## **1. Excel-based Q&A for Data Analytics**

### **1.1 Overview**

The Excel-based Q&A system is a robust tool designed to provide users with the ability to analyze structured data contained in Excel files or CSV files. The system supports dynamic querying, where users can pose questions in natural language, and the system responds with data analysis, visualizations, and insights. By leveraging advanced NLP and data analytics, this system makes it easier for users to interact with large datasets and extract valuable insights.

### **1.2 File Handling**

The file handling process ensures that the Excel or CSV files provided by the user are properly validated before being processed. This step is crucial to ensure the data conforms to the expected format:

* **File Types Supported:** Only CSV or Excel files are allowed, with CSV files being preferred for simplicity and consistency.
* **Structural Guidelines:** The first row in the file must contain column headers. These headers serve as the reference for querying the data and performing operations.
* **Validation:** The system performs a series of checks to ensure the file contains no corrupted data, merged cells, or complex formulas that may interfere with analysis. This ensures smooth operation without data discrepancies.
* **File Preprocessing:** The system can automatically handle missing values, outliers, and duplicate entries before further analysis. This preprocessing step can be customized based on user preferences.

### **1.3 Query Processing**

Once the data is validated, the user can input their queries in natural language. The system utilizes NLP models to interpret the user's intent and extract the appropriate response. Queries can be classified into three categories:

* **Plot-based Questions:** For questions requiring visualizations (e.g., "Show me a bar chart of sales over time").
* **Numeric Analytics Questions:** For mathematical operations on the data (e.g., "What is the average sales figure for Q1?").
* **Hybrid Questions:** A combination of both (e.g., "Plot a graph and also calculate the total sales in 2023").

The system uses a **Query Classifier** to categorize each incoming query, ensuring that the appropriate engine is used for data analysis or visualization generation.

### **1.4 Response Generation**

Once the query is processed, the system generates the response:

* **Plot-based Responses:** For questions related to visual representation, the system generates appropriate charts, graphs, or tables, such as pie charts, bar graphs, histograms, or scatter plots. These visuals are generated using libraries like Matplotlib or Plotly.
* **Numeric Responses:** For questions that require numeric answers, the system calculates the necessary statistics, such as averages, sums, medians, percentiles, or custom formulas.
* **Hybrid Responses:** The system combines both the visual representation and the textual analysis to provide a comprehensive answer, ensuring that the user receives both visual and numerical insights.

The system can also generate multiple types of visualizations, depending on the complexity of the question. For instance, if the query asks for a comparison of sales by product category over multiple years, the system will generate a stacked bar chart along with a text-based summary of the trends.

### **1.5 System Components**

The core components of the Excel-based Q&A system include:

* **File Validator:** Ensures that uploaded files meet the necessary structural guidelines. This component performs an initial check on the file’s integrity before any further processing occurs.
* **Query Classifier:** A natural language processing model that classifies queries into the appropriate categories: plot-based, numeric analytics, or hybrid.
* **Visualization Engine:** Uses advanced libraries to generate accurate charts and graphs, tailored to the question’s requirements.
* **Answer Generator:** Leverages Python-based analytics libraries to calculate numeric results and process the underlying data for textual responses.
* **Base64 Encoder:** Since the system generates graphical outputs, it converts them into base64 format to ensure seamless integration with frontend systems, allowing users to view the visualizations directly.

## **2. PowerPoint (PPT) based RAG & Vision-based Q&A**

### **2.1 Overview**

The PPT-based RAG and Vision-based Q&A system is designed to process and analyze PowerPoint presentations. It extracts both textual and visual content from each slide, applies advanced retrieval-augmented generation (RAG) techniques, and uses vision models to answer questions related to slide content. This system provides a more interactive and dynamic approach to understanding PowerPoint presentations, enabling users to ask questions based on slide content.

### **2.2 Slide Conversion Process**

The first step in processing a PowerPoint file is to convert each slide into an image. This is essential as images provide a consistent and structured format for further processing:

* **On Linux:** The system leverages **LibreOffice in headless mode** to convert slides into high-quality images. This process does not require a graphical interface and can run seamlessly in a server environment. LibreOffice's headless mode ensures that the slides are converted efficiently, even on environments with no GUI.
* **On Windows:** The system uses **Microsoft PowerPoint** through automation tools like **pywin32** to export slides as images. Since PowerPoint provides native support for exporting slides in various formats, this method ensures high fidelity in terms of visual quality.

### **2.3 Information Extraction**

Once the slides are converted into images, they are processed by a **vision-language model** (such as a combination of a CNN for image processing and an LSTM or Transformer for text processing) to extract structured data:

* **Slide Titles:** Each slide’s title is extracted for better understanding of the slide's context.
* **Text Blocks & Bullet Points:** Text within slides, including bullet points, is parsed and structured.
* **Charts, Tables, and Visual Elements:** Visual elements such as graphs, charts, and tables are detected and extracted using specialized image recognition techniques, ensuring that both textual and visual data are retrieved.

Each output is tagged with the corresponding slide number, facilitating the accurate retrieval of content when a user poses a question related to the slides.

### **2.4 Data Storage & Embedding**

The extracted text content is stored in a centralized database, where it is processed using an **embedding model** to convert the text into vector embeddings. These embeddings are essential for enabling similarity-based retrieval, allowing the system to quickly find the most relevant content when answering user queries.

Embeddings are also generated for any visual elements like charts or tables, ensuring that both text and visual content are available for search.

### **2.5 Question Answering and Slide Retrieval**

When a user submits a question:

1. The question is converted into an embedding using the same model used for slide content.
2. The system performs a **similarity search** across the stored embeddings to retrieve the most relevant slide content.
3. A **reranker** is applied to improve the relevance of the results, sorting the matching slides by their similarity to the question.
4. The top-ranked slides are selected, and the system begins the answer generation phase.

### **2.6 Vision-based Answer Generation**

During the answer generation phase, the most relevant slide images are passed to a vision-language model, which interprets both the visual and textual content of the slides. The model uses both layout analysis and text recognition to generate a response based on the user's query.

For example, if the user asks about a chart on slide 5, the system will analyze the visual components of the slide, such as axis labels and data points, along with any accompanying textual explanations. The system then provides an accurate answer and references the corresponding slide number.

### **2.7 System Components**

* **Slide Conversion Tool:** Converts PowerPoint slides into high-resolution images using platform-specific tools.
* **Vision Model:** Extracts both textual and visual content from slide images.
* **Embedding Model:** Converts extracted text into vector embeddings for efficient search.
* **Reranker:** Refines the search results by sorting the retrieved slides by relevance.
* **Answer Generator:** Combines the extracted data to generate a comprehensive answer to the user’s question.

## **3. Confluence Space-based RAG Q&A**

### **3.1 Overview**

The Confluence-based RAG Q&A system enables users to query content from Confluence spaces. Confluence, being a popular collaboration platform, contains rich documentation that can be queried for insights. This system extracts and embeds textual content from Confluence pages, providing users with a seamless way to access detailed information based on their queries.

### **3.2 User Login & Space Selection**

The system allows users to authenticate via their Confluence credentials. After successful login:

* **Space Selection:** The system lists all available Confluence spaces, and users can select one or multiple spaces for embedding. The selection is limited to a maximum of five spaces to ensure manageable embedding sizes.
* **Note on Page Trees:** The user cannot drill down into specific pages or page trees but must select full spaces for embedding. This design decision ensures consistency in querying and simplifies the process for users.

### **3.3 Content Extraction**

Once the spaces are selected, the system uses the **Confluence REST API** to extract all relevant textual content:

* **Text Extraction:** The system extracts plain text from the Confluence pages within the selected spaces.
* **Ignored Content:** Non-text elements such as images, attachments, and user mentions are ignored during extraction. This ensures that only relevant textual data is processed and embedded.
* **Table Handling:** Complex tables with merged cells are flattened into readable chunks. This is done using a **Q-model** that converts the tables into structured, readable text while preserving the integrity of the data.

### **3.4 Text Processing & Embedding**

After extraction, the text is processed into manageable chunks, ensuring that context is preserved for longer pieces of content. The text is split recursively to handle large pages effectively. Each chunk is then passed through an **embedding model** (e.g., MiniLM) to create vector embeddings. These embeddings allow for efficient similarity search during the query phase.

### **3.5 Question Answering Phase**

* **Space Selection for Q&A:** Users can select one or more previously embedded spaces for querying. If multiple spaces are selected, the system will search across all available content.
* **Question Processing:** User queries are converted into embeddings using the same embedding model as the one used during the embedding phase.
* **Answer Generation:** Based on the similarity of the query to the stored embeddings, the system retrieves the most relevant chunks and passes them to a lightweight LLM to generate an answer. The response may include content from multiple pages or chunks, depending on the complexity of the query.

### **3.6 System Components**

* **Confluence Content Extractor:** A tool for extracting textual content from Confluence spaces using the Confluence REST API.
* **Text Chunker:** Breaks down large amounts of content into smaller chunks to ensure better context retention and retrieval performance.
* **Embedding Model:** Processes the extracted text into vector embeddings that allow for efficient similarity-based search.
* **Answer Generator:** Uses the most relevant chunks of text to generate a response to the user's query, including references to the page titles and content.