National College of Ireland

Configuration Manual

MSc Research Project
Data Analytics

Chandan Vijay Pawar
Student ID: x22236775

School of Computing
National College of Ireland

Supervisor: Abdul Qayum

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Chandan Vijay Pawar
Student ID:	x22236775
Programme:	MSc in Data Analytics **Year:** 2024
Module:	MSc Research Project
Supervisor:	Abdul Qayum
Submission Due Date:	16/09/2024
Project Title:	InferTextIQ: Multimodal Document Analysis and Question Answering System with Model Selection
Word Count:	535 **Page Count:** 7

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Chandan Vijay Pawar

Date: 16/09/2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Chandan Vijay Pawar
x22236775

1 INTRODUCTION

InfertextIQ is a next-generation AI-based application aimed at revolutionizing the way users interact with and extract data from CSV files and PDF documents. It offers an intuitive and efficient solution for data analysis and document interrogation, using cutting-edge language models and intelligent querying systems.
This manual is designed to take you through the process of setting up, configuring, and using all areas of InfertextIQ. Whether a Data Analyst looking for an easy way to automate CSV processing workflows, a Researcher looking to extract meaningful information from hundreds of pages of PDF reports, or a Business Professional wishing to interact with documents smarter, InfertextIQ provides you with the tools that will help realize your task.

Key features:

- Intelligent CSV Querying refers to using natural language processing to raise complex queries regarding the data in a CSV file, and getting back relevant and meaningful answers.
- Advanced PDF Analysis: Extract relevant information in the context of lengthy PDFs using a Retrieval-Augmented Generation based system.
- Integrate two AI models: GPT and Gemini, to get the advantages of both. With the integration of GPT and Gemini models, you will always obtain different solutions that give you all-rounded answers to all your questions with absolute comprehensiveness.
- User-Friendly Interface: Go through a fine-tuned and Streamlit-based application for intuitive usage and perfect interaction with data.
- Flexible Handling of Data: It can upload and analyze different CSV formats and PDF documents, thus being accommodating to a huge source of varied data structures.

2 System Requirements and Installation

Before proceeding with the installation, ensure that your system meets the following requirements:

- Python 3.7 or higher
- 8GB RAM (minimum)
- 50GB free disk space
- Internet connection for API access

2.1 Installing Required Libraries

- Open a terminal or command prompt.
    - Navigate to the project root directory:
- **cd path/to/infertextiq**
    - (Optional but recommended) Create and activate a virtual environment:

- python -m venv venv
- source venv/bin/activate # On Windows, use: venv\Scripts\activate
- Install the required packages using pip:
- pip install streamlit panda's python-dotenv langchain openai google-generativeai

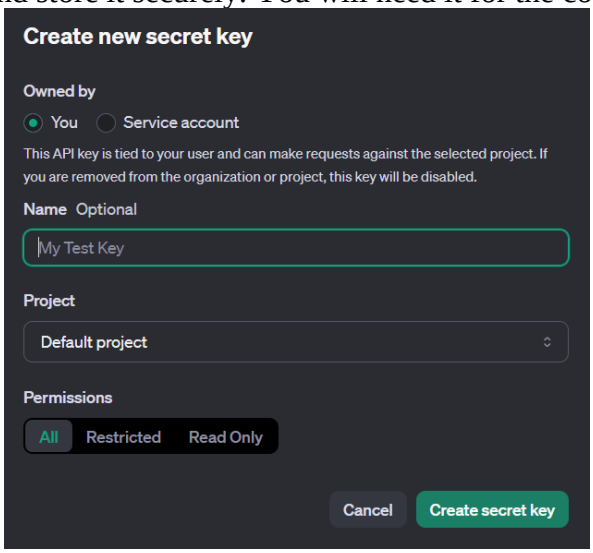
3 Fetching API Keys

The Gemini API is unavailable in Ireland. Please create a VPN to a U.S.-based server to allow the APP to work fine.

GPT 3.5 API Key

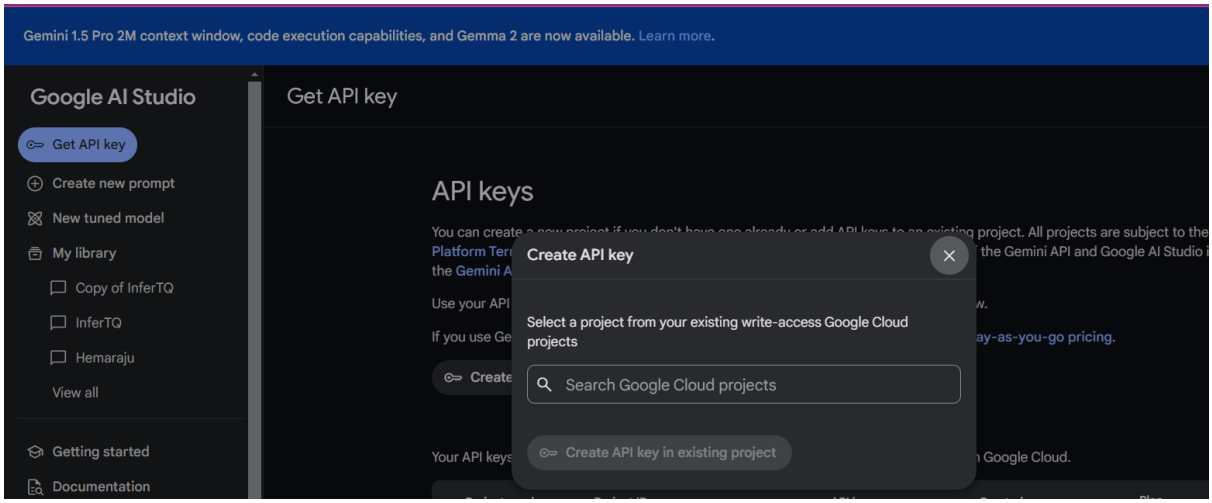
InfertextIQ requires API keys from OpenAI and Google AI to function. Follow these steps to obtain the necessary keys:

- Visit the OpenAI website (https://openai.com) and sign up for an account if you haven't already.
- Navigate to the API section of your account dashboard.
- Create a new API key. Name it "InfertextIQ" for easy identification.
- Copy the API key and store it securely. You will need it for the configuration step.



3.1 Google Gemini

1. Go to the Google Ai studio https://aistudio.google.com/app/apikey
2. Create a new project or select an existing one.
3. Enable the "Generative AI" API for your project.
4. Navigate to the "Credentials" section and create a new API key.
5. Copy the API key and store it securely.



3.2 Configuring API Keys

After obtaining both API keys, you need to add them to the project's environment:

1. In the project root directory, create a file named .env.
2. Open the .env file in a text editor and add the following lines:
    - OPENAI_API_KEY=your_openai_api_key_here
    - GOOGLE_API_KEY=your_google_api_key_here
3. Replace your_openai_api_key_here and your_google_api_key_here with your actual API keys.
4. Save and close the .env file.

4 Running the Application

Follow these steps to run InfertextIQ:.

1. Open a terminal or command prompt
2. Navigate to the project root directory:

cd path/to/infertextiq

3. Activate your virtual environment (if you created one):

streamlit run Home.py

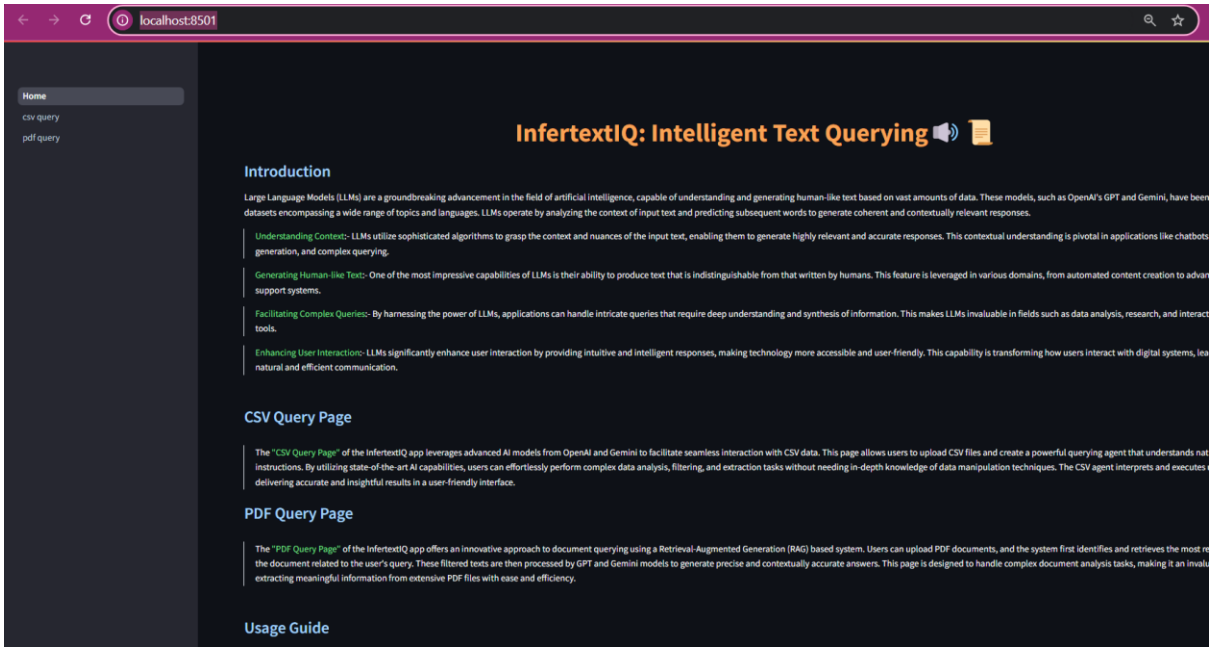
4. This should open the application in your default web browser. If not, you will see a local URL in the terminal (usually http://localhost:8501). Open this in a browser.

5 Navigating the Application

Once the application is running, you'll see the home page with an introduction to InfertextIQ. The application consists of three main pages:

1. Home Page: This summarizes all the information with respect to an application and its capabilities.
2. CSV Query Page: A feature to upload CSVs and further query them using natural language.
3. PDF Query Page: It provides a facility to upload PDF documents and retrieve information through AI-powered querying.

You can easily navigate pages using the dropdown available at the left sidebar of the application.



6 Tools and Languages:

The development of InferTextIQ utilized the following tools and programming languages:

1. Python 3.8: Primary system development language, chosen as the result of robust ecosystem in NLP and machine learning.

2. LangChain 0.1.0:Framework used for integrating various components, particularly document loading, embedding generation, and model interaction.

3. OpenAI API and Google Generative AI API: Used to access the GPT-3.5 and Gemini models, respectively.

4. PyPDF2: Employed for extracting text content from PDF files.

5. Pandas: Utilized for handling and processing CSV data.

6. Streamlit 1.18.0: Used to create the web-based user interface.

7. Chroma: Implemented as the vector database for efficient storage and retrieval of document embeddings.

8. NumPy and SciPy: Used for numerical computations, particularly in calculating similarity scores and performance metrics.

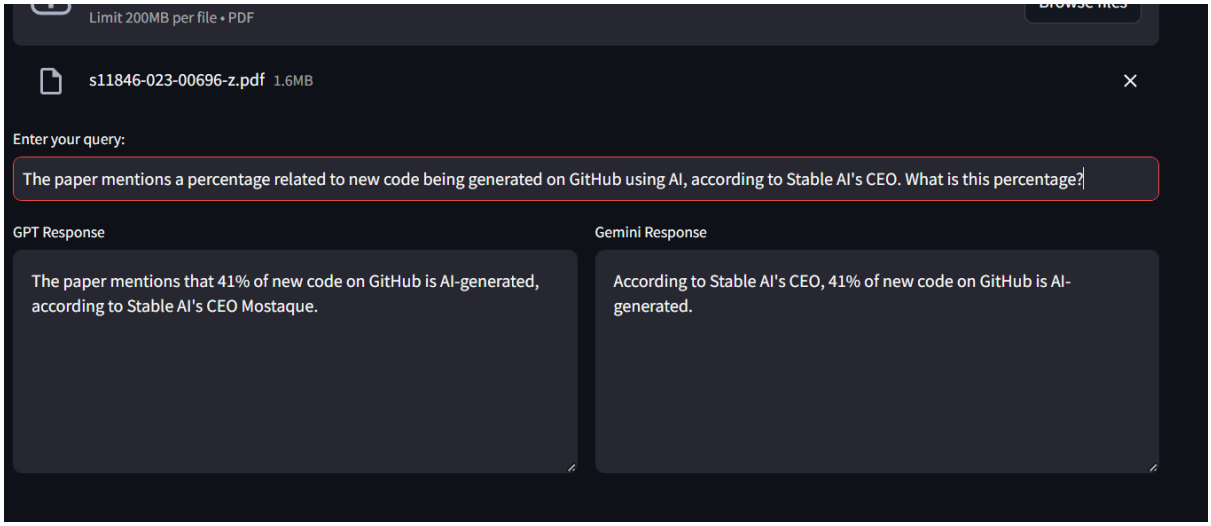
7 SAMPLE CODES and Outputs

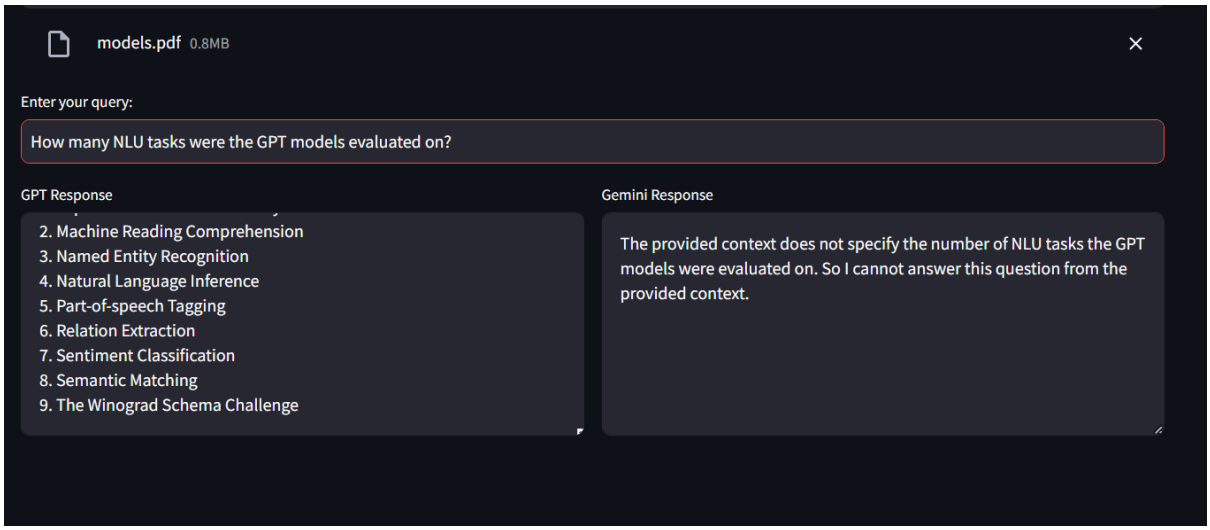
Home.py

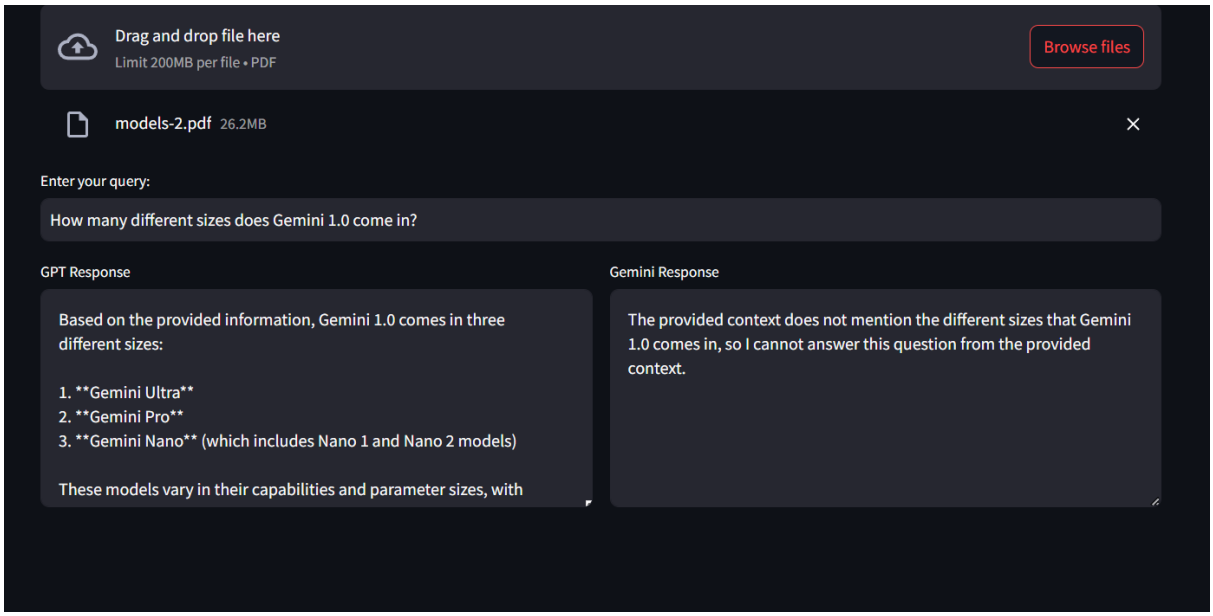
```python
title_style = """
            <style>
            .title {
                text-align: center;
                color:#faa356;
            }
            </style>
        """
st.markdown(title_style, unsafe_allow_html=True)

# Page heading
st.markdown(
    "<h1 class='title'>InfertextIQ: Intelligent Text Querying<span style='color:#b3cee5'>&#x1F50A;</span>&#x1F4DC;</h1>",
    unsafe_allow_html=True,
)

# Introduction markdown text
intro = """
        <h3 style="color:#a2d2fb">Introduction</h3>

        Large Language Models (LLMs) are a groundbreaking advancement in the field of artificial intelligence, capable of understanding and generating human-like text based

        > <span style="color:#7ce38b">Understanding Context</span>:-
        LLMs utilize sophisticated algorithms to grasp the context and nuances of the input text, enabling them to generate highly relevant and accurate responses. This cont

        > <span style="color:#7ce38b">Generating Human-like Text</span>:-
        One of the most impressive capabilities of LLMs is their ability to produce text that is indistinguishable from that written by humans. This feature is leveraged in

        > <span style="color:#7ce38b">Facilitating Complex Queries</span>:-
        By harnessing the power of LLMs, applications can handle intricate queries that require deep understanding and synthesis of information. This makes LLMs invaluable i
```

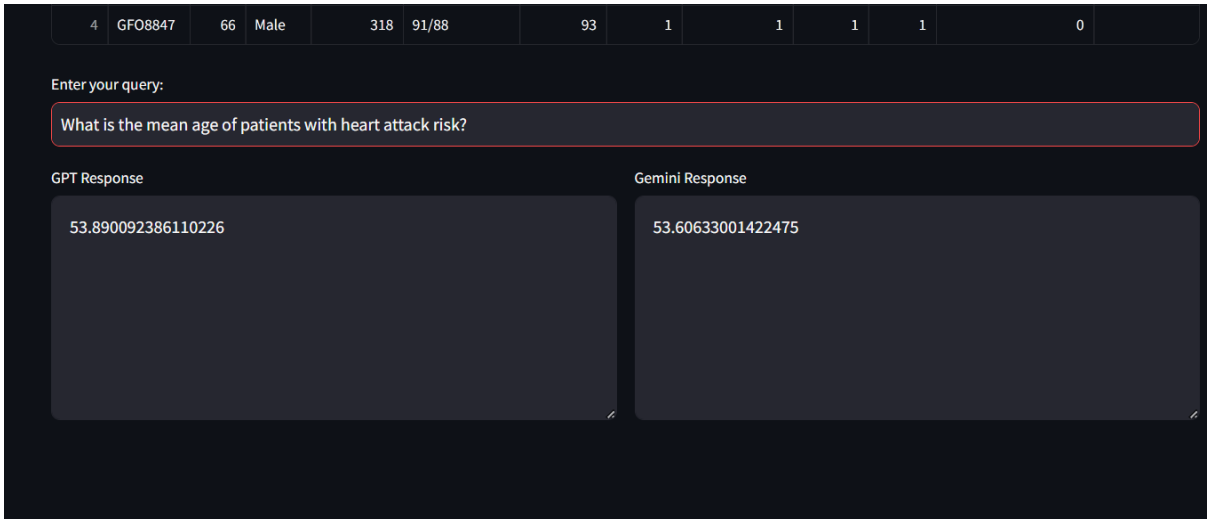
Home.py csv_query.py

```python
# Introduction markdown text
intro = """
        <h3 style="color:#a2d2fb">Introduction</h3>

        Welcome to the <span style="color:#7ce38b">"CSV Query"</span> page of InfertextIQ. Here, we leverage cuttin

        Intelligent agents are autonomous entities designed to handle complex tasks by understanding and processing

        > <span style="color:#7ce38b">LangChain Agents</span>:-
        LangChain provides a robust framework for building and deploying intelligent agents. These agents can proce

        > <span style="color:#7ce38b">GPT and Gemini Models</span>:-
        Our system leverages state-of-the-art language models such as GPT and Gemini to enhance the capabilities of

        """
st.markdown(intro, unsafe_allow_html=True)

# Usage Guide markdown text
usage_guide = """
        <h3 style="color:#a2d2fb">Usage Guide</h3>

        1. **Upload CSV File**: Begin by uploading your CSV file using the file uploader provided below. The system

        2. **Enter Query**: Once the file is uploaded, you can enter your query in natural language. For example, y

        3. **View Results**: The intelligent agent, powered by GPT and Gemini models, will process your query and d

        """
st.markdown(usage_guide, unsafe_allow_html=True)

# Adding vertical spaces
```


Limit 200MB per file • PDF
s11846-023-00696-z.pdf 1.6MB
Enter your query:
The paper mentions a percentage related to new code being generated on GitHub using AI, according to Stable AI's CEO. What is this percentage?
GPT Response
The paper mentions that 41% of new code on GitHub is AI-generated, according to Stable AI's CEO Mostaque.
Gemini Response
According to Stable AI's CEO, 41% of new code on GitHub is AI-generated.


models.pdf 0.8MB
Enter your query:
How many NLU tasks were the GPT models evaluated on?
GPT Response
2. Machine Reading Comprehension
3. Named Entity Recognition
4. Natural Language Inference
5. Part-of-speech Tagging
6. Relation Extraction
7. Sentiment Classification
8. Semantic Matching
9. The Winograd Schema Challenge
Gemini Response
The provided context does not specify the number of NLU tasks the GPT models were evaluated on. So I cannot answer this question from the provided context.


Drag and drop file here
Limit 200MB per file • PDF
Browse files
models-2.pdf 26.2MB
Enter your query:
How many different sizes does Gemini 1.0 come in?
GPT Response
Based on the provided information, Gemini 1.0 comes in three different sizes:
1. **Gemini Ultra**
2. **Gemini Pro**
3. **Gemini Nano** (which includes Nano 1 and Nano 2 models)
These models vary in their capabilities and parameter sizes, with
Gemini Response
The provided context does not mention the different sizes that Gemini 1.0 comes in, so I cannot answer this question from the provided context.



References

Get an API key. (n.d.). Google AI for Developers. https://ai.google.dev/gemini-api/docs/api-key

How to get an OpenAI API Key for ChatGPT | Maisie AI. (n.d.). https://www.maisieai.com/help/how-to-get-an-openai-api-key-for-chatgpt

Witcher, J. (2020, December 24). *Using .env Files for Environment Variables in Python Applications*. DEV Community. https://dev.to/jakewitcher/using-env-files-for-environment-variables-in-python-applications-55a1