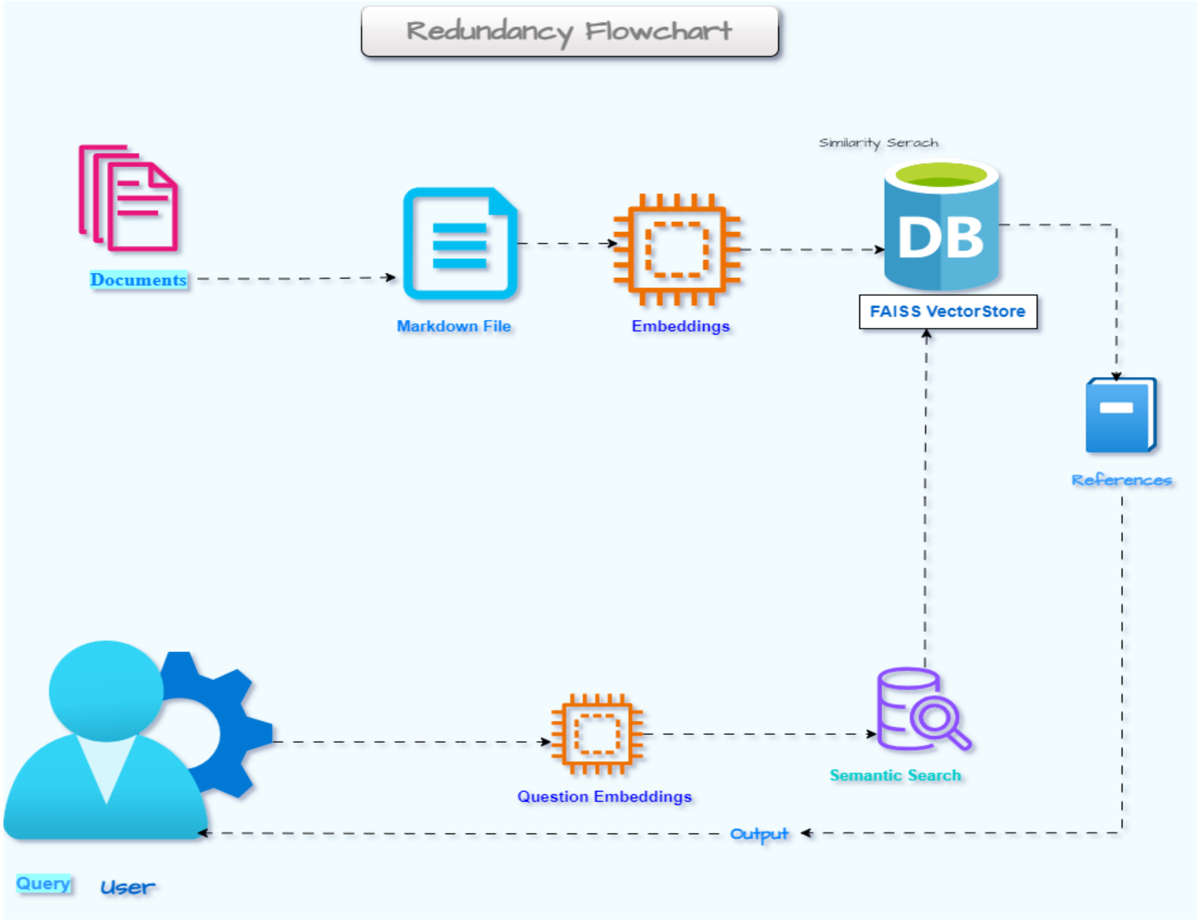
Use-Case-1-Process of Redundancy



Let's break down each section in detail using examples from NASA's MAGIC Mission (Mass Change and Geosciences International Constellation) documents.

1. Documents

Documents are collections of information on various topics. They can be in the form of papers, files, articles, reports, etc. Each document contains content that is structured or unstructured, such as text, images, tables, and other forms of data. In the context of digital information processing, documents are the raw material that we want to manage, search, and retrieve relevant information from

Example:

NASA has a collection of documents related to the MAGIC Mission, which include scientific papers, mission reports, data analysis, and research findings about Earth's mass changes and geosciences.

Markdown File :

Markdown is a lightweight markup language with plain text formatting syntax. It is designed to be easy to read and write, and it can be converted into other formats like HTML. A Markdown file typically uses simple symbols to format text, such as:

# for headings

* for bullet points

** for bold text

Markdown files are particularly useful for digital documentation because they are both human-readable and easy for computers to parse and process.

Example:

The detailed scientific papers and reports on the MAGIC Mission are converted into Markdown files

Markdown is a lightweight markup language with plain text formatting syntax. For example, a document with a section on "Sea-Level Rise Observations" could be converted into a Markdown file like this:

```markdown
# Sea-Level Rise Observations


The MAGIC Mission has provided critical data on the rising sea levels due to polar ice melt. In 2023, the observed rise was 3.2 mm/year.
```

This makes the information easier for computers to process while still being readable to humans.

Embeddings

Embeddings are a way of converting textual information into numerical representations that capture the semantic meaning of the content. In other words, embeddings translate words, phrases, and sentences into vectors (lists of numbers) that a computer can understand and manipulate. These vectors represent the 'DNA' of the text, capturing its context, meaning, and relationships to other pieces of text.

For example, the word "Space" might be converted into a vector of numbers like [0.25, -0.15, 0.10, ...], and the word "AeroSpace" would have a similar vector, showing that they are semantically related.

We have used `'BAAI/bge-large-en-v1.5'` embedding model from the HuggingFace

Example:

The text from the Markdown file about sea-level rise is converted into embeddings.

Embeddings are a way to represent text as vectors (a series of numbers). Think of it as translating words into a unique digital fingerprint. For example, the sentence "Sea-level rise observed by MAGIC Mission" might be converted into a vector like `[0.5, -0.2, 0.1, ...]`.

This conversion allows computers to understand the context and meaning of the text in a numerical format.

FAISS VectorStore:

FAISS (Facebook AI Similarity Search) is a library developed by Facebook AI that is designed for efficient similarity search and clustering of dense vectors. A VectorStore using FAISS can store embeddings (the 'DNA' of documents) and quickly find similar vectors. This makes it possible to search large collections of data very efficiently.

When embeddings of documents are stored in a FAISS VectorStore, it allows for fast retrieval of information by comparing the stored vectors with query vectors.

Example:

These embeddings are stored in a FAISS VectorStore. When the embeddings of all the MAGIC Mission documents are stored here, it allows for quick retrieval of similar information.

For example, all embeddings related to "sea-level rise" can be stored in this database, making it easy to find and compare similar topics.

Query From User:

When a user asks a question or searches for information, their query is processed similarly to the documents. The query is converted into a query embedding, a vector that represents the semantic meaning of the question. This allows the system to compare the query with the stored document embeddings in the FAISS VectorStore.

Example:

A user wants to know about the impact of ice melt on sea levels according to the MAGIC Mission.

The user's query, "What is the impact of ice melt on sea levels according to MAGIC Mission?" is converted into its own embedding (digital fingerprint) and then this embedding will compare the query embedding inside the vectorstore .

Semantic Search :

Semantic search goes beyond simple keyword matching by understanding the meaning and context of the words in the query. By using embeddings, semantic search finds documents that are not just lexically similar (i.e., containing the same words), but also semantically similar (i.e., containing similar meanings).

The FAISS VectorStore uses the query embedding to perform a similarity search among the stored document embeddings. It finds the vectors that are closest to the query vector, indicating the most relevant documents or pieces of information.

Example:

The FAISS VectorStore searches for the most similar embeddings to the user's query. The system uses the query's embedding to find the closest matching embeddings in the VectorStore. For instance, it may find the embeddings related to sections of the MAGIC Mission documents that discuss "ice melt" and "sea-level rise."

Output:

The result of the semantic search is a set of documents or pieces of information that best match the user's query. The system retrieves and presents this information, which is likely to be the most useful and relevant to the user's needs based on the semantic similarity between the query and the documents.

Example:

The system retrieves and presents the most relevant sections from the MAGIC Mission documents.

Based on the search, the system might return information like:

"The MAGIC Mission observed that ice melt in Greenland has contributed to an average sea-level rise of 0.7 mm/year over the last decade."

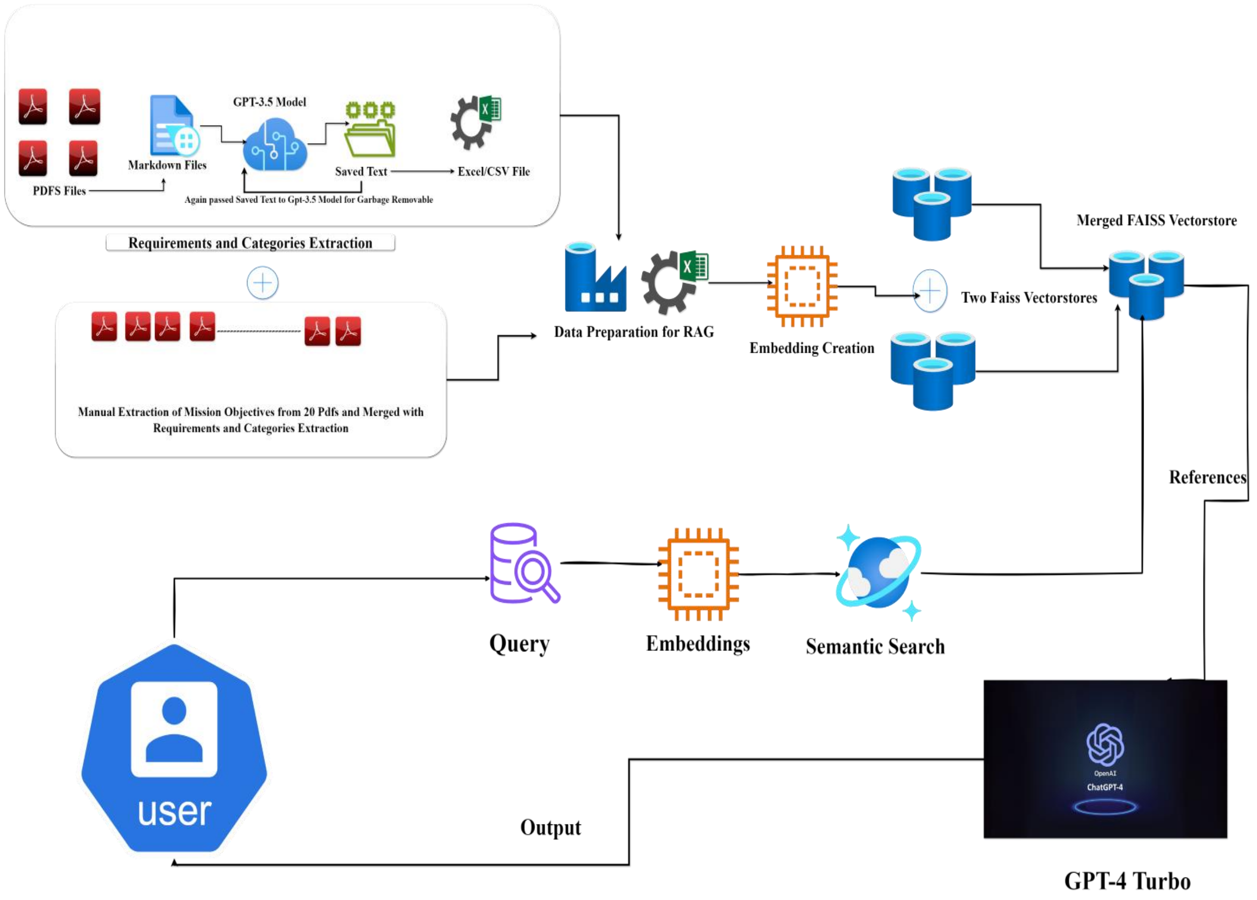
This output gives the user a precise and relevant answer to their question based on the stored documents.

Conclusion:

To summarize, the process of retrieving relevant information involves a series of steps designed to bridge the gap between human language and computer understanding. Documents are first converted into Markdown format for readability and then transformed into embeddings, essentially digital representations of their semantic content. These embeddings are stored in a specialized database, the FAISS Vector Store, optimized for quick similarity searches. When a user submits a query, it too is transformed into an embedding, allowing for semantic search across the document database. The system then returns the most pertinent information, based not just on keyword matches, but on the underlying meaning of the text. This process streamlines information retrieval, providing users with targeted and meaningful responses that address their inquiries effectively

Use-Case-2 Requirements and Categories

Process Diagram



The process diagram outlines a comprehensive and structured workflow for extracting, refining, and utilizing information from PDF documents using a combination of manual effort and advanced AI techniques. This process is exemplified using documents from the NASA MAGIC Mission. Here’s a detailed summary of the process:

PDF Conversion

The workflow begins with the conversion of PDF documents into Markdown files. This step is crucial as it transforms the data into a more accessible and manageable format, laying the groundwork for subsequent processing.

Example: Suppose we have PDF documents from the NASA MAGIC Mission containing various mission details and requirements.

Convert PDF to Markdown: Each PDF is converted into Markdown format for easier processing.

Example: A PDF titled "Magic Mission Requirlements.pdf" is converted to a Markdown file.

Initial Processing with GPT-3.5

Pass Markdown to GPT-3.5:The converted Markdown files are then processed using GPT-3.5. The Markdown content is divided into chunks, each comprising three pages with an overlapping page to ensure context continuity. These chunks are sent to GPT-3.5, which processes the text and provides preliminary insights. The responses from GPT-3.5 are saved, forming the basis for the next stage. This approach ensures that large documents are handled efficiently and context is preserved across the chunks.

Example: The Markdown file from "Magic Mission Requirements.pdf" is divided into chunks, and these chunks are processed by GPT-3.5 to extract text and preliminary insights.

Save GPT-3.5 Responses:

The responses from GPT-3.5 for each chunk are saved.

Example: The output for each chunk of "Magic Mission Requirements.pdf" is saved as separate text files.

Garbage Removal:

Combine Responses: The saved responses are combined into a single text file, which is then passed back to GPT-3.5 for cleaning. This step involves removing any irrelevant or redundant information, resulting in a refined text file containing only the essential requirements and categories. This cleaned text is saved for further use.

Example: Text files from all chunks of "Magic Mission Requirements.pdf" are merged into a single file.

Pass Combined Text to GPT-3.5: This combined text file is passed to GPT-3.5 for cleaning, removing any irrelevant or redundant information.

Example: The merged text from "Magic Mission Requirements.pdf" is refined by GPT-3.5 to eliminate any unnecessary content.

Save Cleaned Text:

The cleaned text, which now contains extracted requirements and categories, is saved.

Example: The cleaned text file containing structured requirements and categories from "Magic Mission Requirements.pdf" is stored.

Manual Extraction:

Manual Extraction of Mission Objectives: While AI provides significant automation, certain aspects still require human intervention for accuracy. Mission objectives, which are often nuanced and context-specific, are manually extracted from the PDFs. This manual effort ensures that the critical mission objectives are accurately captured.

Example: From "Magic Mission Requirements.pdf", mission objectives like "Achieve precise orbit insertion" and "Conduct comprehensive atmospheric studies" are manually noted.

Merge Extracted Data:

Merge Manual and Automated Data: The manually extracted mission objectives are then merged with the requirements and categories derived from the automated process. This integration ensures a comprehensive dataset that combines the precision of human judgment with the efficiency of AI-driven extraction.

Example: The manually extracted objectives are merged with the cleaned requirements from "Magic Mission Requirements.pdf".

Data Preparation for RAG (Retrieval-Augmented Generation):

Prepare Data for Multiple PDFs: Prepare CSV files in the specified format (pdf_name, mission objectives, requirements) for a set of PDFs.

Example: Create a CSV file for each PDF with columns for the PDF name, mission objectives, and requirements.

Prepare Data for Additional PDFs: Similarly, prepare CSV files for another set of PDFs.

Example: Create CSV files for these additional PDFs in the same format.

Embedding Creation:

Create Embeddings: OpenAI embeddings are created for both datasets. These embeddings transform the textual data into a numerical format that captures the semantic meaning of the text, enabling advanced search and retrieval capabilities.

Example: Generate embeddings for the text data in the CSV files of "Magic Mission Requirements.pdf" and other PDFs.

FAISS Vectorstore Creation:

Create Separate FAISS Vectorstores: Create FAISS Vectorstores for both sets of embeddings separately. Separate FAISS Vectorstores are created for each set of embeddings. FAISS (Facebook AI Similarity Search) is a library that efficiently stores and searches through vector representations of data. These Vectorstores are then merged into a single comprehensive Vectorstore, creating a unified system for efficient information retrieval.

Example: Build separate FAISS Vectorstores for the embeddings from the two sets of PDFs.

Merge Vectorstores:

Merge the two separate FAISS Vectorstores into one comprehensive Vectorstore.

Example: Combine the FAISS Vectorstores to have a unified search and retrieval system.

Retrieval and Inference:

Use GPT-4 Turbo:

The final step involves utilizing OpenAI GPT-4 Turbo for retrieval and inference using the merged FAISS Vectorstore. This setup allows for sophisticated query handling, enabling users to retrieve relevant information and gain insights from the processed NASA MAGIC Mission documents. For instance, when queried about specific mission objectives or requirements, GPT-4 Turbo can quickly and accurately provide the necessary information.

Example: Using the combined FAISS Vectorstore, GPT-4 Turbo can now retrieve relevant information and provide inferences when queried about the NASA MAGIC Mission documents.

Scenario:-

Detailed Workflow:

1. User Query Interpretation:

The user's query, such as "What are the mission objectives?" is processed by GPT-4 Turbo. GPT-4 Turbo uses its language understanding capabilities to interpret the intent behind the query.

2. Retrieving Mission Objectives Section:

The system accesses the pre-organized datasets where the mission objectives section is indexed. Using this index, it retrieves all paragraphs under this section.

Example Extraction:

Suppose the mission objectives section contains three paragraphs. The system extracts each paragraph and prepares them for the user.

3. Searching Through Requirements:

The system searches for the phrase "shall be" within the requirements section.

For each occurrence of "shall be," the system extracts the entire sentence or requirement that contains this phrase.

Example Extraction:

If the document states, "The system shall be capable of operating under extreme conditions," this entire sentence is extracted.

4. Compilation of Results:

The mission objectives paragraphs and the "shall be" requirements are aggregated.

Formatted Output:

The system formats the results in a user-friendly manner, ensuring clarity and comprehensiveness.

The user receives a well-organized response that includes:

All paragraphs detailing the mission objectives.

All requirements containing the phrase "shall be."

Conclusion:

The outlined process exemplifies the integration of manual extraction and advanced AI-driven techniques to handle large volumes of complex documents. By converting PDFs to Markdown, leveraging GPT-3.5 for initial processing and cleaning, manually extracting mission-critical information, and creating structured datasets, the workflow ensures accuracy and efficiency. The creation of embeddings and FAISS Vectorstores further enhances the retrieval capabilities, culminating in the use of GPT-4 Turbo for sophisticated inference and query resolution. This systematic approach ensures that essential data is meticulously extracted, cleaned, organized, and made readily accessible, significantly enhancing the capability to manage and utilize large datasets effectively.