# Price Researcher Optimised for Friends (PROF)

## 39th Meeting of the Voorburg Group on Service Statistics

## September 2025

**Abstract**

The Singapore Department of Statistics (DOS) developed a prototype Retrieval-Augmented Generation (RAG) chatbot powered by a large language model (LLM). This chatbot, known as Price Researcher Optimised for Friends (PROF), envisions to be the first stop application for DOS's price statisticians to discover and digest technical papers from the Voorburg Group website.

This paper shares DOS's experience in developing PROF, technical details of the chatbot, challenges faced and upcoming plans to further improve the application.

**Team Members from DOS:**

Chan Zi Qi, Prices Division (chan_zi_qi@singstat.gov.sg)
Tan Bing Xiang, Prices Division (tan_bing_xiang@singstat.gov.sg)
Edwin Boey, Prices Division (edwin_boey@singstat.gov.sg)

# Contents


**Background** ..... 3

**Problem Statement** ..... 3

**Introduction to RAG Chatbots Powered by LLM** ..... 4

**Technical Description of PROF** ..... 5

i) **Preparation of Knowledge Base** ..... 6

ii) **Processing User Queries using Prompt Engineering** ..... 7

iii) **Stacking Open Source Technology** ..... 8

**Evaluation of PROF** ..... 9

**Addressing Challenges of RAG-LLM Applications** ..... 10

**Phase 2 Enhancement: Charting Features** ..... 10

**Learning Points and Future Plans** ..... 11

**Appendix 1** ..... 13

## Background

The Voorburg Group on Service Statistics[1] endeavours to establish and maintain an internationally comparable methodology for measuring output and producer and import price indices for service industries. The Voorburg Group has contributed over the years to building up and sharing a considerable and growing body of knowledge on Service Sector Statistics. It has prompted international cooperation in the development of standards and has assisted in resolving statistical and measurement challenges in the Service Sector. The Group maintains a permanent website, currently hosted by Statistics Canada, on which all of the Group's outputs can be found.

Price Statisticians from the Singapore Department of Statistics (DOS) refer to the papers residing on the website regularly for research purposes. Given the rapid and widespread adoption of Generative AI (GenAI) [2] applications, a team was formed to develop a prototype Retrieval-Augmented Generation (RAG) chatbot powered by a large language model (LLM) to **improve the discovery process** and **enhance the research experience**.

## Problem Statement

Firstly, the simple search and sorting function on the website makes it challenging to locate all relevant documents. For example, if the user was conducting research on the cleaning service industry (Exhibit 1), the keyword search function would only return materials with the search term "cleaning" within its title. It would leave out papers that discuss cleaning services but do not have the word "cleaning" in the Title.

Exhibit 1: Materials Hosted on the Voorburg Group Website

**Voorburg Group on Service Statistics Papers (2005-present)**

Filter items: cleaning — Showing 1 to 10 of 25 entries (filtered from 1,719 total entries) | Show 10 entries

| Year | Location | Format | Title | ISIC Section | 2 digit ISIC Division | Author | Type | Theme | Topic |
|---|---|---|---|---|---|---|---|---|---|
| 2015 | Sydney | Paper | Using administrative data for the Italian Cleaning Services SPPI (paper) | N - Administrative and support service activities | 81 - Services to buildings and landscape activities | Sola, Giuseppina | Country Industry | Prices | Administrative / Alternative Data |
| 2015 | Sydney | Paper | Using administrative data for the Italian Cleaning Services SPPI (paper) | N - Administrative and support service activities | 81 - Services to buildings and landscape activities | Cecconi, Cristina | Country Industry | Prices | Administrative / Alternative Data |
| 2015 | Sydney | Paper | Using administrative data for the Italian Cleaning Services SPPI (paper) | N - Administrative and support service activities | 81 - Services to buildings and landscape activities | Brogi, Federico | Country Industry | Prices | Administrative / Alternative Data |
| 2015 | Sydney | Presentation | Using administrative data for the Italian Cleaning SPPI (ppp) | N - Administrative and support service activities | 81 - Services to buildings and landscape activities | Sola, Giuseppina | Country Industry | Prices | Administrative / Alternative Data |
| 2015 | Sydney | Presentation | Using administrative data for the Italian Cleaning SPPI (ppp) | N - Administrative and support service activities | 81 - Services to buildings and landscape activities | Cecconi, Cristina | Country Industry | Prices | Administrative / Alternative Data |

Secondly, after identifying the relevant documents, much time is spent having to read and digest the documents. This can be time consuming for officers who may only have specific questions to find out and reply to their policy users or management.

[1] United Nations Statistics Division, "Voorburg Group on Service Statistics".

[2] GenAI: a type of AI that can create new content based on existing data

Thirdly, for officers who are new to service statistics, having to search and read through many papers with technical jargons may be overwhelming. An application to guide them through this search and discovery process would be helpful.

Hence in 2024, the team from DOS participated in a government organised 'AI Champions Bootcamp', which aimed to train participants to develop their own Proof-of-Concept (PoC) LLM applications to address their own agencies' use cases. With the problem in mind, the team underwent 3 months of intensive training to create a LLM RAG chatbot that could ingest a large amount of technical academic papers from the Voorburg Group meetings and return easily understood output to users. Since then, the team has improved the performance and efficiency of the chatbot over many iterations to come up with a pilot prototype known as Price Researcher Optimised for Friends (PROF).

## Introduction to RAG Chatbots Powered by LLM

LLMs are a type of GenAI technology. They are models that are trained on huge amounts of data and can extract meaning from a sequence of text and understand the relationships between words and phrases in it. This allows LLMs to be able to not only assess text but also generate original content based on user input[3]. As such, chatbots are a common application of LLMs. An LLM chatbot typically follows the linear flow seen in Exhibit 2 below. When a user submits a query to the chatbot, the system converts it into a structured prompt that the LLM can understand. The prompt may include additional context, instructions and formatting to guide the LLM in generating a desired response. The LLM then processes the prompt and generates a response by predicting the most appropriate sequence of words.

*Exhibit 2: A typical LLM chatbot process flow*


However, the chatbot is limited only to the knowledge that the LLM has been trained on. As such, the RAG framework has been developed to augment an LLM chatbot that is required to generate responses based on context outside the scope of its original training data without having to retrain the entire model. The RAG framework extends the capabilities of the LLM chatbot by redirecting the LLM chatbot to retrieve information from a separate knowledge base[4].

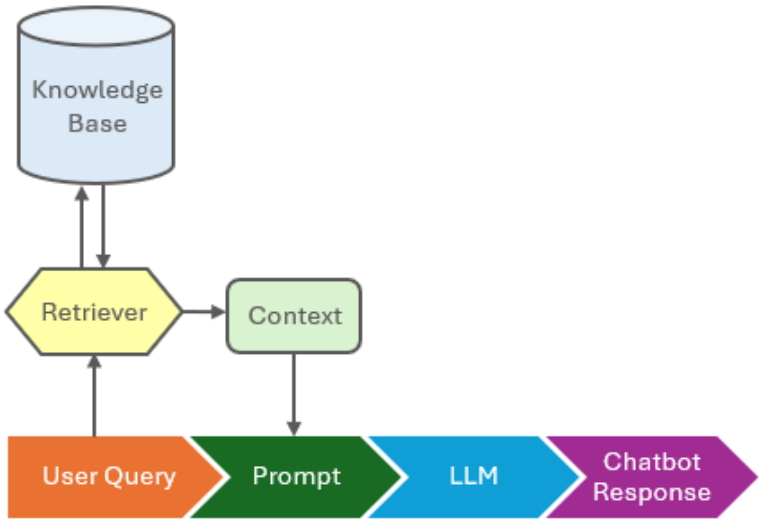
Exhibit 3 below illustrates this updated process. After a user submits a query to the chatbot, the query is passed on to a document retriever which is connected to the external knowledge base. Using the query as input, the retriever scours the knowledge base for the most relevant information (termed as "Context"), which is provided back to the LLM with the query as part of the prompt. The LLM is then able to generate a contextually relevant response. The team applied this approach in developing the chatbot, with the **materials from the Voorburg Group website as the 'Knowledge Base'**, distinguishing itself from other general LLM applications.

---

[3] European Commission Knowledge Centre on Translation and Interpretation, "What is a Large Language Model?"

[4] Amazon Web Services, "What is RAG (Retrieval-Augmented Generation)?"

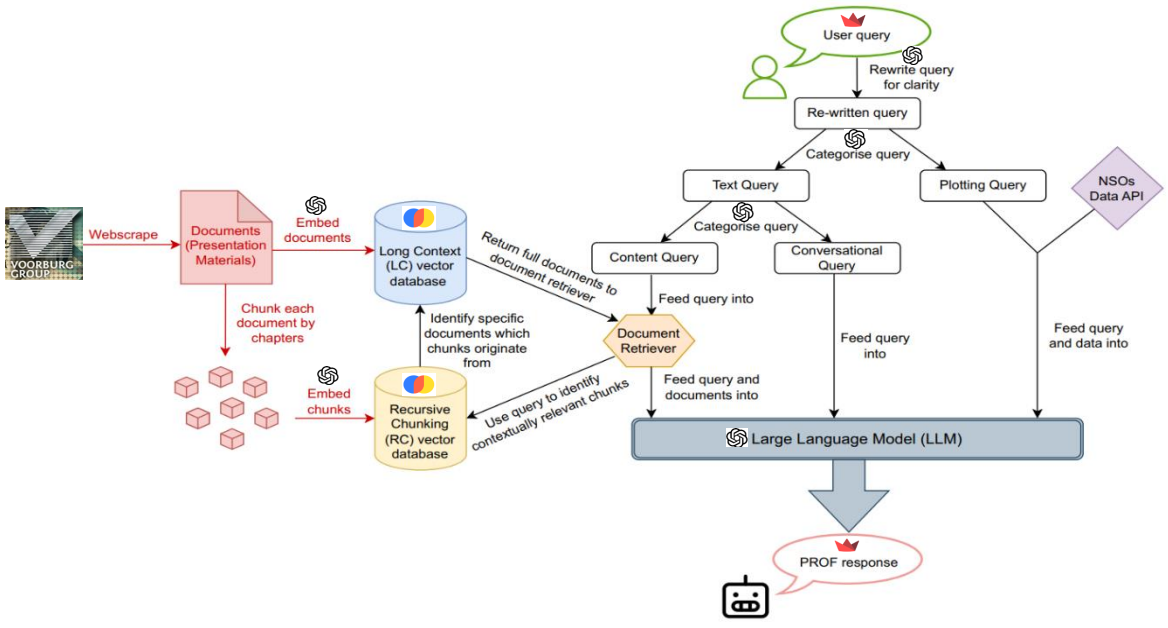
Exhibit 3: A typical LLM RAG chatbot process flow




## Technical Description of PROF

Exhibit 4 below illustrates the detailed architecture of PROF (a larger version of the architecture is provided in Appendix I at the end of the paper). The development of PROF will be elaborated in three parts, namely, (i) Preparing knowledge base, (ii) Processing user queries and (iii) Stacking open source technology.

Exhibit 4: PROF Architecture



### i) Preparation of Knowledge Base

a. Extraction of documents from Voorburg Group website

The Voorburg Group website contains more than 3000 documents comprising papers and presentations dating back to 1987. For prototype purposes, the team web-scraped and downloaded all the documents from 2005, which consisted of PDF and PowerPoint files made up of both text and image content in the form of charts, diagrams and tables. A vision language model, also an LLM, was used to extract and interpret information contained in images into text. This allowed each document to be converted fully into text content.

This is a one-off effort that was done for documents dating up to 2024. In future, whenever the document repository on the Voorburg Group website is updated with new materials, the knowledge base will also need to be updated.

b. Chunking of documents

Chunking describes the process of splitting large datasets into smaller, manageable segments[5]. It controls the amount of information an LLM receives to reduce time and cost while improving relevancy of document retrieval.

For PROF, several chunking methods were compared before arriving at the final strategy. The table below lists out the different chunking strategies tested out and the performance of each one when asked the same question.

| **Strategy** | **Description** | **Cost** | **Time taken (s)** | **Response quality** |
|---|---|---|---|---|
| Recursive | Divides text using a separating identifier (e.g. \n) iteratively. | $ | 26 | Low |
| Semantic | Extract meaning from text embeddings, compare the semantic relationship between the text, and group chunks that have similar meaning. | $$ | 76 | Low |
| Agentic | Using LLMs to categorise text, and putting text in similar categories into chunks. | $$$$ | 24 | Medium |
| Long Context | Whole or large portions of documents in each chunk. | $ | 12 | Medium |
| Recursive + Long Context | Hybrid | $ | 25 | High |
| Agentic + Long Context Hybrid | Hybrid | $$$$ | 32 | High |

Recursive chunking with long context was found to strike the best balance between cost, time taken and response quality for PROF's use case. This hybrid approach required the creation of two separate databases:

1. Recursive chunking (RC) database: contains the same documents but recursively split into chapters (i.e. each chunk corresponds to one chapter)

[5] Pinecone, "Chunking Strategies for LLM Applications"

2. Long context (LC) database: contains the documents in their entirety

In practice, the user query will be directed to the RC database first, where the most relevant chunk or chapter that can answer the query is identified. The entire original document which the chapter belonged to is then located in the LC database and used as context for PROF to generate a response. In essence, the system narrows down relevant content using smaller chunks, maintaining semantic integrity which is lost with larger chunks, then enriches the chunks with the full document context, retaining long-range coherence.

c. Embedding of documents

Text data from the chunking process was converted into a format that can be efficiently used for document retrieval. The text content was embedded using LLM and stored in vector databases. Embeddings are numerical vector representations of text that capture semantic meaning, which allow for processing of relationships between two different pieces of text efficiently[6]. This means that vector databases can easily be searched for relevant documents given a keyword or a query.

### ii) Processing User Queries using Prompt Engineering

Prompt engineering is the art and science of crafting effective instructions for LLMs to produce desired outputs, bridging the gap between human intent and machine understanding[7]. When a user types a query, the set of instructions given to the LLM will help to provide context and direction to the LLM to guide its response. To optimise PROF's performance, the following prompt engineering techniques listed below were applied:

| Technique | Description | Sample Excerpt from PROF's Prompt |
|---|---|---|
| Role setting | Instructing the LLM to assume a specific role related to the task at hand | "Your task is to analyse user queries, categorise them as either 'Plotting/Data Analytics' or 'Other questions', and process them accordingly to provide helpful responses based on multiple document sources." |
| Chain-of-thought reasoning | Guides the LLM to process thoughts step-by-step to reduce errors and improve response quality | "To process this task, follow these steps:<br><br>1. Carefully read the data provided.<br>2. Analyse the query<br>3. If the query asks to plot or display data, you should return the data from the JSON, prefixed with 'Plot '.<br>4. If the query asks a specific question about the data, you should provide a direct answer based on the information in the JSON string." |
| XML tags | Introduces clear structure to the LLM by defining placeholders for dynamic | "Extract relevant keywords following the guidelines below |

[6] Couchbase, "A Guide to LLM Embeddings"
[7] What is Prompt Engineering - Meaning, Working, Techniques - GeeksforGeeks

| | | |
|---|---|---|
| | data, helping LLM distinguish between static instructions, variable input and expected output. | Format as: <query>User's question</query> followed by <keywords>extracted keywords</keywords><br>Provide a comprehensive answer based on the retrieved document chunks" |

These prompt techniques allow LLM to be used to carry out the following query processing steps:

a. Rewrite query

At this step, the LLM is instructed to rewrite the query as a standalone question given the chat history, to make it clearer for further processing.

b. Categorise query

Using the re-written query from before, the LLM is instructed to decide whether the query is a Text Query or a Plotting Query. If it is a Text Query, the LLM will further decide if the query requires reference to Voorburg Group materials to answer (i.e. a Content Query), or if it is a Conversational Query.

c. Identify keywords and collate relevant external data

For a Content Query, the LLM will identify keywords that will inform about topic or industry from the query, e.g. the query "What do you know about legal services producer price indices?" would have the keywords "legal services" and "producer price indices" associated with it. The keywords are then used to identify relevant documents in the vector databases that can provide rich context for answering the query sufficiently.

For a Conversational Query, the query is not processed further at this stage as no additional context is required to generate a response.

For a Plotting Query, the LLM will identify keywords that will inform about the frequency of data, whether a chart or raw data is requested, and the country of interest. From there, appropriate data will be extracted from the relevant country or countries' API and used as context to answer the Plotting Query.

d. Generate a response

The queries and context (if any) from the previous step are provided to the LLM again, which will use this refined input to generate a balanced, informative response to the query.

**iii) Stacking Open Source Technology**

During the development of PROF, the team considered and experimented with several open stack technologies. The following technologies listed below were used to build and run PROF mainly due to the team's knowledge and tools' reputation and scalability:

| **Technology** | **Benefits** |
|---|---|
| Programming language: Python | • Open source<br>• Team is comfortable with it<br>• Extensive libraries<br>• Integrates easily with most AI models |

| | |
|---|---|
| Frontend user interface: Streamlit | • Open source<br>• Beginner-friendly<br>• Built-in packages for chatbot interface<br>• Integrates well with Python applications |
| LLM: OpenAI<br>*Used for embedding of data and query processing | • Large token limit<br>• Easy to use<br>• Integrates well into prompt processing pipeline |
| Vector database: ChromaDB | • Open source<br>• Efficient document search<br>• Scalability |
| Hosting infrastructure: Amazon Elastic Compute 2 (EC2) | • Cost-effective<br>• Scalability and flexibility |

## Evaluation of PROF

Apart from user testing and feedback, PROF's performance was also evaluated using LLM-as-a-judge, which means that a different LLM is used to evaluate PROF's performance. This is becoming increasingly common in chatbot evaluation as it allows for evaluation of many responses more quickly, cheaply and consistently compared to manual (human) evaluation[8].

The team collated an extensive list of possible questions that could be answered by PROF and evaluated the responses using the DeepEval Python library which is recognised as one of the leading frameworks for LLM-as-a-judge[9]. For each selected evaluation metric, LLM-as-a-judge calculated a score and the median of all the scores was taken.

The following evaluation metrics were considered:

i) Answer Relevancy

Evaluates how relevant PROF's output is, relative to the original query. The metric can be represented conceptually by the below formula:

$$Answer\ Relevancy\ = \frac{Number\ of\ Relevant\ Statements\ in\ PROF\ output}{Total\ Number\ of\ Statements\ in\ PROF\ output}$$

ii) Faithfulness

Quantifies how accurate PROF's output is, relative to the retrieved context. This can also be viewed as quantifying PROF's hallucination rate, which is the propensity that PROF generates nonsensical content[10]. The metric can be represented conceptually by the below formula:

$$Faithfulness\ = \frac{Number\ of\ Truthful\ Claims\ in\ PROF\ output}{Total\ Number\ of\ Claims\ in\ PROF\ output}$$

The scores of each metric are summarised below.

[8] Towards Data Science, "LLM-as-a-Judge: A Practical Guide"
[9] Felix Dobslaw, "Challenges in Testing Large Language Model Based Software: A Faceted Taxonomy"
[10] Lakera, "The Beginner's Guide to Hallucinations in Large Language Models"

| Evaluation Metric | Median Score | Interpretation |
|---|---|---|
| Answer Relevancy | 0.8 | PROF output is mostly able to answer the user query |
| Faithfulness | 1.0 | PROF's output is highly unlikely to contain hallucinations |

Even though evaluation metrics provide a way to quantify performance, the team is still mindful to use it in conjunction with actual user feedback. More evaluation metrics can be considered in future to supplement user feedback, painting a more holistic picture of PROF's performance as we continue to enhance and develop PROF according to users' needs.

## Addressing Challenges of RAG-LLM Applications

While developing PROF, the team encountered challenges related to common risks faced by all LLMs. Most of these challenges can be mitigated by simple instructions within the prompt.

i) Hallucinations

This could occur when the LLM generates false information, including citing non-existent sources, when it does not have the available information. This risk is mitigated by preventive prompt techniques, i.e. by instructing the LLM via the prompt to forbid such actions. For example, to include the following snippets in the prompt:

- "Validate and format URLs properly."
- "If context is missing or incomplete, state: 'I'm sorry, but I don't have enough information to answer that question accurately."

ii) Bias

This may occur when the LLM generates content that propagates harmful stereotypes due to the inherently biased data it is trained on. Similar to dealing with hallucinations, the LLM can be instructed to check its response for harmful stereotypes before providing the response to the user.

iii) Prompt hacking

This could occur when the LLM is misused, such as if malicious users attempt to use PROF for unintended purposes. Snippets such as the below can be included in the prompt to prevent this:

"If you detect a potentially malicious prompt, respond with 'I'm sorry, but I can't process that request.'"

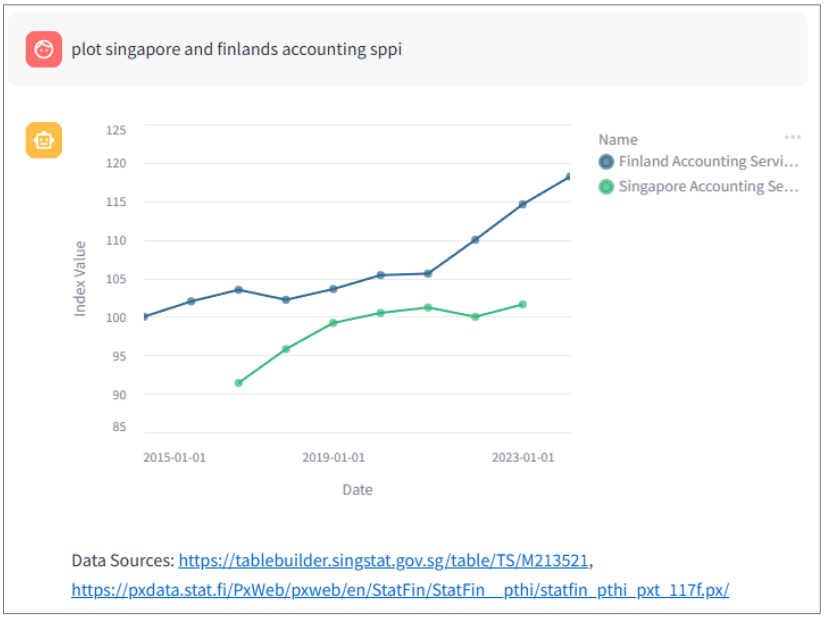
## Phase 2 Enhancement: Charting Features

PROF was initially developed with only the text query module, but the team recognised the potential in enhancing PROF's abilities to include plotting and analysing live data from different countries, which would be useful for statisticians to benchmark their price indices with their

international counterparts. This would make PROF an even more powerful tool which can combine insights not just from written papers but also compare price trends.

Published statistics from DOS are accessible via the SingStat Table Builder[11]. For a start, the Services Producer Price Indices (SPPIs) were mapped in PROF and can be retrieved via APIs and plotted as charts in the chatbot. As an example, the team also adopted a similar approach to map Finland's SPPIs via API subsequently to illustrate the use case (Exhibit 5).

Exhibit 5: PROF Charting Features




However, as different countries have different naming and labelling conventions for their SPPIs, much effort will be required to map common SPPIs across the other countries. Given more time and resources, other NSOs' APIs can gradually be included as data sources for retrieval and charting in PROF.

## Learning Points and Future Plans

As price statisticians, the team was new to GenAI and had to quickly gain knowledge as well as rely on advice from experts when they participated in the AI Bootcamp for 3 months. A lot of resources was spent to further build expertise inhouse and improve PROF subsequently within the team. Advances in technology and LLM capabilities provided both opportunities and obstacles to the learning process. Newer general LLM models (e.g., ChatGPT-5) which are smarter and faster may even produce better outputs than PROF, in relation to queries on service statistics (or SPPIs in particular). On the other hand, these newer LLM models can allow PROF to experiment further and better meet user needs, especially as the wealth of knowledge increases with the yearly upload of more documents. PROF is still valuable so long as specialised chatbots continue to outperform general-purpose LLMs in their respective fields.

On future plans, DOS intends to focus on phase 2 enhancements. Furthermore, the experience in developing a chatbot specialised in Service Statistics content opens the possibility of

[11] Singapore Department of Statistics (DOS) | SingStat Table Builder

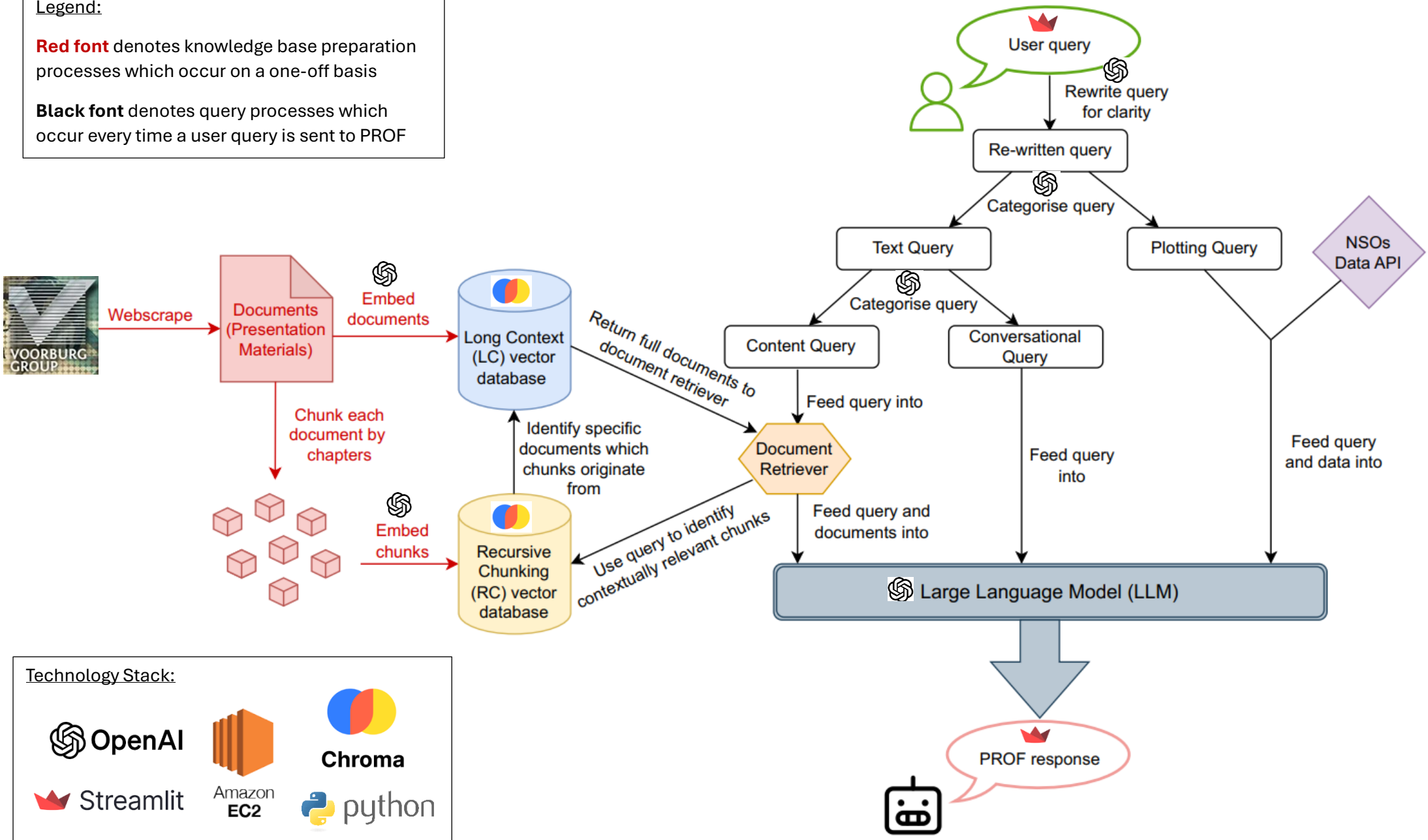
replicating the process for other specialised academic, conceptual and methodological content, e.g., consumer prices, national and international accounts, etc.

## Appendix 1

Legend:

**Red font** denotes knowledge base preparation processes which occur on a one-off basis

**Black font** denotes query processes which occur every time a user query is sent to PROF




Technology Stack:

OpenAI · Streamlit · Amazon EC2 · Chroma · python