A logo of a space agency

Description automatically generated

*Internship Report Entitled*

**WELD DEFECT DETECTION AND SEGEMENTATION IN RADIOGRAPHIC IMAGES USING DEEP LEARNING TECHNIQUES**

*Submitted by*

**ARAVIND VIJAYAN**

*Under the guidance of*

**Smt. SMITHA K.K**

TECH.OFR-C, CND/CISDG

**June, 2025**

A logo of a space agency

Description automatically generated

***Certificate Of Completion***

This is to certify that **Aravind Vijayan,** Bachelor of Technology in Computer Science and Engineering of Cochin University of Science and Technology has completed the work presented in the report titled "***Weld Defect Detection and Segmentation in Radiographic images using Deep Learning Techniques*** " at **Liquid Propulsion Systems Centre**, Valiamala, Trivandrum during the period from October 2024 – November 2024. This is a bonafide record of his original work carried out under my supervision and guidance.

**Smt. Smitha K.K**  
Tech.Ofr-C, CND/CISDG

LPSC, Thiruvananthapuram

**Acknowledgement**

With a deep sense of gratitude, I wish to express my sincere thanks to my guide **Smt. Smitha K.K** (Tech.Ofr-C, CND/CISDG), for her unwavering support, scholarly guidance, and relentless cooperation throughout this project. His mentorship has been invaluable in helping me to complete the internship smoothly.

I am also deeply grateful to **Mr. Dushyant M.P** (GD, CISDG) and **Mr. Ajith S** (Head, PPEG) for allowing me to carry out my project in their esteemed department at the Liquid Propulsion Systems Centre, Valiamala.

Additionally, I would like to thank all the Sci/Engineers of Computer Infrastructure and Software Development Group (CISDG) who assisted me during my internship. It is with immense pleasure and gratitude that I acknowledge their support.

I extend my heartfelt thanks to the department staff members and friends who contributed to the successful completion of this internship.

I perceive this opportunity as a significant milestone in my professional growth. I will strive to use the skills and knowledge gained in the best possible way and will continue to work on their improvement to attain the desired level of expertise.

Sincerely,

Aravind Vijayan

**ABSTRACT**

Weld quality inspection is a critical aspect of manufacturing across various industries, including aerospace, where structural integrity directly impacts safety and performance. Radiographic testing (RT) using X-ray imaging is a widely employed Non-Destructive Testing (NDT) method to detect internal defects in welded joints without damaging the components. At the Liquid Propulsion Systems Centre (LPSC), ISRO, precise inspection of welded aluminium tanks used in the PSLV rocket is essential to ensure mission success.

Traditionally, the analysis of these radiographic X-ray images has been performed manually by skilled inspectors, a process that is time-consuming and prone to subjectivity. To address these challenges, this project focuses on developing an automated deep learning-based system for detecting and segmenting weld defects from X-ray images. The process involves validating the quality of input images, isolating the weld area, and extracting embedded textual annotations for traceability. The core of the project lies in employing the YOLOv8 segmentation model, which enables accurate pixel-level identification of weld defects such as isolated, aligned, and clustered pores.

This deep learning-driven approach not only improves detection accuracy but also significantly reduces inspection time and human error. The solution is designed to be scalable and adaptable, with potential applications extending beyond aerospace to various industrial sectors requiring rigorous quality assurance.

By automating the segmentation of weld defects, this project contributes to modernizing traditional NDT workflows, enhancing efficiency, repeatability, and reliability in weld quality inspection processes.

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| **ABBREVIATIONS** | **FULL FORMS** |
| ISRO | Indian Space Research Organisation |
| LPSC | Liquid Propulsion Systems Centre |
| NDT | Non-Destructive Testing |
| DL | Deep Learning |
| YOLO | You Only Look Once |
| PDF | Portable Document Format |
| FAISS | Facebook AI Similarity Search |
| GPU | Graphics Processing Unit |
| CPU | Central Processing Unit |
| EMB | Embedding |
| QA | Question Answering |
| DB | Database |
| UI | User Interface |
| UX | User Experience |

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TITLE** | **PAGE NO.** |
| **1** | **ABSTRACT** | **3** |
|  | **LIST OF ABBREVIATIONS** |  |
| **2** | **INTRODUCTION** | **6** |
| **3** | **ABOUT** | **7** |
| **4** | **LITERATURE SURVEY** | **7** |
| **5** | **SYSTEM ARCHITECTURE** | **11** |
| **6** | **MODEL SELECTION AND CONFIGURATION** | **14** |
| **7** | **SYSTEM DESIGN** | **15** |
| **8** | **IMPLEMENTATION** | **18** |
| **9** | **EMBEDDING AND INDEXING** | **23** |
| **10** | **TESTING** | **24** |
|  | **RESULTS** |  |
|  | **CONCLUSION** |  |
|  | **FUTURE SCOPE** |  |
| **11** | **REFERENCES** | **31** |

**1. INTRODUCTION**

Weld quality assurance is a fundamental requirement across various industries, especially in aerospace, where structural integrity is critical for mission success and safety. The inspection of welds using radiographic X-ray images is a widely adopted Non-Destructive Testing (NDT) technique that enables the detection of internal flaws without damaging the component. However, the traditional manual analysis of these images by trained inspectors is time-consuming, subjective, and often inconsistent, which can lead to missed defects or inaccurate evaluations.

This project aims to develop an automated deep learning-based system for the detection and segmentation of weld defects in radiographic images, with a particular focus on welded aluminium tanks used in the PSLV rocket at the Liquid Propulsion Systems Centre (LPSC), ISRO. The system improves upon conventional methods by incorporating advanced image processing techniques to validate X-ray quality, isolate weld regions, and extract embedded textual annotations. Central to this approach is the use of a state-of-the-art segmentation model that precisely identifies various types of weld defects, including isolated, aligned, and clustered pores.

By integrating deep learning into the weld inspection workflow, this project seeks to enhance the accuracy, speed, and reliability of defect detection. The automation not only reduces human effort and error but also offers scalability, making the solution applicable to a wide range of industrial quality assurance processes beyond aerospace. Through this work, the project contributes to modernizing traditional NDT practices, enabling more efficient and objective weld quality evaluations.

**2. ABOUT**

**2.1 Indian Space Research Organisation (ISRO)**

India’s journey into space began with the establishment of the Indian National Committee for Space Research (INCOSPAR) by the Government of India in 1962. The visionary Dr. Vikram Sarabhai, regarded as the father of the Indian space program, played a pivotal role in this initiative by founding the Thumba Equatorial Rocket Launching Station (TERLS) in Thiruvananthapuram, dedicated to upper atmospheric and microgravity research.

In 1969, INCOSPAR was superseded by the Indian Space Research Organisation (ISRO), which has since remained committed to its mission of harnessing space technology for the benefit of the nation and its people. Over the years, ISRO has evolved into one of the world’s leading space agencies.

ISRO operates one of the largest fleets of communication satellites under the INSAT series and remote sensing satellites under the IRS series, addressing the growing demands for efficient communication and earth observation. It develops and delivers a wide range of application-specific satellite products and tools supporting national broadcasting, telecommunications, weather forecasting, disaster management, geographic information systems (GIS), cartography, navigation, telemedicine, and distance education.

In terms of launch vehicle development, ISRO has successfully designed the Polar Satellite Launch Vehicle (PSLV) for deploying IRS satellites and the Geosynchronous Satellite Launch Vehicle (GSLV) for GSAT class satellites. The organisation also actively pursues ambitious space science missions such as Chandrayaan-2 and the Mars Orbiter Mission (MOM).

ISRO supports academic and research institutions across India by fostering projects aligned with the national space program. To commercialize its space products and services, the government-owned Antrix Corporation was established in 1992.

Looking ahead, ISRO is focused on future technologies including the development of heavy lift launch vehicles, human spaceflight programs, reusable launch vehicles, semi-cryogenic engines, and single and two-stage-to-orbit (SSTO and TSTO) vehicles. Additionally, it is advancing the use of composite materials and other innovative technologies to meet the evolving needs and ambitions of the country’s space endeavors.

**2.2 Liquid Propulsion Systems Centre (LPSC)**

The Liquid Propulsion Systems Centre (LPSC) is the premier centre responsible for the development and realization of advanced earth-to-orbit propulsion stages for launch vehicles, as well as in-space propulsion systems for spacecraft. LPSC’s activities and facilities are distributed across two campuses: the headquarters and design offices located at Valiamala, Thiruvananthapuram, and the Spacecraft Propulsion Systems division situated in Bengaluru, Karnataka.

LPSC is entrusted with the design, development, and system engineering of high-performance space propulsion systems utilizing both earth-storable and cryogenic propellants for ISRO’s launch vehicles and satellites. The centre also undertakes the development of critical components such as fluid control valves, transducers, propellant management devices, and other essential elements of liquid propulsion systems.

The Valiamala campus serves as the Centre Headquarters and is responsible for research and development, system design and engineering, and project management. It houses core entities including the Fluid Control Components Entity, the Materials & Mechanical Engineering Entity, and the Earth Storable & Cryogenic Propulsion Entities, all of which play vital roles in the centre’s mission.

LPSC has made significant contributions to the development of liquid propulsion stages for ISRO’s Polar Satellite Launch Vehicle (PSLV) and Geosynchronous Satellite Launch Vehicle (GSLV). Notably, the centre developed the cryogenic upper stage for the GSLV MkIII, greatly enhancing ISRO’s payload capacity to geostationary orbits. This breakthrough cryogenic technology is critical for launching heavier payloads and has positioned India competitively in the global satellite launch market. Additionally, LPSC is responsible for the propulsion systems of interplanetary missions, including the Mars Orbiter Mission and the forthcoming Chandrayaan-3.

Beyond launch vehicle propulsion, LPSC designs and develops propulsion systems for spacecraft, encompassing attitude control, orbital correction, and primary propulsion. The centre has engineered propulsion solutions for communication satellites, remote sensing satellites, and scientific missions. A notable achievement includes the liquid apogee motor, which facilitates orbit-raising maneuvers for geostationary satellites—an essential capability for precise satellite positioning and operation in space.

2.3 Deep Learning Image-Based Models and Applications

Deep learning has emerged as a transformative approach in the field of image analysis, enabling automated and highly accurate interpretation of complex visual data. In the context of non-destructive testing (NDT), deep learning image-based models combine the power of advanced neural network architectures with large datasets to detect, classify, and segment defects in radiographic images with unprecedented precision.

**Key Components of Deep Learning Image-Based Models:**

1. **Feature Extraction:** These models automatically learn and extract hierarchical features from raw image data, capturing spatial patterns, textures, and subtle variations that are critical for distinguishing weld defects.
2. **Defect Detection:** By leveraging convolutional neural networks (CNNs) and region-based algorithms, deep learning models localize defects within radiographic images, enabling rapid and reliable identification of anomalies such as isolated pores, aligned pores, and clustered pores.
3. **Segmentation:** Advanced segmentation models provide pixel-level delineation of defect boundaries, offering detailed insights into defect morphology, size, and distribution. This fine-grained analysis is essential for accurate assessment and quality control.
4. **Integration with Traditional Techniques:** When combined with classical image processing and optical character recognition (OCR), these models create comprehensive inspection workflows that include image validation, weld localization, defect segmentation, and annotation extraction.

**Applications in Industrial Quality Assurance:**

* **Automated Weld Inspection:** Deep learning image-based models streamline the evaluation of welded structures by providing consistent, objective, and rapid defect detection and segmentation, reducing reliance on manual inspections.
* **Scalability Across Sectors**: Although initially developed for aerospace applications such as inspection of aluminium tanks in launch vehicles, these models are broadly applicable across industries requiring stringent weld quality assurance, including automotive, construction, and energy sectors.
* **Enhanced Decision Support:** By providing precise spatial and morphological defect information, these models enable engineers and quality control personnel to make better-informed decisions regarding material integrity and safety.

2.4 Project Overview and Objectives

This project focuses on developing a deep learning-based system for the automatic detection and segmentation of weld defects in radiographic images of welded aluminium tanks used in launch vehicles. The primary goal is to create an accurate and efficient solution that supports quality assurance by identifying critical weld defects such as isolated, aligned, and clustered pores, enhancing the structural integrity and safety of these tanks.

During the project, multiple models and techniques were explored and evaluated to address different sub-tasks in the inspection pipeline. Traditional Convolutional Neural Networks (CNN) and region-based architectures like Faster R-CNN were tested for validating the authenticity of X-ray images. Image processing techniques using OpenCV were employed to accurately detect and isolate the weld area within the radiographs. For weld defect segmentation, advanced models like YOLOv8 were utilized to achieve precise pixel-level defect delineation.

**Key Objectives of the Project:**

1. **X-ray Image Validation:**  
   Evaluate and compare multiple deep learning models, including CNN and Faster R-CNN, to develop a reliable method for confirming the validity of radiographic images before further analysis.
2. **Weld Area Detection:**  
   Apply OpenCV-based image processing techniques to localize the weld region in the X-ray images, enabling focused and accurate defect detection.
3. **Annotation Extraction:**  
   Use Optical Character Recognition (OCR) tools to extract relevant textual information such as weld IDs and inspection notes from the images, facilitating detailed documentation.
4. **Weld Defect Segmentation:**  
   Implement state-of-the-art segmentation models, primarily YOLOv8, to detect and precisely segment weld defects including isolated, aligned, and clustered pores, providing detailed defect mapping.
5. **Performance Evaluation and Optimization:**  
   Continuously assess the performance of all models and optimize the integrated system to ensure accuracy, speed, and robustness suitable for industrial deployment.
6. **Scalability and Generalization:**  
   Develop a scalable system adaptable to different welded aluminium tanks and potentially other welded components across industries, ensuring broad applicability.

3. LITERATURE SURVEY

**3.1 Deep Learning for Weld Defect Detection in Radiographic Images**

**3.1.1 Overview of Weld Defect Detection**  
Weld inspection is critical for ensuring structural integrity in industries such as aerospace, shipbuilding, and pressure vessel manufacturing. Traditional methods for inspecting welds, such as manual radiographic interpretation, are time-consuming and subject to human error. In recent years, deep learning (DL) techniques have gained popularity due to their high accuracy and automation potential in detecting weld defects from radiographic images.

**3.1.2 Challenges in Radiographic Weld Defect Detection**  
The detection of weld defects through X-ray images presents several challenges:

* **Low Contrast and Noise**: Weld defects often appear faint and can be obscured by image noise, making them difficult to detect with traditional algorithms.
* **Class Imbalance**: Certain defects, like cracks or inclusions, are rarer than others, complicating the training of robust classification models.
* **Tiny Defect Sizes**: Small-scale defects such as clustered pores may be lost in down sampled feature maps used by some detection models.
* **Data Scarcity**: Annotated weld radiograph datasets are limited due to proprietary concerns and annotation complexity.

**3.1.3 Technological Advancements**  
Deep learning methods have significantly advanced the capabilities of weld defect detection:

* **Convolutional Neural Networks (CNNs)**: CNN-based architectures like ResNet and VGG have been used for classifying weld defects such as porosity, cracks, and lack of fusion.
* **Object Detection Models**: Models such as Faster R-CNN, YOLOv5, and YOLOv8 have been applied to detect and localize weld defects in real-time with high precision.
* **Segmentation Models**: YOLOv8 and Mask R-CNN architectures provide pixel-level segmentation of defects, aiding in precise localization and measurement.
* **Preprocessing Techniques**: Methods such as Adaptive Thresholding and curve transforms have been employed to enhance defect visibility prior to model input.

**3.1.4 YOLO-Based Detection Approaches**  
YOLO (You Only Look Once) has emerged as one of the most popular object detection architectures due to its speed and accuracy, especially in real-time applications. Various modified versions of YOLO have been tailored for weld defect detection in X-ray images, addressing challenges such as low contrast, small object size, and deployment on resource-constrained devices:

* **S-YOLO (Slim YOLO using YOLOv8-Nano)**  
  This approach leverages the lightweight YOLOv8-Nano model, optimized for deployment on edge devices with limited computational power. It reduces model complexity while maintaining adequate detection performance, making it suitable for small-scale weld defect detection in real-time inspection systems.
* **WT-YOLO (Wavelet-Transformed YOLO using YOLOv5)**  
  Built on the YOLOv5 architecture, WT-YOLO introduces Discrete Wavelet Transform (DWT) layers prior to the backbone to enhance multi-resolution analysis. This improves the model’s sensitivity to low-contrast or blurred features typical in industrial X-ray images, leading to significant improvements in multi-class defect classification accuracy.
* **WD-YOLO (Weighted Dual-Attention YOLO using YOLOv7)**  
  WD-YOLO extends the YOLOv7 framework by integrating dual attention modules—spatial and channel-wise attention—as well as image enhancement preprocessing. This model is especially effective in scenarios with complex backgrounds or uneven illumination, improving both precision and recall for subtle defect types.

**3.1.6 Public Datasets and Benchmarks**

* **GDXray**: One of the most widely used public datasets for weld and casting defect detection.
* **Industrial Datasets**: Proprietary datasets from research institutions and companies provide higher variability in defect types and imaging conditions, though often not publicly available.

**3.1.7 Case Studies in Industrial Applications**

* **Aerospace Industry**: CNN-based systems integrated into automated inspection pipelines to identify lack of penetration and porosity in aluminum fuel tanks.
* **Manufacturing Quality Control**: YOLO and UNet models used for in-line detection of cracks in high-speed production environments, reducing inspection time and human workload.

**3.1.8 Future Directions**  
Advancements in weld defect detection are expected to focus on:

* **Multimodal Inspection**: Fusion of X-ray, ultrasonic, and thermographic data to enhance defect detection reliability.
* **Domain Adaptation**: Techniques to transfer models trained on one domain (e.g., steel welds) to another (e.g., aluminum welds) without retraining from scratch.
* **Real-Time Edge Deployment**: Developing lightweight models suitable for deployment on embedded systems at manufacturing sites.
* **Explainability and Trust**: Integrating visualization tools to explain model decisions and foster trust among inspectors and engineers.

### 4. SYSTEM ARCHITECTURE

The architecture of the proposed Retrieval-Augmented Generation (RAG) system for ISRO is designed to efficiently load documents, generate embeddings, and retrieve relevant context for answering user queries. Below, we detail the existing solutions for knowledge retrieval, followed by the design of the proposed RAG system and its components.

#### 4.1 Existing Solutions for Knowledge Retrieval

Existing solutions for knowledge retrieval rely heavily on traditional search engines and static information retrieval models. These methods are effective for structured queries but struggle to handle unstructured, domain-specific knowledge, such as the technical documentation and scientific papers used in ISRO. Key issues include:

* Keyword-based Retrieval: Most existing systems use keyword matching or simple query-to-document retrieval methods, which lack semantic understanding and often retrieve irrelevant content.
* Limited Context Understanding: Traditional models do not capture the broader context of scientific documents and mathematical expressions, leading to incomplete or inaccurate responses.
* Manual Document Parsing: In current systems, human experts often manually extract equations and other critical technical data from documents, which is time-consuming and error-prone.

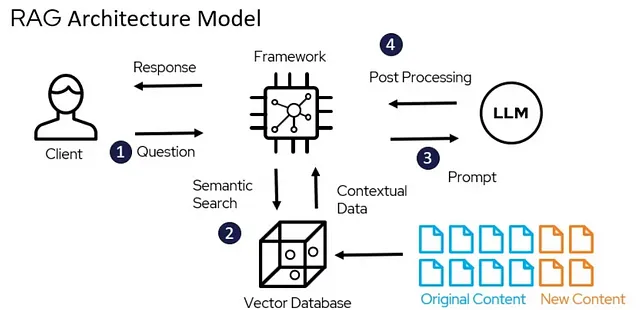
The shortcomings of these systems necessitate the development of a more sophisticated approach that can understand and process the complex technical content, particularly equations and mathematical relationships, present in ISRO’s scientific documentation.

#### 4.2 Proposed RAG LLM System

The proposed RAG system integrates state-of-the-art models for embedding generation, document retrieval, and language model-based response generation. The architecture is designed to process both textual content and mathematical expressions accurately, providing detailed explanations and contextually relevant answers.

##### **System Workflow**

1. Document Loading: The system supports loading documents in PDF and TXT formats. It uses the langchain library to load and split large documents into smaller chunks for easier processing.
2. Embedding Generation: Sentence embedding models, such as sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2, are used to generate vector embeddings for each chunk of text. These embeddings capture the semantic content of the text and are used to match relevant chunks with user queries.
3. FAISS Indexing: Once the embeddings are generated, they are indexed using FAISS (Facebook AI Similarity Search). FAISS allows for efficient nearest-neighbour search in high-dimensional spaces, enabling the system to quickly retrieve relevant document chunks based on the similarity between the query and the document embeddings.
4. Query Processing: The user submits a query, and the system generates an embedding for the query. The FAISS index is then queried to find the most relevant document chunks, which are returned based on similarity scores.
5. Contextual Response Generation: The system uses an LLM (meta-llama/Llama-2-7b-hf) to generate a detailed response based on the retrieved context. The response generation is customized to focus on extracting and explaining mathematical content, providing definitions for variables, and describing the significance of equations.
6. Interactive Interface: The user interacts with the system through a simple command-line interface, where they can load documents, ask questions, and retrieve answers based on the loaded content.



Key Features of the Proposed System

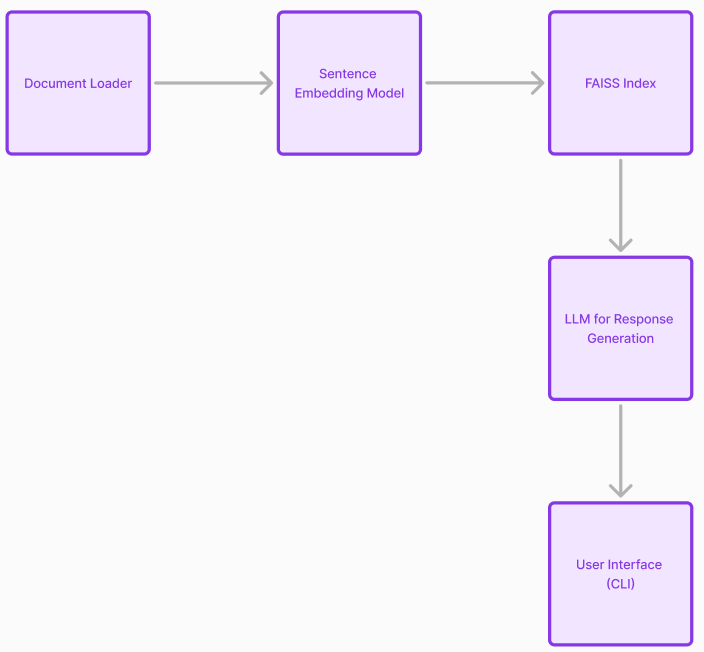
* Domain-Specific Optimization: The system is designed specifically for ISRO’s technical documents, enabling it to handle equations, scientific terms, and other domain-specific content efficiently.
* Enhanced Accuracy: By using semantic embeddings and FAISS indexing, the system improves the relevance and accuracy of retrieved documents compared to traditional keyword-based search methods.
* Mathematical Analysis: The system is capable of extracting, analyzing, and explaining complex mathematical expressions from documents, making it highly suitable for technical fields like rocket propulsion and related areas.

#### 4.3 Component Breakdown

The system consists of several components, each responsible for a specific part of the process:

1. Document Loader:
   * PyPDFLoader: Handles the loading of PDF documents, extracting text while maintaining structure.
   * TextLoader: Loads plain text documents and prepares them for further processing.
2. Text Splitter:
   * RecursiveCharacterTextSplitter: This component splits large documents into smaller, manageable chunks, ensuring each chunk is under a specified size (e.g., 1000 characters) to avoid input length limitations in the models. It also allows for some overlap between chunks to preserve context.
3. Embedding Model:
   * SentenceTransformer: The embedding model (sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2) is used to generate high-dimensional embeddings of document chunks. These embeddings are later used for semantic search and context retrieval.
4. FAISS Index:
   * FAISS: A library for efficient similarity search and clustering of dense vectors. It is used here to index the document embeddings and enable fast retrieval of relevant chunks based on a user’s query.
5. LLM for Response Generation:
   * Llama-2-7b-hf: This is the large language model used to generate detailed responses to the user’s query. It takes the retrieved context and produces a structured answer that includes mathematical analysis, equation extraction, and technical explanations.
6. User Interface:
   * The system provides an interactive interface where users can load documents, submit queries, and view detailed answers. The interface is designed for ease of use and efficiency, ensuring smooth interaction with the backend components.
7. Error Handling and Feedback:
   * The system includes error handling mechanisms to ensure robustness. If any issues arise during document loading, indexing, or query processing, appropriate feedback is provided to the user.

##### **Component Interaction Diagram**



1. Documents are loaded and split into chunks.
2. Embeddings are generated and indexed using FAISS.
3. The user asks a question, triggering the search and response generation process.
4. The LLM generates a detailed, contextually relevant answer, which is presented to the user via the interface.

### 5. MODEL SELECTION AND CONFIGURATION

### **5.1 Comparison of Embedding Models**

In the proposed Retrieval-Augmented Generation (RAG) system, embedding models play a crucial role in transforming textual data into vector representations. These models help calculate similarities between the user query and documents, ensuring accurate retrieval. Here’s a comparison of some popular embedding models based on their performance, size, and relevance to the space research domain:

* paraphrase-multilingual-MiniLM-L12-v2:
  + Strengths: A multilingual model optimized for semantic similarity tasks across different languages. It is lightweight, efficient, and performs well in multilingual contexts.
  + Performance: Offers strong performance for sentence-level embeddings while ensuring cross-lingual compatibility. Suitable for multilingual space research teams or datasets.
  + Use Case: Ideal for scenarios where multilingual queries and documents need to be processed efficiently.
  + Limitations: May not provide the same level of granularity for highly specialized or technical content in a single language compared to domain-specific models.
* all-MiniLM-L6-v2:
  + Strengths: A lightweight and efficient model that balances speed and performance, making it ideal for real-time applications with limited computational resources.
  + Performance: Solid performance on sentence similarity tasks, though it may not capture nuanced relationships in specialized technical content.
  + Use Case: Best suited for scenarios where computational efficiency is a priority and domain-specific fine-tuning is not required.
  + Limitations: Lacks the ability to capture fine-grained semantic details, making it less ideal for highly specialized space research queries.
* all-mpnet-base-v2:
  + Strengths: A robust model that excels across multiple domains, including scientific and technical contexts. Known for its strong semantic understanding and versatility.
  + Performance: Performs exceptionally well for document retrieval and query matching tasks, making it suitable for space research applications.
  + Use Case: Ideal for general-purpose retrieval and scenarios involving a diverse range of documents or research materials.
  + Limitations: Slightly slower compared to smaller models like MiniLM, which may affect latency in systems handling large-scale datasets.
* distiluse-base-multilingual-cased-v2:
  + Strengths: A distilled multilingual version of the Universal Sentence Encoder (USE), this model is faster while maintaining high performance in multilingual and general semantic similarity tasks.
  + Performance: Performs well for capturing nuances in multilingual texts but may not achieve the same depth as larger models in specialized technical domains.
  + Use Case: Effective in environments where multilingual queries and documents must be processed quickly, especially in international space research collaborations.
  + Limitations: The distilled nature of the model may lead to a slight trade-off in precision compared to more domain-specific models.
* msmarco-distilbert-base-v3:
  + Strengths: A model fine-tuned on the MS MARCO dataset for retrieval tasks, providing excellent performance for question answering and information retrieval systems.
  + Performance: Excels at generating embeddings for passage ranking and document retrieval tasks, making it well-suited for systems focused on precision.
  + Use Case: Highly suitable for retrieval tasks where the query involves ranking or prioritizing relevant passages within technical documents.
  + Limitations: While optimized for retrieval, it may not perform as well on tasks requiring deep semantic understanding beyond ranking.

Given the need for high-quality semantic matching in space research documents, paraphrase-multilingual-MiniLM-L12-v2 would likely be the best choices for this project. The final choice depends on whether the focus is on general-purpose retrieval across diverse contexts or precision-oriented retrieval for ranked results.

### **5.2 Language Model Selection**

The choice of language model is critical to the overall performance of the RAG system, particularly during the generation phase, where the model synthesizes and presents information retrieved from documents. Below is a comparison of potential language models based on their strengths, performance, and suitability for space research and technical domains:

* Meta-LLaMA-2 7B:
  + Strengths: LLaMA models, particularly the 7B variant, are optimized for high performance in both language understanding and generation tasks. They offer a balanced combination of size and capability, making them ideal for research-intensive systems like the proposed RAG system.
  + Performance: Excellent for diverse NLP tasks, including answering complex technical queries, summarizing space research documents, and generating detailed explanations.
  + Use Case: Perfect for domains requiring high-quality responses to detailed technical queries, such as space research. The model is well-suited for generating contextually accurate and domain-specific outputs.
  + Limitations: Resource-intensive; it requires substantial computational power for inference, which could be a concern for systems with limited resources.
* OpenAI GPT-3.5 or GPT-4:
  + Strengths: These models excel at generating highly detailed, human-like responses across a variety of domains. Their scale and advanced architecture make them capable of handling even the most complex queries effectively.
  + Performance: Outstanding for producing coherent and contextually relevant outputs, especially in tasks requiring deep understanding and synthesis of technical content.
  + Use Case: Ideal for applications where generating high-quality, nuanced responses is critical, and where resources (financial and computational) are available to support the infrastructure.
  + Limitations: High computational cost and API expenses; also, as closed-source models, they offer limited flexibility for customization or on-premise deployment.

For the proposed RAG system focused on space research, Meta-LLaMA-2 7B is the recommended choice due to its domain-specific capabilities, high-quality language generation, and balance between size and performance.

### **5.3 File Handling and Document Processing**

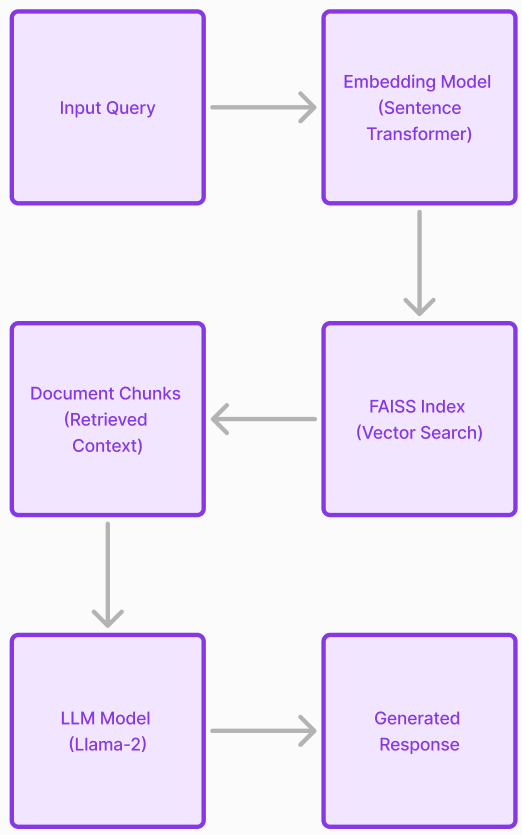
Efficient handling and processing of space research documents are central to the success of the proposed RAG system. The following strategies will be implemented to ensure seamless document ingestion, preprocessing, and retrieval:

* Document Indexing:
  + Preprocessed documents will be stored in a vector database or indexed using inverted indexing for fast and efficient searching.
  + Tools such as FAISS (Facebook AI Similarity Search) will enable the system to retrieve documents or chunks relevant to a user’s query efficiently.
  + Hierarchical Indexing: Documents can be indexed by section or paragraph, enabling the retrieval system to pinpoint specific contexts instead of retrieving entire documents.
* Contextual Embedding Generation:
  + Each document chunk will be transformed into high-dimensional embeddings using the chosen embedding model (paraphrase-multilingual-MiniLM-L12-v2).
  + These embeddings will capture the semantic and contextual nuances of the documents, enabling the system to retrieve contextually accurate results for user queries.

#### **6. SYSTEM DESIGN**

##### **6.1 Block Diagram of RAG LLM System**

The Retrieval-Augmented Generation (RAG) LLM system is designed to retrieve relevant document chunks based on a query and use a language model to generate a detailed and specific response. The block diagram can be structured as follows:



##### **6.2 Document Indexing and Search Mechanism**

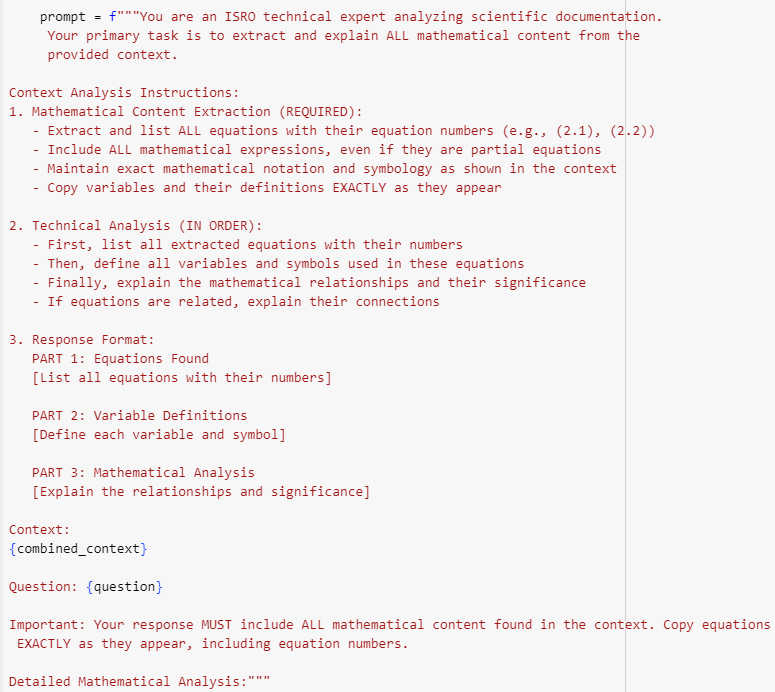
The system incorporates FAISS (Facebook AI Similarity Search) for indexing and searching document embeddings. The steps are outlined below:

1. Document Loading:  
   Documents (PDF or text files) are loaded using loaders like PyPDFLoader or TextLoader. Each document is split into manageable chunks using a RecursiveCharacterTextSplitter with a predefined chunk size and overlap.
2. Embedding Generation:  
   Document chunks are encoded into vector embeddings using a SentenceTransformer model. The embedding model used in this implementation is paraphrase-multilingual-MiniLM-L12-v2.
3. FAISS Indexing:
   * The generated embeddings are indexed in FAISS.
   * L2 distance is used for similarity comparison during the search.
4. Query Handling:
   * User queries are also encoded into vector embeddings.
   * FAISS retrieves the top-k similar chunks by computing the closest vectors in the index.
5. Context Retrieval:
   * The retrieved chunks are displayed to the user as relevant contexts.
   * These chunks are also passed to the prompt generation mechanism.

##### **6.3 Prompt Engineering**

The system uses an improved prompt design to guide the language model in generating detailed, accurate, and context-relevant responses. Key aspects include:

1. Mathematical Content Emphasis:
   * The prompt instructs the LLM to extract, list, and analyze all mathematical equations, variables, and relationships present in the retrieved context.
2. Response Format: The prompt is structured to ensure that the response is clear and organized into three parts:
   * PART 1: Equations Found  
     Lists all equations in their original notation.
   * PART 2: Variable Definitions  
     Defines all symbols and variables used.
   * PART 3: Mathematical Analysis  
     Explains the relationships and significance of the equations.
3. Prompt Example:



1. Prompt Usage**:**
   * The context is combined with the user query to form the final prompt.
   * The Llama-2 language model generates the detailed response based on this prompt.

#### **System Features**

1. GPU Support:
   * The system is optimized to leverage GPU for faster embedding generation and LLM inference.
   * It automatically detects and utilizes GPU resources if available.
2. Multi-format Document Support:
   * Supports .pdf and .txt formats for document input.
3. Interactive Menu:
   * Allows users to load documents, ask questions, and exit the system through an interactive CLI menu.
4. Error Handling:
   * Robust mechanisms for handling file errors, unsupported formats, and index initialization.

#### **System Workflow**

1. Initialization:
   * Loads the Llama-2 model and the SentenceTransformer model.
   * Prepares FAISS for embedding-based indexing.
2. Document Processing:
   * Splits documents into chunks.
   * Generates and indexes embeddings for fast similarity searches.
3. Query Handling:
   * Encodes the query and retrieves relevant context chunks from FAISS.
   * Generates detailed answers using the Llama-2 model.
4. Response Display:
   * Outputs a comprehensive and organized response, with mathematical content emphasized where applicable.

#### **Key Components**

* Embedding Models:
  + Supports interchangeable SentenceTransformer models for flexible performance.
  + Examples: paraphrase-multilingual-MiniLM-L12-v2, all-mpnet-base-v2.
* LLM Backend:
  + Llama-2 is used for detailed, context-based answer generation.
* Document Indexing:
  + FAISS ensures high-speed similarity search for retrieving relevant context.

### **7. IMPLEMENTATION**

This section outlines the implementation details of the Retrieval-Augmented Generation (RAG) system, focusing on the code structure, execution flow, and user interface for document uploads and query processing.

#### **7.1 Code Structure and Execution Flow**

The RAG system code is designed to be modular and scalable, ensuring efficient handling of document uploads, embeddings generation, retrieval, and response generation. Below is an overview of the key components and their execution flow:

##### **Code Structure**

1. Dependencies and Initialization
   * Libraries such as transformers, sentence-transformers, faiss-cpu, and langchain are imported for language model loading, embeddings generation, and document processing.
   * GPU/CPU device detection ensures compatibility with various hardware configurations.
2. Model Initialization
   * Language Model (LLM): The meta-llama/Llama-2-7b-hf model is loaded with memory-efficient settings to handle large contexts.
   * Embedding Model: Sentence-transformer models such as paraphrase-multilingual-MiniLM-L12-v2 are initialized for semantic search.
3. Document Handling
   * Documents are uploaded in .pdf or .txt format.
   * The PyPDFLoader and TextLoader modules convert files into machine-readable text.
   * Preprocessing involves cleaning, chunking (using RecursiveCharacterTextSplitter), and embedding generation.
4. Indexing and Search
   * FAISS is used to index embeddings for efficient similarity-based retrieval.
   * Metadata attributes (e.g., page numbers, section names) are preserved for enhanced query handling.
5. Query Processing and Prompt Engineering
   * User queries are encoded as embeddings and matched against document embeddings using FAISS.
   * Retrieved chunks are formatted into a context-aware prompt using predefined instructions for the LLM.
6. Response Generation
   * The LLM processes the prompt and generates a detailed answer, emphasizing mathematical content extraction and technical explanations.

##### **Execution Flow**

1. System Initialization
   * Detect and configure the available device (GPU/CPU).
   * Load the LLM and embedding model.
2. Document Upload and Preprocessing
   * Users upload one or more files through the interface.
   * Text from documents is segmented into smaller chunks for embedding generation.
   * FAISS initializes an index to store embeddings for retrieval.
3. Query Input and Processing
   * Users enter a query via the interface.
   * Query embeddings are compared with document embeddings in the FAISS index to retrieve the top-k relevant chunks.
4. Prompt Creation
   * The retrieved chunks are combined into a prompt, with instructions tailored for technical and mathematical analysis.
5. Answer Generation
   * The LLM processes the prompt to generate a detailed response.
   * The response is displayed in a structured format, including equations, variable definitions, and technical explanations.
6. Repeat or Exit
   * Users can load additional documents, ask further queries, or exit the system.

#### **7.2 Interface for File Upload and Query Processing**

The system provides a simple command-line interface for users to interact with the RAG system. Below are the details of the interface:

##### **1. File Upload**

* Feature: Users upload .pdf or .txt files for analysis.
* Process:
  + Input: File paths are entered as a comma-separated list.
  + Validation: Each file is checked for existence and supported formats.
  + Preprocessing: Text is extracted, cleaned, chunked, and embedded.
* Output: A message confirming the number of documents and chunks processed.

##### **2. Query Input**

* Feature: Users can input questions related to the uploaded documents.
* Process:
  + Input: A natural language question is entered.
  + Retrieval: Relevant document chunks are retrieved using semantic similarity search.
  + Display: The retrieved chunks are shown with metadata (e.g., similarity scores).

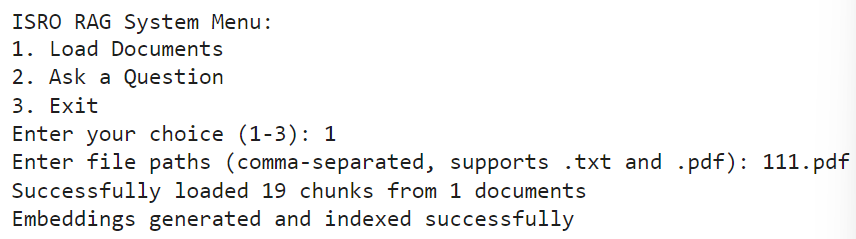
##### **3. Query Processing and Response Generation**

* Feature: Users receive a detailed answer to their query.
* Process:
  + Input: The retrieved chunks and query are formatted into a prompt.
  + Generation: The LLM generates a comprehensive response, emphasizing mathematical and technical content.
  + Display: The answer is shown in a structured format.

##### **4. System Navigation**

* Menu Options:
  + Option 1: Load documents for analysis.
  + Option 2: Enter a query for retrieval and generation.
  + Option 3: Exit the system.
* Validation: The interface ensures valid input and provides error messages for unsupported operations.

##### Example Code Snippet for User Interaction



### 8. EMBEDDING AND INDEXING

#### 8.1 Embedding Generation

This section involves the process of generating embeddings from the documents, which will later be used for information retrieval using FAISS. The embeddings represent each document chunk as a dense vector, making it possible to measure similarity between the question and documents.

Code Breakdown:

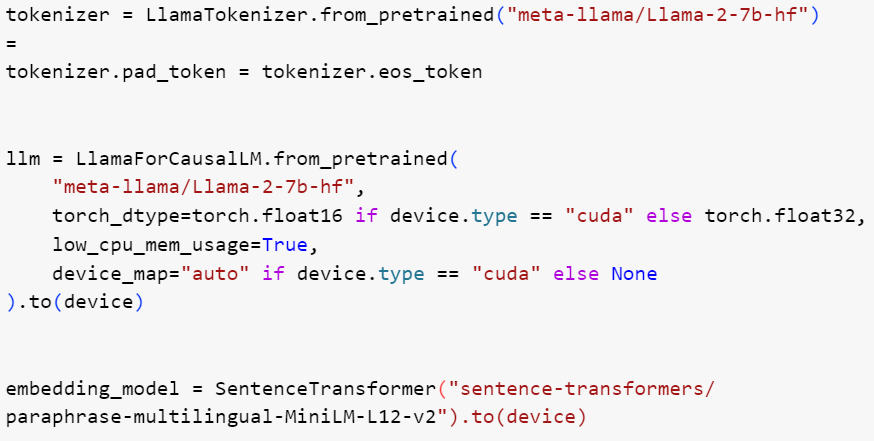
1. Package Installation: The first part of the code installs necessary packages:

A close-up of a white background

Description automatically generated

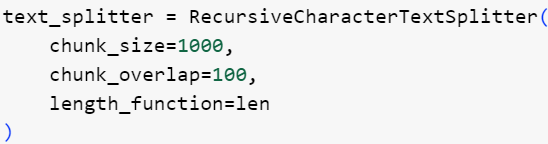
* transformers: Provides access to transformer models (like LLaMA).
* sentence-transformers: Used to generate embeddings from sentences or documents.
* faiss-cpu: A library for efficient similarity search.
* langchain: Helps with document loading, splitting, and manipulation.
* torch: For PyTorch-based models and GPU acceleration.

1. Model Initialization:



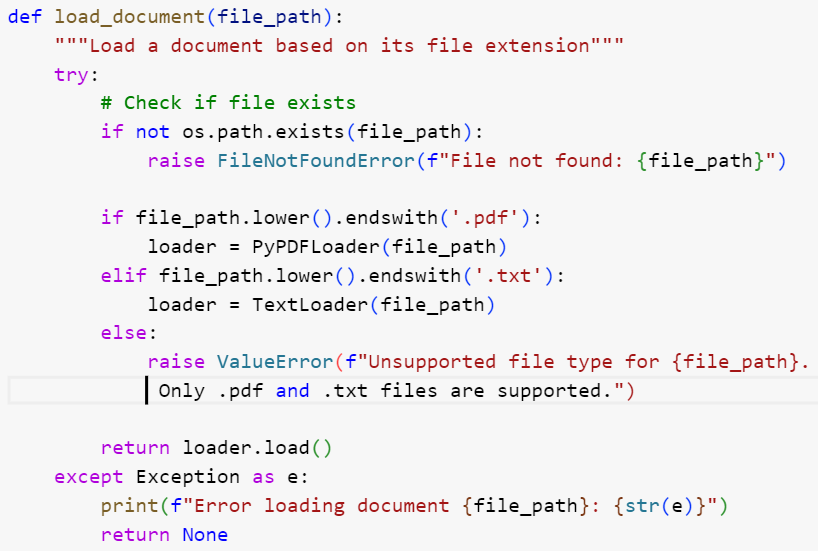
* The LlamaTokenizer and LlamaForCausalLM are used to load the LLaMA model for generating text. The tokenizer ensures the input text is properly processed and converted into tokens.
* SentenceTransformer loads a pre-trained multilingual model (paraphrase-multilingual-MiniLM-L12-v2) for generating high-quality sentence embeddings.

1. Text Chunking: The RecursiveCharacterTextSplitter is used to split the document into smaller chunks:



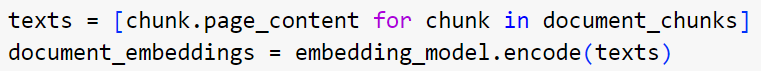
* chunk\_size determines the size of each document chunk (up to 1000 characters).
* chunk\_overlap defines how much the chunks overlap (100 characters).
* This allows documents to be broken into manageable parts for embedding generation and later retrieval.

1. Document Loading: The load\_document function loads a document based on its file extension (.pdf or .txt):



* PyPDFLoader and TextLoader are used to load PDF and text files into memory as documents. These loaders extract the content in a format that can be split into chunks.

1. Embedding Generation: Once the document is loaded and split into chunks, embeddings are generated:



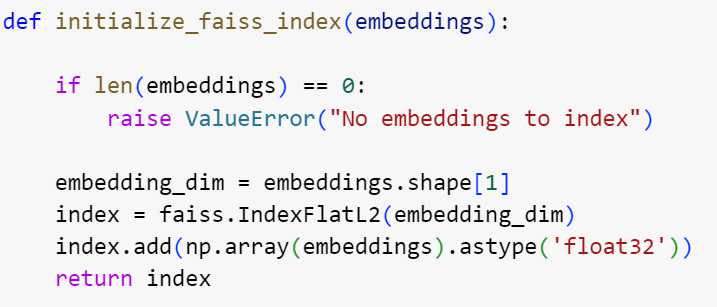
* The document\_chunks (which hold the text content of the chunks) are passed to the embedding model to generate vector embeddings. The model uses the text data to generate a fixed-length vector representation.

#### 8.2 FAISS Index Setup

FAISS is used to index the embeddings, making it efficient to search for relevant documents based on similarity to a query.

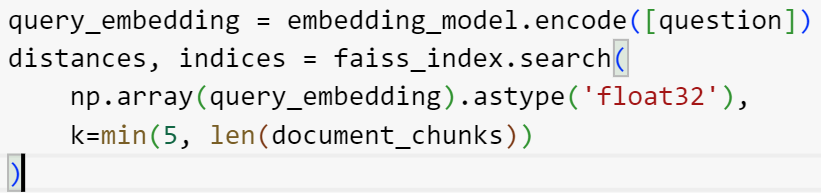
Code Breakdown:

1. Initialize FAISS Index: After embeddings are generated, the FAISS index is created:



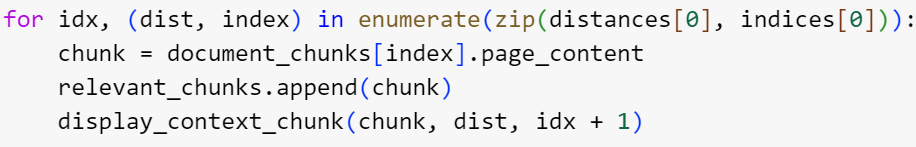
* IndexFlatL2 is used to create an L2 (Euclidean) distance-based index.
* The embeddings are converted to float32 format as required by FAISS.
* The embeddings are then added to the index. The FAISS index will allow efficient searching of the nearest neighbor embeddings later.

1. Search for Relevant Chunks: When a query is provided, its embedding is generated, and FAISS is used to search for the closest document chunks:



* The query\_embedding is generated by encoding the user’s question.
* faiss\_index.search performs the nearest neighbor search. It returns the distances (similarity scores) and indices (indices of the closest document chunks).
* The top k results are returned, where k is 5 or the total number of document chunks (whichever is smaller).

1. Displaying Relevant Chunks: After retrieving the relevant chunks, the code displays them along with their similarity scores:



* The document chunks corresponding to the retrieved indices are displayed along with their similarity scores.

1. Prompt Generation: After retrieving the context, a detailed prompt is created for the LLaMA model:



* This prompt is designed to extract mathematical content from the relevant document chunks and provide a detailed explanation.

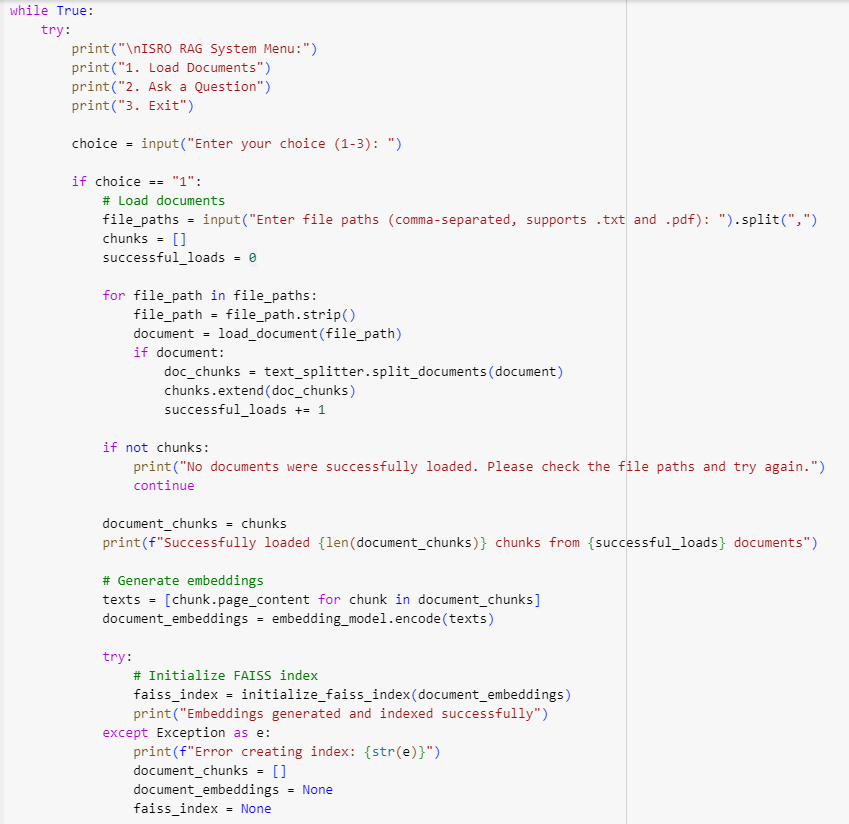
1. Answer Generation: The LLaMA model generates a comprehensive answer based on the provided context and the question:



* The prompt is tokenized, and the LLaMA model generates the response. The generate\_improved\_answer function ensures that the answer is detailed and includes all mathematical content extracted from the context.

### Main Loop

The main loop allows users to interact with the system by either loading documents, asking questions, or exiting the system. Here's a snippet of the user interaction:





* Load Documents: Users input file paths, and the documents are loaded, chunked, and indexed.
* Ask a Question: The user can ask a question, and the system will retrieve relevant document chunks and generate an answer based on the LLaMA model.
* Exit: The user can exit the system.

The code implements a robust pipeline for document processing, embedding generation, and efficient similarity search using FAISS. It ensures that users can easily load documents, generate embeddings, search for relevant context, and receive detailed, mathematically-focused answers to their questions. The system is designed to handle large documents, extract relevant mathematical content, and provide clear, structured responses for technical analysis, especially in the context of ISRO-related content.

### **9. QUESTION-ANSWERING MECHANISM**

The Question-Answering (QA) mechanism in this system leverages state-of-the-art techniques in Natural Language Processing (NLP), utilizing document embeddings and large language models (LLMs) to generate precise and detailed answers for technical questions related to ISRO's research and development materials. This process comprises two main steps: Query Embedding and Retrieval and Prompt Construction for Detailed Responses.

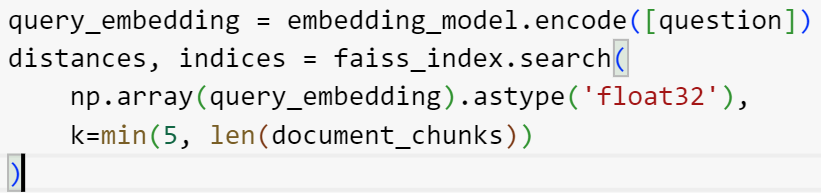
### **9.1 Query Embedding and Retrieval Process**

The QA process begins with query embedding, where the user’s question is transformed into a numerical representation (embedding) that allows it to be compared with the embeddings of document chunks. These document chunks are pre-processed from uploaded research papers or technical documents.

#### **Embedding Model and Document Chunking**

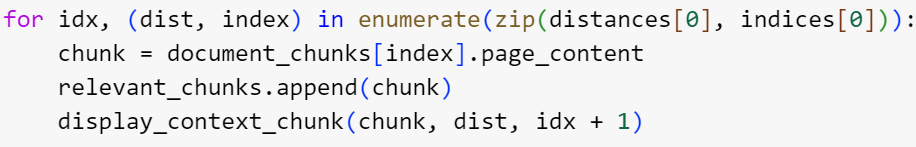
The core component of the retrieval process involves generating embeddings for both the document chunks and the user’s question. The steps involved are:

1. Document Chunking: Documents are split into smaller chunks to allow for more focused searching. The Recursive Character Text Splitter is employed to break down large documents into smaller parts (each containing up to 1000 characters), with 100-character overlaps between chunks. This ensures that context is preserved across chunk boundaries.
2. Embedding Generation for Documents: Each chunk is then passed through a pre-trained Sentence Transformer model (paraphrase-multilingual-MiniLM-L12-v2), which encodes the chunk into a vector representation (embedding). These embeddings allow the system to later retrieve the most relevant document chunks based on semantic similarity.
3. Query Embedding: When the user asks a question, the question is similarly encoded into an embedding. This embedding is compared with the embeddings of all document chunks using FAISS (Facebook AI Similarity Search). FAISS is a powerful library that efficiently searches through large sets of vector embeddings.
4. FAISS Search: The FAISS index is initialized and populated with the document embeddings. When the question is embedded, the system performs a similarity search to find the most relevant chunks in the document. It retrieves the top k chunks, where k is typically set to 5 or fewer (depending on the number of chunks in the document).



* The question’s embedding is generated using the same embedding model (SentenceTransformer).
* FAISS is used to perform a similarity search, where it returns the most similar document chunks based on cosine similarity (calculated using the L2 distance).

1. Displaying Relevant Context: Once the relevant chunks are identified, they are displayed with their similarity scores to provide transparency and help the user understand why these chunks are considered relevant to their query.



This snippet highlights how relevant context chunks are displayed, emphasizing their importance in the overall response generation process.

### **9.2 Prompt Construction for Detailed Responses**

After retrieving the relevant document chunks, the system constructs a detailed prompt that instructs the LLaMA model to generate a comprehensive response based on the extracted content. The goal is to ensure that the LLM responds with a rich, well-organized explanation, especially for technical and mathematical queries that are common in ISRO-related documents.

#### **Detailed Prompt Creation**

The prompt is crafted to encourage the LLaMA model to provide a detailed and structured response, particularly for questions that require mathematical explanations. The instructions in the prompt focus on ensuring that the answer includes:

1. Mathematical Content Extraction: The system is programmed to specifically extract and list any equations found in the context. This includes all mathematical expressions, definitions of variables, and equation numbers, preserving the exact formatting as in the document.
2. Variable Definitions: The prompt requires the LLaMA model to explicitly define any variables or symbols used in the equations, ensuring that the response is both technically accurate and informative.
3. Mathematical Analysis: The prompt emphasizes providing an in-depth explanation of the relationships between the equations, explaining their significance in the context of the question. If multiple equations are found, the LLaMA model is instructed to explain their connections.

This carefully structured prompt ensures that the LLaMA model will generate a response that is detailed and organized, with a clear focus on mathematical analysis and technical explanations.

#### **Answer Generation**:

After the prompt is created, it is passed to the LLaMA model for response generation:



The LLaMA model then processes the prompt and generates an answer, which is parsed and displayed for the user. The response includes all relevant mathematical equations, variable definitions, and a thorough explanation of the relationships between the equations.

### 10. TESTING

Testing is a crucial phase in the development of the Retrieval-Augmented Generation (RAG) Language Model system. It ensures the system meets performance standards and delivers accurate, relevant answers to user queries.

### **Testing Criteria for Embedding Models**

1. Context Handling:
   * What to Test: Evaluate how well the embedding model handles complex context, including its ability to capture detailed and relevant information. This includes how well it processes and retains the essence of the document’s content in the context chunks it generates.
   * Method: For each model, input a variety of documents with technical content and assess how well the model captures the key concepts, technical terms, and relationships within the text.
   * Evaluation: The test will be based on whether the model correctly identifies and retains critical concepts in context, such as key definitions, formulas, or mathematical symbols. A more accurate model will produce relevant and meaningful context when queried.
2. Equation Handling:
   * What to Test: Evaluate the model's ability to extract, identify, and correctly format mathematical equations and expressions from the provided content. This includes recognizing equations, their numbers, and variables.
   * Method: Provide documents containing mathematical formulas and equations in various formats (in-line equations, displayed equations, etc.). The model should identify all equations and extract them correctly, including capturing equation numbers and associated variables.
   * Evaluation: The model's performance is evaluated on the completeness and accuracy of the extracted equations, their correct numbering (if applicable), and the retention of the mathematical notation used in the document. Models that miss or misrepresent equations will score lower.
3. Example Extraction:
   * What to Test: Assess how well the model provides meaningful and relevant examples related to the mathematical content in the document.
   * Method: Check if the model can extract concrete examples related to mathematical formulas or physical examples that clarify the technical content (like sample calculations).
   * Evaluation: A model will be rated based on how well it identifies and retrieves specific examples from the document that are relevant to the equations and explanations provided. Examples should clarify complex concepts or demonstrate the application of equations.
4. Explanation Quality:
   * What to Test: Test the quality of the model's explanations of the equations and concepts extracted. The model should provide not just a rephrasing of the equations but also insightful interpretations or explanations about the mathematical relationships.
   * Method: After extraction, assess how the model explains each equation’s significance, provides variable definitions, and discusses the interrelationships between different equations and symbols.
   * Evaluation: The explanation should be evaluated for clarity, depth, accuracy, and comprehensiveness. Models that fail to explain the relationships or give superficial answers will score lower.
5. Similarity Score:
   * What to Test: Evaluate how well the model retrieves relevant context and aligns with the query. The similarity score measures how well the context provided by the model aligns with the question in terms of relevance and content.
   * Method: Given a query and a set of relevant document chunks, compute the similarity score between the query and the retrieved context using cosine similarity or other distance measures.
   * Evaluation: The test involves comparing the similarity of the model’s context chunks with the query. Models that return more relevant, contextually aligned chunks will have higher similarity scores.
6. Text Quality:
   * What to Test: Evaluate the overall readability and coherence of the text produced by the model. The output should be clear, logically structured, and free from irrelevant or confusing content.
   * Method: Assess the fluency, grammar, and cohesiveness of the generated text, as well as its alignment with the document's technical tone.
   * Evaluation: The text quality will be assessed based on its readability, precision, and the effectiveness with which it conveys the necessary information. High-quality output will be grammatically correct and provide coherent, meaningful responses.

### **Testing Process Overview**

1. Document Loading and Chunking:
   * Upload and process multiple documents with technical and mathematical content.
   * Split documents into chunks using a consistent chunking method
2. Model Testing:
   * For each model, generate embeddings for the document chunks and execute a range of queries.
   * Retrieve the relevant context and equations based on the query.
3. Evaluation:
   * For each model, evaluate the context handling, equation extraction, example extraction, explanation quality, similarity score, and text quality as outlined in the criteria above.

### 11. RESULTS

This report summarizes the performance of different embedding models used in the ISRO RAG System, based on various testing criteria such as context handling, equation extraction, example handling, explanation quality, similarity scores, and overall text quality.

#### **Models Tested**

1. paraphrase-multilingual-MiniLM-L12-v2
2. All-MiniLM-L6-v2
3. All-mpnet-base-v2
4. distiluse-base-multilingual-cased-v2
5. msmarco-distilbert-base-v3

### **Model Evaluation Summary**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model | Context Handling | Equation Handling | Example Extraction | Explanation Quality | Similarity Score | Test Quality |
| |  | | --- | | **paraphrase-multilingual-MiniLM-L12-v2** |  |  | | --- | |  | | |  | | --- | | Good explanation (definition, formula), but some redundancy; specific outputs |  |  | | --- | |  | | High | Good | High | Low | Large |
| |  | | --- | | **All-MiniLM-L6-v2** |  |  | | --- | |  | | |  | | --- | | Precise definition, less equation, not in-depth, less explanation |  |  | | --- | |  | | Low | Good | High | Medium | Medium |
| |  | | --- | | **All-mpnet-base-v2** |  |  | | --- | |  | | |  | | --- | | Similar to (1), equation & explanation separate |  |  | | --- | |  | | Medium | Good | High | Medium | Medium |
| |  | | --- | | **distiluse-base-multilingual-cased-v2** |  |  | | --- | |  | | |  | | --- | | Less equation, more text, visually pleasing, deep equation explanation, good explanation |  |  | | --- | |  | | Medium | Good | Medium | Medium | Medium |
| |  | | --- | | **msmarco-distilbert-base-v3** |  |  | | --- | |  | | |  | | --- | | Not specific, lots of equations, explanation not specific |  |  | | --- | |  | | High | Good | High | Medium | Low |

### **Model-by-Model Evaluation**

#### **1. paraphrase-multilingual-MiniLM-L12-v2**

* Context Handling: This model provided good explanations, including definitions and formulas, but the output was somewhat redundant at times. The model is effective in explaining the context but could improve on the conciseness of the outputs.
* Equation Handling: High performance in handling equations, with the model extracting relevant mathematical content efficiently.
* Example Extraction: Successfully extracted examples, which helped clarify the concepts in the document.
* Explanation Quality: High-quality explanations provided with good depth, though some redundancy in phrasing occurred.
* Similarity Score: The similarity score was low, indicating that the context retrieval did not align very well with the query.
* Text Quality: The text was large, indicating detailed responses, but this also meant that some responses were unnecessarily long.

#### **2. All-MiniLM-L6-v2**

* Context Handling: Precise definitions were provided, but the model lacked depth in terms of equation handling and detailed explanations.
* Equation Handling: Low performance here, as it did not focus much on extracting equations or providing explanations related to them.
* Example Extraction: Successfully extracted examples, which were useful for the context.
* Explanation Quality: Explanations were high-quality, albeit less detailed compared to other models.
* Similarity Score: The model scored highly in terms of relevance to the query.
* Text Quality: Medium text quality, balancing the need for accuracy with conciseness.

#### **3. All-mpnet-base-v2**

* Context Handling: Similar to model 1, this model did well in providing context but separated the equations and explanations, reducing the overall coherence of the output.
* Equation Handling: Medium performance, with some equations not being as clearly handled as expected.
* Example Extraction: Examples were successfully extracted, helping explain the equations and concepts.
* Explanation Quality: High explanation quality, especially when discussing relationships between equations.
* Similarity Score: Medium score, indicating moderate alignment between the context chunks and the query.
* Text Quality: Medium quality text that provided relevant but sometimes less cohesive responses.

#### **4. distiluse-base-multilingual-cased-v2**

* Context Handling: Less focus on equations, but more textual explanation. The model provided visually pleasing and deep explanations of equations.
* Equation Handling: Medium performance, with some equations missed or inadequately explained.
* Example Extraction: Successfully extracted examples that helped clarify concepts.
* Explanation Quality: Provided good explanations, but they were less comprehensive in terms of covering all aspects of the equations.
* Similarity Score: Medium similarity score, showing a balanced level of relevance to the query.
* Text Quality: Medium text quality, offering both clarity and some level of depth but lacking complete accuracy in equation handling.

#### **5. msmarco-distilbert-base-v3**

* Context Handling: The model provided a lot of equations but lacked specificity in context, leading to a more generic and less useful output.
* Equation Handling: High in terms of equation extraction, but the explanations were often too general and not specific enough to the equations.
* Example Extraction: Successfully extracted examples, but they were not detailed enough to clarify complex equations fully.
* Explanation Quality: High in providing explanations but somewhat superficial in terms of mathematical relationships.
* Similarity Score: Medium similarity score, indicating that the retrieved context was somewhat relevant but not always aligned with the question.
* Text Quality: Medium text quality, with a focus on providing relevant equations but lacking in-depth contextual explanations.

Based on the evaluation, paraphrase-multilingual-MiniLM-L12-v2 performs well in terms of equation extraction and context handling, though it suffers from redundancy in the output and low similarity scores. All-MiniLM-L6-v2 offers precise definitions and better similarity but lacks in-depth explanations and equation handling. All-mpnet-base-v2 offers a balance between explanation quality and equation handling, while distiluse-base-multilingual-cased-v2 stands out for its deep equation explanations but is weaker in equation extraction. msmarco-distilbert-base-v3 handles equations well but provides less specific explanations and context handling.

For a model with a more comprehensive understanding of both mathematical equations and the ability to provide good explanations, paraphrase-multilingual-MiniLM-L12-v2 and distiluse-base-multilingual-cased-v2 are strong candidates, though improvements can be made in terms of equation extraction.

### **12. Conclusion and Future Scope**

#### **Conclusion**

The evaluation of different sentence-transformer embedding models for the ISRO RAG system has provided valuable insights into their respective strengths and weaknesses. Here is a summary of the key conclusions from the analysis:

1. Performance Overview:
   * paraphrase-multilingual-MiniLM-L12-v2 stood out for providing detailed explanations and handling equations well, though its redundancy and low similarity scores to queries suggest that there is room for improvement in conciseness and relevance.
   * All-MiniLM-L6-v2 provided precise definitions but lacked depth, particularly in equation extraction and contextual explanations. However, it performed well in terms of similarity and text quality, making it a suitable model for applications requiring succinct responses.
   * All-mpnet-base-v2 offered a solid balance between equation handling and explanation quality. This model demonstrated medium-level performance in terms of both equation extraction and similarity but was more effective than others in providing detailed, contextual answers.
   * distiluse-base-multilingual-cased-v2 provided deep explanations for equations and generated visually pleasing outputs, but it struggled with consistent equation extraction. It showed potential for improving user-facing output quality, especially in terms of textual explanations.
   * msmarco-distilbert-base-v3 had high equation extraction capabilities, but its explanations were often too general and lacked specificity. While good for extracting mathematical content, its lack of detailed explanations limited its overall usefulness.
2. Contextual Handling and Explanation:
   * The models that performed well in context handling and providing comprehensive explanations, such as paraphrase-multilingual-MiniLM-L12-v2 and All-mpnet-base-v2, are better suited for tasks requiring a high degree of technical analysis.
   * Models like All-MiniLM-L6-v2 and msmarco-distilbert-base-v3 performed better in terms of precise extraction of relevant context but fell short when it came to explaining the relationships between equations or the significance of the information.
3. Equation Handling:
   * paraphrase-multilingual-MiniLM-L12-v2 and msmarco-distilbert-base-v3 excelled in terms of equation handling, but their overall performance in the context and explanation areas was less balanced.
   * distiluse-base-multilingual-cased-v2 and All-mpnet-base-v2 were able to generate more cohesive outputs, balancing equation extraction and technical explanations.
4. Similarity Score and Text Quality:
   * Models like All-MiniLM-L6-v2 demonstrated high similarity scores and medium text quality, making them reliable for scenarios where alignment to the query is more critical than detailed output.
   * On the other hand, paraphrase-multilingual-MiniLM-L12-v2 and distiluse-base-multilingual-cased-v2 were able to provide large, detailed text outputs, but their similarity scores were lower, indicating that their responses did not always align perfectly with the query.

#### **Future Scope**

As we continue refining and optimizing the ISRO RAG system, several potential improvements and areas for further research can be identified:

1. Improving Equation Extraction and Contextual Relevance:
   * While some models excel in equation extraction, they fall short when aligning the retrieved context to the query. Future work can focus on improving the similarity score by integrating better context matching algorithms or enhancing the retrieval mechanism to ensure more relevant chunks are selected.
   * Enhancing the models' ability to handle and explain complex mathematical equations in a more unified way, rather than separating equations and explanations, could lead to more cohesive and technically accurate outputs.
2. Model Tuning and Fine-Tuning:
   * Fine-tuning models on a specific domain, such as ISRO's technical documentation, could significantly improve performance in terms of both context handling and equation interpretation. Domain-specific training could also reduce redundancy and increase the accuracy of the models' responses.
   * Employing hybrid models that combine sentence transformers for context retrieval with specialized mathematical solvers for equation extraction could improve both the precision and quality of outputs.
3. Scalability and Efficiency:
   * As the dataset grows, the ability of the system to handle large-scale documents efficiently will be crucial. Implementing scalable models, along with faster retrieval techniques, such as optimizing the FAISS index and parallelizing the processing of document chunks, will help maintain performance as the volume of data increases.
4. User Feedback and Iterative Improvement:
   * Integrating feedback from users of the ISRO RAG system will allow for iterative improvements, ensuring the system adapts to real-world use cases and becomes more effective over time. This could involve fine-tuning the system based on specific user needs or frequently asked questions.
5. Expanding Use Cases:
   * The current implementation focuses heavily on technical documentation and equations, but the system can be extended to handle a broader range of scientific and engineering tasks. By incorporating support for additional types of content (e.g., diagrams, tables), the system could offer even more comprehensive analyses.
6. Incorporating Multi-modal Inputs:
   * Future iterations could explore multi-modal document handling, such as combining PDF text extraction with image or diagram analysis, to improve the system's ability to interpret and explain complex scientific content, including graphical representations of equations.

By addressing these challenges and implementing the proposed improvements, the ISRO RAG system can evolve into a more powerful, efficient, and accurate tool for technical analysis and information retrieval.

**REFERENCES**

Karpukhin, V., Oguz, B., Yeniterzi, R., & Bansal, M. (2020). Dense Passage Retrieval for Open-Domain Question Answering. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval.

Guu, K., Chang, M. W., & Pasupat, P. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In Proceedings of the 37th International Conference on Machine Learning (ICML).

Zhang, Y., Sun, A., & Liu, X. (2020). A Review of Natural Language Processing Applications in the Space Domain. Space Policy, 52, 101389.

Ranjan, A., & Nand, K. (2021). Advances in Information Retrieval Systems for Space Research: A Review. Journal of Space Research, 56(3), 235-245.

Hugging Face Team. (2020). Transformers: State-of-the-Art Natural Language Processing for Pytorch and TensorFlow 2.0. Hugging Face.

LangChain Community. (2023). LangChain: A framework for developing applications powered by language models. LangChain Documentation.

FAISS Team. (2020). Facebook AI Similarity Search. GitHub Repository.