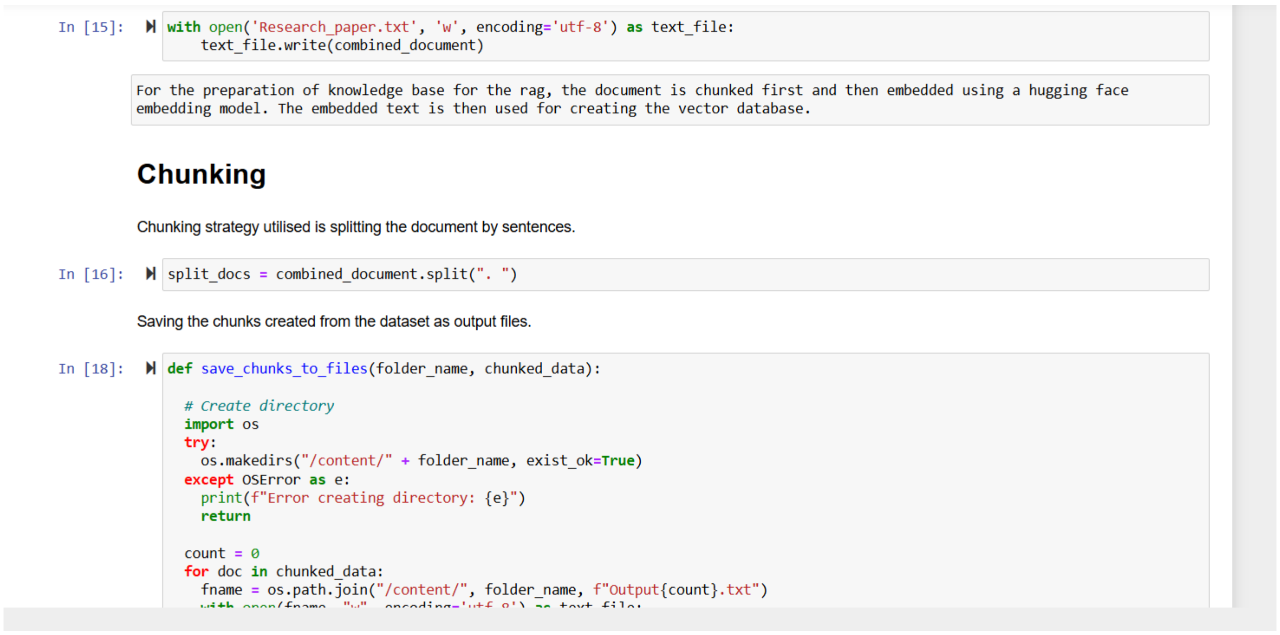
Chunking

Chunking strategy utilised is splitting the document by sentences.

```
In [16]:  split_docs = combined_document.split(". ")
```

Saving the chunks created from the dataset as output files.

```
In [18]:  def save_chunks_to_files(folder_name, chunked_data):

            # Create directory
            import os
            try:
              os.makedirs("/content/" + folder_name, exist_ok=True)
            except OSError as e:
              print(f"Error creating directory: {e}")
              return

            count = 0
            for doc in chunked_data:
              fname = os.path.join("/content/", folder_name, f"Output{count}.txt")
              with open(fname, "w", encoding='utf-8') as text_file:
```



```
In [15]:  with open('Research_paper.txt', 'w', encoding='utf-8') as text_file:
              text_file.write(combined_document)
```

```
For the preparation of knowledge base for the rag, the document is chunked first and then embedded using a hugging face
embedding model. The embedded text is then used for creating the vector database.
```

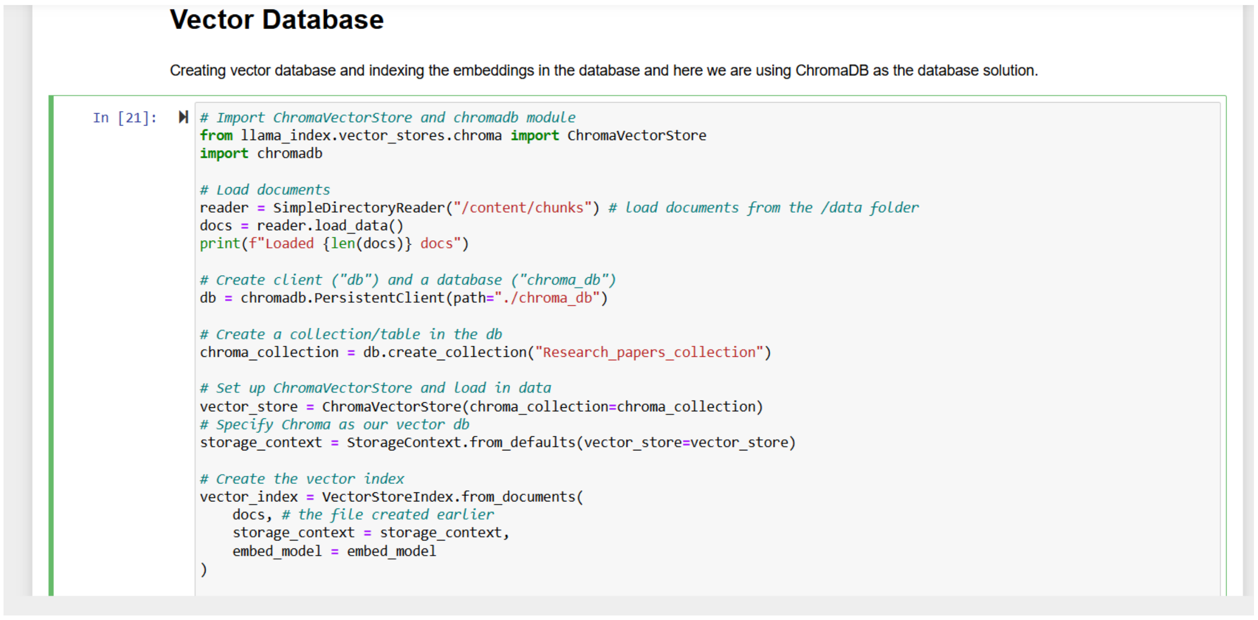
Chunking

Chunking strategy utilised is splitting the document by sentences.

```
In [16]:  split_docs = combined_document.split(". ")
```

Saving the chunks created from the dataset as output files.

```
In [18]:  def save_chunks_to_files(folder_name, chunked_data):

            # Create directory
            import os
            try:
              os.makedirs("/content/" + folder_name, exist_ok=True)
            except OSError as e:
              print(f"Error creating directory: {e}")
              return

            count = 0
            for doc in chunked_data:
              fname = os.path.join("/content/", folder_name, f"Output{count}.txt")
              with open(fname, "w", encoding='utf-8') as text_file:
```

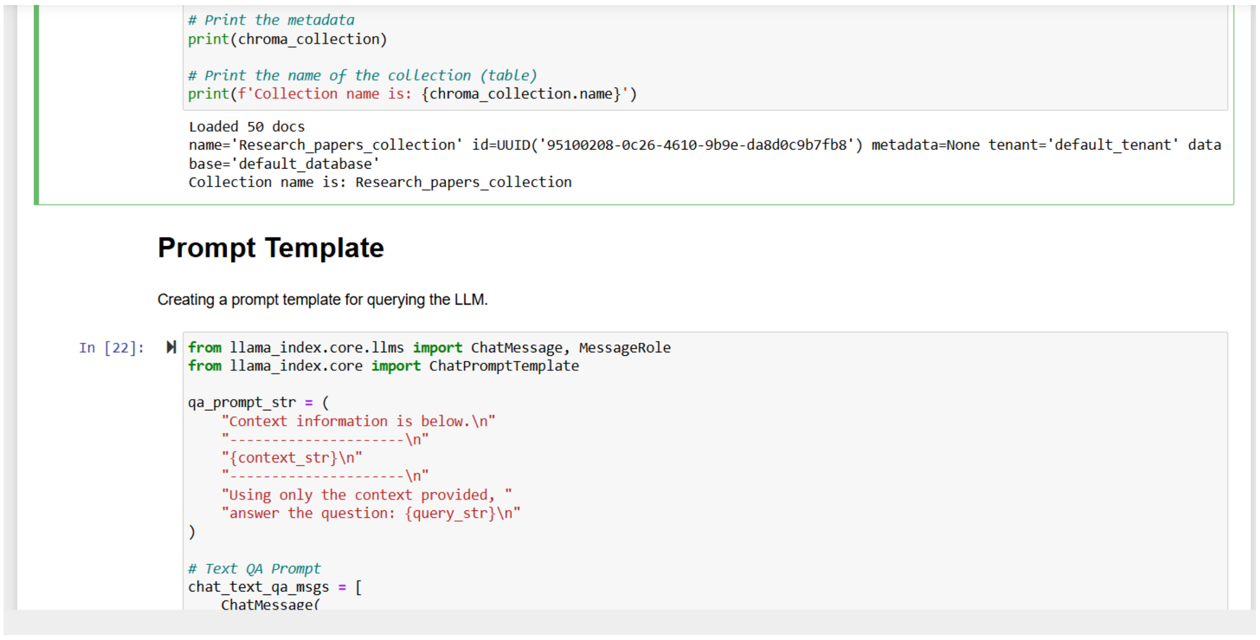


Vector Database

Creating vector database and indexing the embeddings in the database and here we are using ChromaDB as the database solution.

In [21]:

```python
# Import ChromaVectorStore and chromadb module
from llama_index.vector_stores.chroma import ChromaVectorStore
import chromadb

# Load documents
reader = SimpleDirectoryReader("/content/chunks") # load documents from the /data folder
docs = reader.load_data()
print(f"Loaded {len(docs)} docs")

# Create client ("db") and a database ("chroma_db")
db = chromadb.PersistentClient(path="./chroma_db")

# Create a collection/table in the db
chroma_collection = db.create_collection("Research_papers_collection")

# Set up ChromaVectorStore and load in data
vector_store = ChromaVectorStore(chroma_collection=chroma_collection)
# Specify Chroma as our vector db
storage_context = StorageContext.from_defaults(vector_store=vector_store)

# Create the vector index
vector_index = VectorStoreIndex.from_documents(
    docs, # the file created earlier
    storage_context = storage_context,
    embed_model = embed_model
)
```



```python
# Print the metadata
print(chroma_collection)

# Print the name of the collection (table)
print(f'Collection name is: {chroma_collection.name}')
```

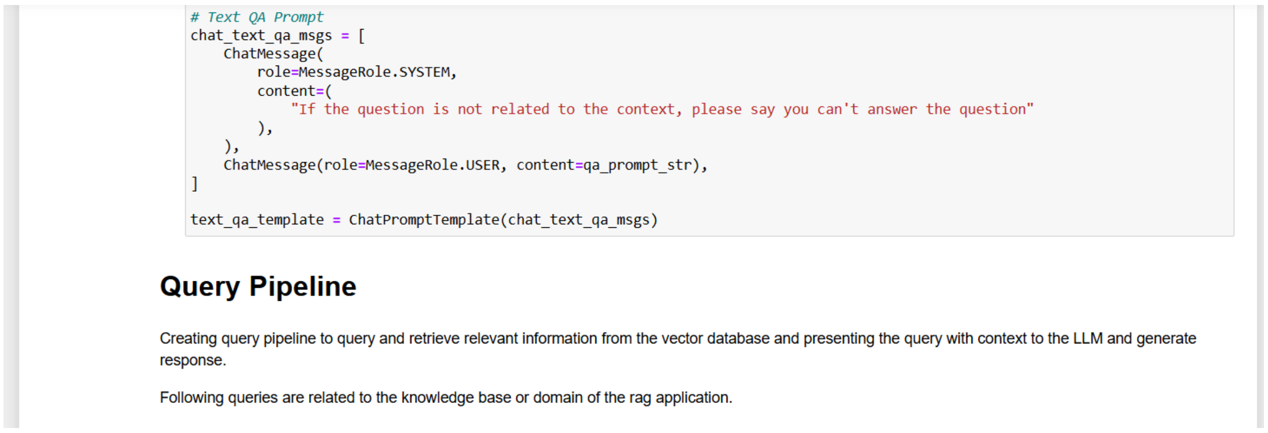
```
Loaded 50 docs
name='Research_papers_collection' id=UUID('95100208-0c26-4610-9b9e-da8d0c9b7fb8') metadata=None tenant='default_tenant' data
base='default_database'
Collection name is: Research_papers_collection
```

Prompt Template

Creating a prompt template for querying the LLM.

In [22]:

```python
from llama_index.core.llms import ChatMessage, MessageRole
from llama_index.core import ChatPromptTemplate

qa_prompt_str = (
    "Context information is below.\n"
    "---------------------\n"
    "{context_str}\n"
    "---------------------\n"
    "Using only the context provided, "
    "answer the question: {query_str}\n"
)

# Text QA Prompt
chat_text_qa_msgs = [
    ChatMessage(
```

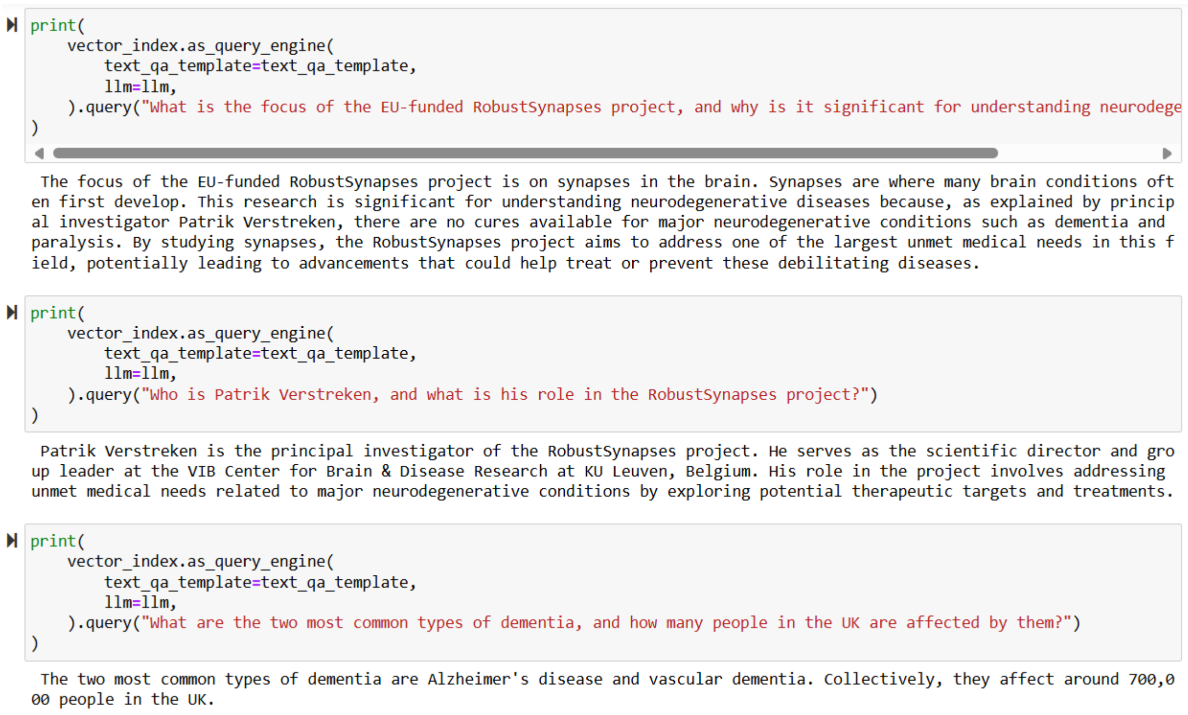


```python
# Text QA Prompt
chat_text_qa_msgs = [
    ChatMessage(
        role=MessageRole.SYSTEM,
        content=(
            "If the question is not related to the context, please say you can't answer the question"
        ),
    ),
    ChatMessage(role=MessageRole.USER, content=qa_prompt_str),
]

text_qa_template = ChatPromptTemplate(chat_text_qa_msgs)
```

Query Pipeline

Creating query pipeline to query and retrieve relevant information from the vector database and presenting the query with context to the LLM and generate response.

Following queries are related to the knowledge base or domain of the rag application.



```
print(
    vector_index.as_query_engine(
        text_qa_template=text_qa_template,
        llm=llm,
    ).query("What is the focus of the EU-funded RobustSynapses project, and why is it significant for understanding neurodege
)
```

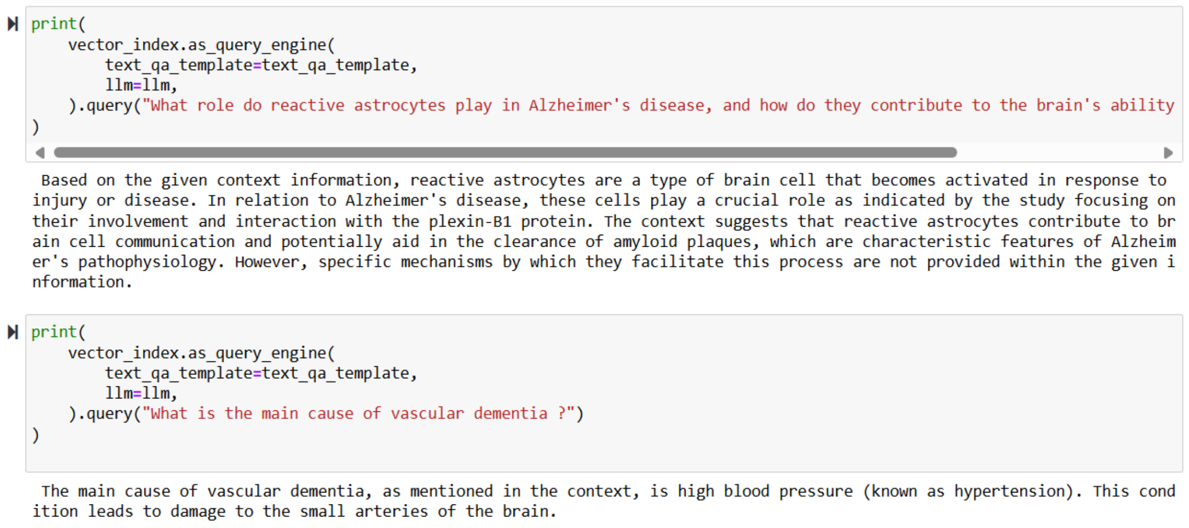
```
 The focus of the EU-funded RobustSynapses project is on synapses in the brain. Synapses are where many brain conditions oft
en first develop. This research is significant for understanding neurodegenerative diseases because, as explained by princip
al investigator Patrik Verstreken, there are no cures available for major neurodegenerative conditions such as dementia and
paralysis. By studying synapses, the RobustSynapses project aims to address one of the largest unmet medical needs in this f
ield, potentially leading to advancements that could help treat or prevent these debilitating diseases.
```

```
print(
    vector_index.as_query_engine(
        text_qa_template=text_qa_template,
        llm=llm,
    ).query("Who is Patrik Verstreken, and what is his role in the RobustSynapses project?")
)
```

```
 Patrik Verstreken is the principal investigator of the RobustSynapses project. He serves as the scientific director and gro
up leader at the VIB Center for Brain & Disease Research at KU Leuven, Belgium. His role in the project involves addressing
unmet medical needs related to major neurodegenerative conditions by exploring potential therapeutic targets and treatments.
```

```
print(
    vector_index.as_query_engine(
        text_qa_template=text_qa_template,
        llm=llm,
    ).query("What are the two most common types of dementia, and how many people in the UK are affected by them?")
)
```

```
 The two most common types of dementia are Alzheimer's disease and vascular dementia. Collectively, they affect around 700,0
00 people in the UK.
```

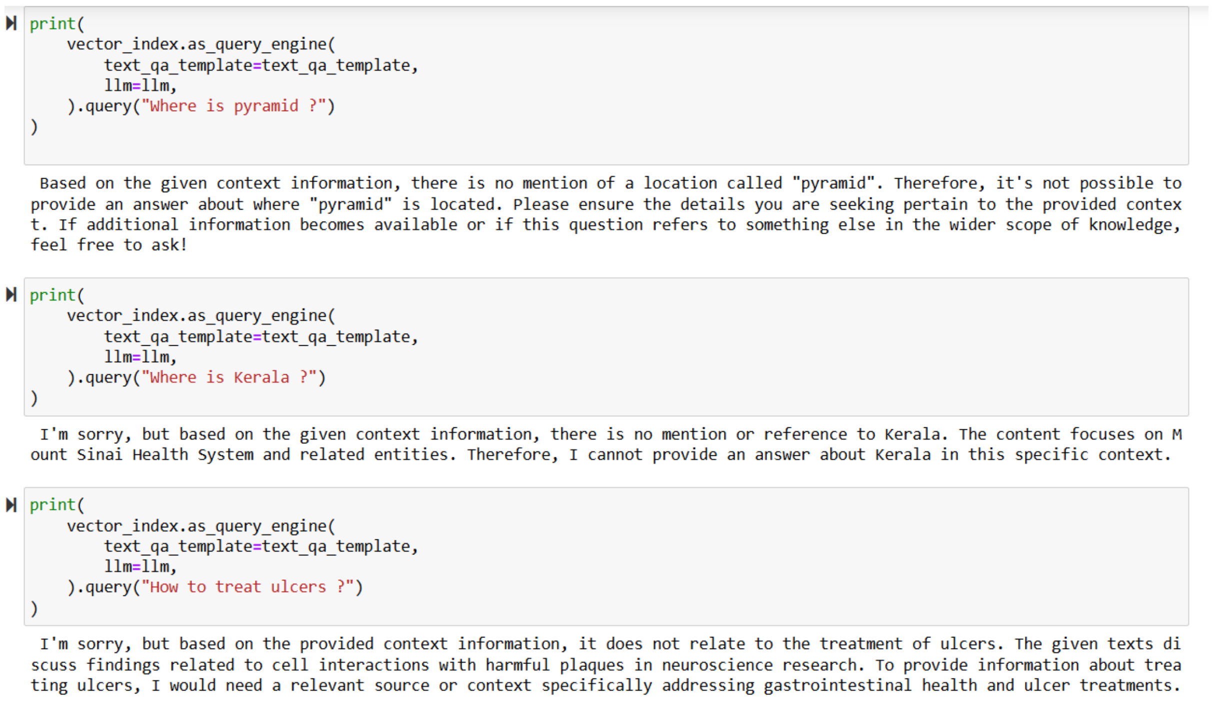


```
print(
    vector_index.as_query_engine(
        text_qa_template=text_qa_template,
        llm=llm,
    ).query("What role do reactive astrocytes play in Alzheimer's disease, and how do they contribute to the brain's ability
)
```

```
 Based on the given context information, reactive astrocytes are a type of brain cell that becomes activated in response to
injury or disease. In relation to Alzheimer's disease, these cells play a crucial role as indicated by the study focusing on
their involvement and interaction with the plexin-B1 protein. The context suggests that reactive astrocytes contribute to br
ain cell communication and potentially aid in the clearance of amyloid plaques, which are characteristic features of Alzheim
er's pathophysiology. However, specific mechanisms by which they facilitate this process are not provided within the given i
nformation.
```

```
print(
    vector_index.as_query_engine(
        text_qa_template=text_qa_template,
        llm=llm,
    ).query("What is the main cause of vascular dementia ?")
)
```

```
 The main cause of vascular dementia, as mentioned in the context, is high blood pressure (known as hypertension). This cond
ition leads to damage to the small arteries of the brain.
```



```
print(
    vector_index.as_query_engine(
        text_qa_template=text_qa_template,
        llm=llm,
    ).query("Where is pyramid ?")
)
```

```
 Based on the given context information, there is no mention of a location called "pyramid". Therefore, it's not possible to
provide an answer about where "pyramid" is located. Please ensure the details you are seeking pertain to the provided contex
t. If additional information becomes available or if this question refers to something else in the wider scope of knowledge,
feel free to ask!
```

```
print(
    vector_index.as_query_engine(
        text_qa_template=text_qa_template,
        llm=llm,
    ).query("Where is Kerala ?")
)
```

```
 I'm sorry, but based on the given context information, there is no mention or reference to Kerala. The content focuses on M
ount Sinai Health System and related entities. Therefore, I cannot provide an answer about Kerala in this specific context.
```

```
print(
    vector_index.as_query_engine(
        text_qa_template=text_qa_template,
        llm=llm,
    ).query("How to treat ulcers ?")
)
```

```
 I'm sorry, but based on the provided context information, it does not relate to the treatment of ulcers. The given texts di
scuss findings related to cell interactions with harmful plaques in neuroscience research. To provide information about trea
ting ulcers, I would need a relevant source or context specifically addressing gastrointestinal health and ulcer treatments.
```