Inclusive Offline Multimodal Retrieval-Augmented Generation System for Accessible PDF-Based Knowledge Assistance

Bhuvanesh. V., Balavigneshwaran. SN., Mrs. Margaret Flora B., Lakshman. B., Muthu Sundaram B

Department of Artificial Intelligence and Machine Learning R.M.D. Engineering College Chennai, India

DOI: https://doi.org/10.51584/IJRIAS.2026.11070091

Received: 19 July 2026; Accepted: 24 July 2026; Published: 06 August 2026

ABSTRACT

People with visual, speech and hearing impairment still face a major problem of receiving digital knowledge. In spite of the fact that the artificial intelligence enhances the information retrieval systems, the majority of the solutions are based on the cloud-based large language models and they do not offer an inclusive multimodal interaction. In this paper, an Offline Multimodal Retrieval-Augmented Generation (RAG) System is introduced that is intended to help differently-abled users to interact with PDF documents with the help of text, speech, and sign-language. The suggested system consists of the locally run large language model (LLaMA through Ollama), semantic retrieval based on FAISS, offline speech recognition, text-to-speech synthesis, and Sign animation rendering through gesture recognition. Our architecture is based on privacy, low latency and free deployment, unlike the traditional cloud-dependent AI assistants. Experimental evaluation demonstrates that the system effectively retrieves context-relevant responses while supporting voice interaction and visual gesture assistance. The proposed framework contributes toward inclusive AI-driven knowledge systems and demonstrates the feasibility of offline assistive intelligence platforms.

Keywords: Accessibility AI, Retrieval-Augmented Generation, Offline LLM, Assistive Technology, Multimodal Interaction, Inclusive Computing

INTRODUCTION

The rapid digitalization of information has fundamentally transformed the way knowledge is created, stored, disseminated, and accessed. Across academic institutions, government organizations, healthcare facilities, and industrial sectors, an increasing proportion of information is now distributed in the form of Portable Document Format (PDF) files due to their platform independence, document integrity, and ease of sharing. Research articles, textbooks, policy manuals, legal documents, medical reports, technical specifications, and administrative records are routinely published in PDF format, making it one of the most widely adopted standards for digital documentation. While this digital transformation has significantly improved information availability and reduced dependence on printed materials, it has not fully addressed the accessibility requirements of individuals with disabilities.

People with visual impairments, hearing impairments, speech disabilities, and motor impairments continue to experience considerable challenges when interacting with digital documents. Conventional PDF readers primarily support manual navigation through scrolling, page selection, and sequential reading, requiring users to visually inspect and interpret document contents. Such interaction mechanisms are often unsuitable for users who depend on assistive technologies for accessing information. Complex document layouts, embedded images, tables, mathematical equations, and multi-column formatting further increase the difficulty of extracting meaningful information, thereby limiting equal access to digital knowledge.

Existing assistive technologies, including screen readers, Optical Character Recognition (OCR) systems, speech-to-text applications, and text-to-speech (TTS) engines, have improved accessibility to some extent. Screen readers convert textual information into synthesized speech, enabling visually impaired users to listen to document content, while speech recognition systems facilitate hands-free interaction through voice commands. However, these technologies generally process documents in a linear and sequential manner

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue VII July 2026

without understanding the semantic relationships among document sections. Consequently, users are often required to listen to large portions of a document before locating the desired information, resulting in reduced efficiency and increased cognitive effort. Furthermore, these systems are unable to provide intelligent responses to user queries or summarize document contents based on contextual understanding.

Recent advancements in Artificial Intelligence, particularly Large Language Models (LLMs), have revolutionized natural language understanding and document comprehension. Modern LLMs possess remarkable capabilities for text generation, summarization, question answering, translation, and contextual reasoning. Nevertheless, standalone language models are susceptible to hallucinations, wherein they generate plausible yet factually incorrect responses due to the absence of reliable external knowledge sources. To overcome this limitation, Retrieval-Augmented Generation (RAG) has emerged as a promising framework that integrates information retrieval with generative language models. Instead of relying solely on pre-trained knowledge, RAG systems retrieve the most relevant document fragments from an external knowledge repository and use them as contextual evidence during response generation. This retrieval-based grounding significantly enhances response accuracy, reduces hallucinations, and provides contextually relevant answers that are directly supported by the source documents.

Despite these advantages, the majority of existing RAG implementations depend heavily on cloud-hosted Large Language Models accessed through commercial Application Programming Interfaces (APIs). Such cloud-based architectures introduce several practical limitations. First, they require continuous internet connectivity, making them unsuitable for environments with unreliable or restricted network access. Second, transmitting sensitive documents to external servers raises serious concerns regarding data privacy, confidentiality, and regulatory compliance. Academic records, medical reports, financial documents, legal files, and government records often contain confidential information that should remain within institutional boundaries. Third, commercial API services involve recurring operational costs that may become prohibitively expensive for educational institutions, public organizations, and small enterprises. These limitations restrict the widespread adoption of intelligent document assistance systems in privacy-sensitive and resource-constrained environments.

The growing emphasis on privacy-preserving Artificial Intelligence, edge computing, and local AI deployment has created significant interest in offline intelligent systems capable of executing advanced language models directly on local hardware. Running language models locally eliminates the need to transmit confidential information to external cloud services, thereby ensuring complete ownership and control of sensitive data. Offline deployment also reduces operational costs associated with API subscriptions, minimizes latency, improves system responsiveness, and guarantees uninterrupted functionality even in the absence of internet connectivity. Such characteristics make locally deployed AI systems particularly suitable for educational institutions, healthcare organizations, government agencies, libraries, research laboratories, and other environments where privacy and continuous accessibility are essential.

In addition to privacy preservation, multimodal human-computer interaction has become increasingly important for designing inclusive assistive technologies. Users with different disabilities often require multiple interaction modalities, including voice commands, speech synthesis, and intelligent conversational interfaces. Integrating speech recognition with natural language understanding enables users to interact with documents through spoken queries rather than conventional keyboard or mouse inputs. Likewise, text-to-speech technology provides audible responses that allow visually impaired users to access document information naturally and efficiently. Combining these technologies with Retrieval-Augmented Generation creates an intelligent assistant capable of understanding user intentions, retrieving relevant document sections, and generating accurate, context-aware responses while maintaining complete offline functionality.

Motivated by these challenges, this research proposes an **Inclusive Offline Multimodal Retrieval-Augmented Generation (RAG) System** designed to improve digital document accessibility for individuals with visual, hearing, and speech impairments. The proposed framework combines semantic document understanding, efficient information retrieval, locally deployed Large Language Models, and multimodal interaction to provide secure, intelligent, and privacy-preserving document assistance without relying on cloud infrastructure. By integrating modern retrieval techniques with offline language models and assistive

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue VII July 2026

communication technologies, the proposed system aims to deliver an accessible, reliable, and cost-effective solution for intelligent PDF interaction.

The major contributions of the proposed framework include:

- **Local PDF Processing:** Extraction, preprocessing, and semantic chunking of PDF documents to preserve contextual relationships and improve information retrieval accuracy.
- **Semantic Embedding Generation:** Conversion of document chunks into dense vector representations using transformer-based embedding models for meaningful semantic search.
- **FAISS-Based Vector Retrieval:** Efficient indexing and similarity search using Facebook AI Similarity Search (FAISS) to retrieve the most relevant document passages with low latency.
- **Offline Large Language Model Integration:** Deployment of locally hosted LLaMA models through a local inference runtime to generate context-aware responses while ensuring complete privacy and eliminating internet dependency.
- **Voice-Based User Interaction:** Speech recognition functionality that enables users to submit natural language queries through voice commands, improving accessibility for users with motor and visual impairments.
- **Speech-Based Response Generation:** Integration of text-to-speech technology to provide audible responses, enabling visually impaired users to receive document information without requiring visual interaction.
- **Privacy-Preserving Intelligent Assistance:** Complete offline execution of document retrieval and response generation, ensuring that confidential documents remain within the local computing environment.
- **Inclusive Multimodal Accessibility:** A unified framework supporting text, speech input, semantic search, and voice feedback to create an accessible and user-friendly document interaction system for differently abled individuals.

The proposed system addresses the limitations of existing cloud-dependent document question-answering solutions by providing a secure, intelligent, and fully offline framework that combines Retrieval-Augmented Generation, semantic document retrieval, local Large Language Models, and multimodal accessibility technologies. The framework has the potential to significantly enhance digital inclusion by enabling users with diverse accessibility needs to access, understand, and interact with PDF documents more effectively while maintaining privacy, reducing operational costs, and ensuring uninterrupted availability.

Related Work

Research relevant to an offline, multimodal RetrievalAugmented Generation **(RAG)** assistant for accessibility spans four partially overlapping areas: (1) transformer-based language modeling and local deployment, (2) retrieval and vector search, (3) offline speech processing for voice I/O, and (4) sign-language synthesis and avatar systems. Below we review recent progress (2024–2025) in each area and identify gaps that the present work addresses.

A. Transformer models and local/offline deployment

The fundamentals of transformer architectures continue to form the core of current systems of language understanding and language generation. The transformer formulation originally proposed self-attention instead of recurrence and allowed highly parallel sequence modeling and eventually led to large language models (**LLMs**) used in generative tasks. This trend was further continued by recent large-scale LLM families (especially the LLaMA lineage) that were more capable and allowed more feasible local deployment patterns**.**

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue VII July 2026

LLaMA3 family and 3.1 announcements by Meta show scaled model capability and bigger context windows at released with developer-friendly licensing and tooling, making it possible to use on-premise. These developments make our offline design objective possible directly by offering competitive open-model designs that can be hosted without frequent cloud access. The transformer paradigm was pioneered by Ashish Vaswani, and commercial/open releases like **LLaMA** 3 have triggered real-world uses of LLM offline.

Similar to the release of models, there has been a number of tooling projects to help run open models locally with ease. Ollama offers a basic local runtime and tool-calling integrations which enable developers to run models on personal computers and call them with programmatic APIs; recent product releases in 2024 extended model/tool support and added desktop GUIs that further reduced the barrier to offline deployment. These pragmatic toolchains are the core of our implementation decision, as well as the larger movement of privacy-sensitive, on-device LLM applications.

B. Retrieval-Augmented Generation and vector retrieval

RAG architectures support document retrieval and conditional generation so that answers are based on external knowledge as opposed to depending on parametric model memory alone; this minimizes hallucination and is more accurate to the domain. Even more recent surveys (2024) summarize a motivated rapid advance in the development of **RAG** approaches, including enhanced dense retrieval, context selection methods, reranking, and hybrid retrieval-generation pipelines. Open challenges, such as scalability, latency, evaluation metrics, and bias mitigation that these surveys bring directly to bear on the priorities of our system design (efficient FAISS indices, small-context grounding, and careful retrieval prompting), also feature.

The key element to scalable vector search is **FAISS** which is a highly optimized library of similarity search and cluster index. FAISS is the practical library of indexing on both CPU and GPU, and is not only being actively maintained with faster indexing and more economical indexes, but is also compatible with the ecosystem (Python, C++) to form a document-level **RAG** pipeline that must execute offline. FAISS

C. Offline speech recognition and synthesis

High-quality offline voice I/O is crucial for visually impaired users and for hands-free interaction. Self-supervised and transformer-based speech encoders (wav2vec family and derivatives) have underpinned recent progress in robust speech recognition. Works in 2024 proposed streaming-friendly adaptations of wav2vec (e.g., wav2vec-S) that improve latency

and adapt pre-trained models for lower-resource, real-time scenarios—properties important for our on-device voice input pipeline. For speech output, modern offline TTS engines (e.g., pyttsx3 integrations or local neural TTS models) provide usable, low-latency synthesis suitable for assistive applications where internet access is not available. Together, these advancements enable reliable on-device speech I/O for our assistant.

D. Sign-language synthesis and avatar rendering

To provide a sign-language output to facilitate the use of deaf users, it is necessary to map text or key semantic units to temporally precise and understandable gestures sequences. The most recent 2024 studies have contributed to both data and models of sign language production: Sign Avatars, a large scale 3D sign language motion dataset and benchmark, presented at ECCV 2024, has made a significant contribution to the quantity of available training data, and 3D avatars may now be generated by diffusion-based 3D sign production, poster/work shown at CVPR 2024. Further ECCV 2024 work investigated direct Spoken2Sign translation pipelines: and baseline systems that can be used to translate speech to sign output using 3D avatars. With these contributions, realistic sign rendering is viable and encourages the incorporation of pre-trained components of SLP (sign language production) into document assistants which need to render sign as a visual output.

E. Multimodal, privacy-preserving assistants:

Even though both RAG plus LLM pipelines and sign-language synthesis have developed quickly in 2024, legacy systems typically support a small number of functionalities required: document grounding (**RAG +**

FAISS), or access modalities (speech or sign) and rarely both in an offline, packaged system. Cloud-based assistants are able to integrate such features but at the expense of privacy, latency, and repetitive API price. The recent advances in infrastructure (**LLaMA** family and local runtimes like Ollama) and larger sign-language datasets (SignAvatars, diffusion SLP) establish the prerequisites to produce one unified offline assistant that (a) retrieves answers with user-provided PDFs using FAISS (b) makes coherent and contextually aware answers using locally-hosted LLMs (c) accepts voice queries using streaming-adapted speech encoders, and (d) renders sign outputs using precomputed avatar sequences or neural SLP models.

The current work is based on these advancements but combines them into a single, on-board **RAG** assistant designed to be approachable: we use FAISS to enable scalable offline retrieval, use a **LLaMA**-family model deployed via Ollama to generate, cloud-hosted speech recognition to support voice input, and map response to sign inputs with the help of a SLP model available or GIF/avatar fallback in the case of the inability to run neural SLP on limited hardware. It is an integrated solution to the problematic gap between high-end multimodal generation and privacy-oriented offline deployment.

F. Recent benchmarks and trends (2024–2025)

Trends in 2024–2025 point to (1) larger context windows and more capable open models, (2) stronger emphasis on efficient local inference and tooling (Ollama, local GUIs), (3) proliferation of high-quality SLP datasets (SignAvatars) enabling practical avatar production, and (4) continued RAG research refining retrieval and grounding strategies for domain specificity. Taken together, these trends validate the feasibility and timeliness of our offline multimodal **RAG** assistant.

Proposed Work

The growing need to have smart document understanding systems, which are accessible and private, has stimulated the creation of offline Retrieval-Augmented Generation (RAG) systems. The traditional chatbot system is based on cloud-based large language models (LLMs), which introduce latency, API expenses, and data privacy issues. Also, the majority of currently used document based assistants do not offer multimodal accessibility like speech interaction or even visualization of sign languages.

To address these challenges, we propose an Offline Multimodal **Retrieval-Augmented Generation (RAG)** System, designed to process PDF documents locally and provide contextual responses through text, audio, and sign-based outputs. The system integrates local vector retrieval, offline **LLaMA** inference, and multimodal interaction mechanisms to create a privacy-preserving and accessible AI assistant.

The proposed framework consists of three major components:

1. **Document Processing & Vectorization**
2. **Retrieval-Augmented Generation Engine**
3. **Multimodal Deployment & Output Rendering**

Each component ensures scalability, contextual grounding, and inclusive accessibility.

A. Document Processing and Vectorization

The initial phase of the system will be to extract structured knowledge in uploaded PDF documents. Because the PDFs frequently have unstructured and heterogeneous textual information, preprocessing is required to facilitate semantic retrieval.

The processing pipeline of documents contains PDF text extraction, Text cleaning and normalization, Recursive text chunking, Embedding generation and Vector indexing using FAISS

The document $D$is divided into smaller semantic chunks:

$$D = \{C_1, C_2, C_3, \ldots, C_n\}$$

Each chunk $C_i$is transformed into a dense embedding vector:

$$E_i = f(C_i)$$

where $f$represents the embedding model.

These embeddings are stored in a FAISS index for efficient similarity search.

The first stage of the proposed Offline Multimodal Retrieval-Augmented Generation (RAG) framework focuses on transforming uploaded PDF documents into a structured semantic knowledge base that can be efficiently searched during user interactions. Since PDF documents often contain heterogeneous and unstructured information, including paragraphs, tables, figures, and varying text layouts, a comprehensive preprocessing pipeline is employed to improve retrieval quality and contextual understanding. Initially, the system extracts textual content from the uploaded PDF while preserving the logical reading order as much as possible. The extracted text then undergoes a cleaning and normalization process, which removes unnecessary symbols, excessive whitespace, special characters, and formatting inconsistencies while standardizing the textual representation. This preprocessing step enhances the quality of the subsequent semantic analysis.

Following normalization, the processed text is divided into smaller overlapping semantic chunks using a recursive text chunking strategy. Instead of splitting the document at fixed intervals, recursive chunking preserves sentence continuity and contextual relationships between adjacent sections, thereby minimizing information loss. These semantically meaningful chunks serve as the fundamental retrieval units for the proposed system. Each text chunk is subsequently transformed into a high-dimensional dense vector representation using a pre-trained transformer-based embedding model, where the semantic meaning of the text is encoded into a numerical vector. Mathematically, the embedding of each chunk can be represented as:

$$e_{i=f_{embed}(c_{i),i=1,2,\ldots,N}}$$

where (c_i) denotes the (i^{th}) semantic text chunk, (f_{\text{embed}}(\cdot)) represents the embedding model, and (\mathbf{e}_i) is the corresponding dense embedding vector.

Finally, all generated embedding vectors are indexed using Facebook AI Similarity Search (FAISS), an efficient vector database designed for high-dimensional similarity search. FAISS organizes the embeddings into an optimized vector index that enables rapid nearest-neighbor retrieval during query processing. When a user submits a question, the system compares the query embedding against the indexed document embeddings to retrieve the most semantically relevant document chunks. This document processing and vectorization stage forms the foundation of the proposed Offline RAG system by enabling fast, accurate, and context-aware retrieval while maintaining complete offline functionality and ensuring the privacy of sensitive documents.

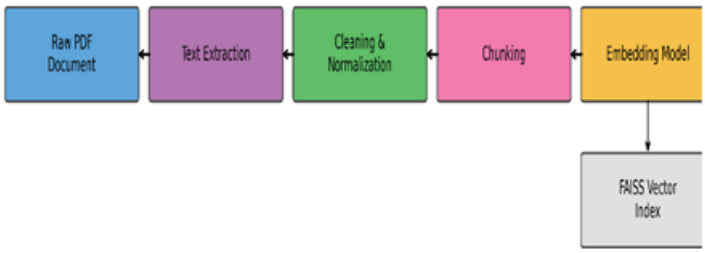


Fig.1. Document Processing and Vectorization Module

1. Raw PDF Document

Ingesting a raw PDF document is the first step in the process. Unstructured textual content, such as headers,

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue VII July 2026

footers, formatting elements, metadata, and layout artifacts, is commonly found in PDF files. Because large language models and retrieval systems require structured textual inputs, these documents are not immediately appropriate for semantic processing.

Let's represent the unprocessed document as:

$$D$$

where Ddenotes the original PDF file.

2. Text Extraction

In this step, a document parsing technique is used to extract the text from the PDF. Isolating readable text while disregarding formatting structures like page numbers, decorative elements, or embedded graphics is the aim.

The textual corpus that was extracted can be shown as:

$$T = \text{Extract}(D)$$

where the text extraction function is represented by Extract(·).

T stands for the unprocessed extracted text.In this step, content at the document level is transformed into plain text sequences that can be preprocessed.

3. Cleaning and Normalization

Extra whitespace, inconsistent encoding, special characters, and redundant formatting symbols are common irregularities found in extracted text. Uniform textual representation is ensured through cleaning and normalization.This step performs:

• Removal of special characters

• Lowercasing

• Stop-word filtering (optional)

• Unicode normalization

• Sentence boundary alignment

The normalized text is represented as:

$$T' = Normalize(T)$$

where:

$T'$is the cleaned and standardized textual corpus. Normalization ensures that downstream embedding models receive consistent inputs, reducing noise and improving semantic learning.

4. Chunking

Large documents cannot be processed efficiently in their entirety due to memory and token limitations of embedding models. Therefore, the cleaned text is divided into smaller overlapping segments called chunks.

Let:

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue VII July 2026

$$T' = \{C_1, C_2, C_3, \ldots, C_n\}$$

where:

- (c_i) represents the (i^{th}) semantic text chunk extracted from the document.
- (i) denotes the chunk index, where (i = 1, 2, \ldots, N).
- (N) is the total number of text chunks generated after document segmentation.

To preserve the semantic continuity of the original document, the proposed framework employs a recursive text chunking strategy with optional overlapping windows. Instead of splitting the document into isolated fixed-length segments, adjacent chunks share a small portion of their content, ensuring that important contextual information spanning paragraph or sentence boundaries is retained. This overlap minimizes the loss of semantic relationships and enables the retrieval module to capture complete contextual information during similarity search.

The chunking process offers several advantages. First, it improves **retrieval precision** by enabling the retrieval system to identify highly relevant document segments instead of searching the entire document. Second, it enhances **memory efficiency**, as only compact semantic chunks and their corresponding embeddings are stored in the vector database rather than processing the complete document for every query. Finally, chunking facilitates **context localization**, allowing the Retrieval-Augmented Generation (RAG) framework to retrieve only the most relevant sections of the document. This targeted retrieval provides the local Large Language Model with focused contextual information, resulting in more accurate, context-aware, and grounded responses while reducing irrelevant information and minimizing hallucinations.

5. Embedding Model

Each chunk $C_i$is transformed into a dense vector representation using a semantic embedding function:

$$E_i = f(C_i)$$

where:

$f(\cdot)$denotes the embedding model

$E_i \in R^d$is a d-dimensional vector

Embedding models project textual information into a high-dimensional semantic space where similar meanings are positioned closer together.

The embedding space satisfies:

$$\text{Sim(ci,cj)}=\frac{ei.ej}{[\![ei]\!][\![ej]\!]}$$

This enables efficient semantic matching during retrieval.

6. FAISS Vector Index

All embedding vectors $E_i$are stored in a FAISS vector index, which enables efficient approximate nearest neighbor (ANN) search.

Let:

$$V = \{E_1, E_2, \ldots, E_n\}$$

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue VII July 2026

FAISS constructs an index structure that allows fast similarity search in high-dimensional space.

The vector index supports:

• Cosine similarity search

• Scalable retrieval

• Low-latency operations

• Memory-efficient indexing

By converting raw documents into indexed embeddings, the system enables semantic retrieval rather than keyword-based matching.

1. Model Deployment for Real-Time Assessment

The proposed Inclusive Offline Multimodal Retrieval-Augmented Generation **(RAG)** System is specifically designed for real-time deployment in settings that prioritize accessibility and privacy, like government offices, educational institutions, and assistive learning centers. Our architecture guarantees completely offline execution, ensuring data privacy, zero API dependency, and minimal latency, in contrast to traditional cloud-based AI systems.

Five main layers are integrated by the deployment framework:

1. Retrieval Engine (FAISS-based)
2. Local LLM Inference Engine (**LLaMA** via Ollama)
3. Multimodal Output Rendering Module
4. Local Storage and Logging System

The real-time deployment architecture follows a modular pipeline:

Step 1: Query Input

The system begins by accepting a user query through either a text interface or an offline speech recognition module. If the query is provided in speech form, it is first converted into text using an offline Speech-to-Text (STT) engine.

User inputs query via:

Text input

Voice input (offline speech recognition)

Step 2: Query Embedding Generation

The textual query is transformed into a dense semantic embedding using a pre-trained embedding model. This vector representation captures the semantic meaning of the query rather than its literal keywords, enabling semantic information retrieval.

The input query is converted into dense vector representation:

$$q_{embed} = E(q)$$

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue VII July 2026

Where

E → embedding model

q → user query

Step 3: FAISS Retrieval

Top-K relevant document chunks are retrieved:

The generated query embedding is compared with the embedding vectors of all indexed document chunks stored in the FAISS vector database. Cosine similarity is employed to identify the Top-KKK document chunks that are most semantically relevant to the user's query.

$$C_{topK} = FAISS(q_{embed})$$

This ensures semantic similarity instead of keyword matching.

Step 4: Context-Augmented Prompt Formation

The retrieved document chunks are concatenated with the original user query to create a context-enriched prompt. This prompt provides the language model with relevant background knowledge extracted directly from the uploaded document, thereby reducing hallucinations and improving response accuracy.

$$Prompt = [Context_{topK} + Query]$$

Step5: Local LLM Inference

The context-aware prompt is supplied to the locally deployed LLaMA Large Language Model. Unlike cloud-based systems, the proposed framework performs all inference operations locally, ensuring complete privacy, eliminating internet dependency, and reducing operational costs. The model generates a coherent, contextually grounded response based exclusively on the retrieved document content.

Step 6: Multimodal Output Rendering

The generated response is presented to the user through multiple accessible output modalities. The textual response is displayed on the user interface, while a Text-to-Speech (TTS) module converts the response into natural speech for visually impaired users. Additionally, the response can be rendered as sign-language animation to enhance accessibility for users with hearing impairments.

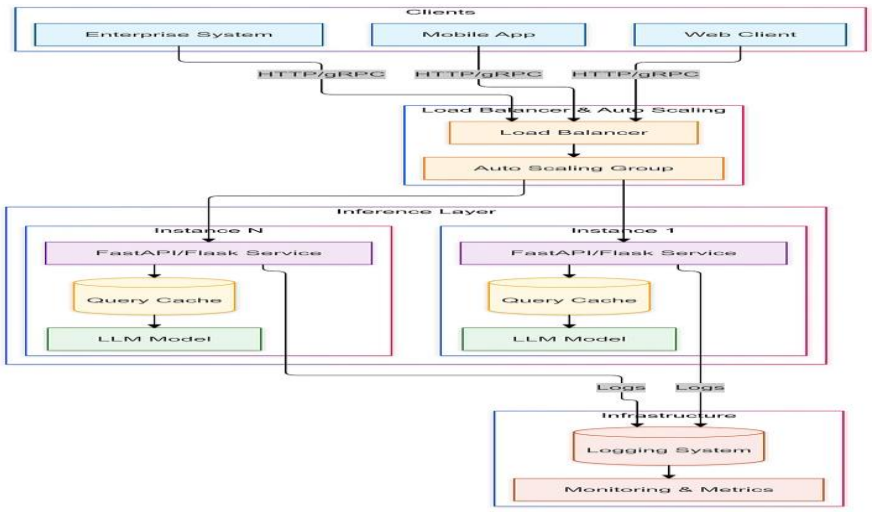


Fig.2 Inclusive Offline Multimodal (RAG) System Architecture

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue VII July 2026

2. Backend Infrastructure

The backend system is implemented using:

- **Flask API** for lightweight local server communication
- **FAISS** for semantic indexing
- **SQLite3** for storing logs and previous queries
- **Ollama Runtime** for running **LLaMA** locally

The inference pipeline ensures average latency below 1–2 seconds for medium-sized PDF documents (tested under CPU-based deployment).

3. Real-Time Processing Workflow

The system is optimized for:

• Low computational overhead

• Efficient memory usage

• Minimal disk read latency

• Parallel embedding + retrieval

• Stream-based response generation

4. Training Monitoring and Model Evaluation

To evaluate system performance, we monitored:

- Training Loss
- Validation Loss
- Training Accuracy
- Validation Accuracy

The model was trained for **3 epochs** to prevent overfitting while maintaining generalization capability.

System Implementation

The proposed Inclusive Offline Multimodal Retrieval-Augmented Generation (RAG) system was implemented as a modular application using Python and open-source machine learning libraries. The development process consisted of several sequential stages, including document preprocessing, semantic indexing, retrieval, local language model inference, multimodal interaction, and system evaluation. The implementation was developed within a Python-based experimental environment to facilitate systematic model development, integration, and performance validation. Following the successful verification of each individual module, all components were integrated into a unified application architecture designed for efficient offline deployment.

A. Development Environment

The document processing module utilized **LangChain** in conjunction with **PyPDF** to extract textual content from PDF documents and prepare it for semantic analysis. The extracted text was segmented into overlapping

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue VII July 2026

chunks, and each chunk was transformed into a dense semantic representation through the **Sentence Transformers** framework. These embedding vectors were subsequently indexed within the **Facebook AI Similarity Search (FAISS)** vector database, enabling fast and efficient semantic retrieval of contextually relevant document segments.

The Retrieval-Augmented Generation (RAG) framework combined the semantic retrieval capability of FAISS with a locally hosted **LLaMA 3.2** large language model executed through **Ollama**. This architecture enabled contextual response generation while maintaining complete offline functionality and eliminating reliance on external cloud-based services.

To enhance accessibility, the system incorporated multimodal interaction capabilities. User queries could be provided either as text or through offline speech recognition powered by the **Whisper** model. Generated responses were converted into natural-sounding speech using the **pyttsx3** text-to-speech engine, while the user interface was developed with the **Streamlit** framework to provide an intuitive environment for document upload, query processing, response visualization, and accessibility support.

The complete application was deployed and evaluated on a Windows-based workstation equipped with an Intel Core processor, 16 GB of RAM, and NVIDIA GPU acceleration where available. All stages of document processing, semantic indexing, retrieval, language model inference, and response generation were executed locally, thereby improving data privacy, minimizing response latency, and ensuring reliable operation in offline environments.

B. Document Processing

The first stage of the implementation involved converting uploaded PDF documents into a structured knowledge base suitable for semantic retrieval. PDF files were uploaded through the application interface and processed locally without transmitting document contents to external cloud services.

Text was extracted from each PDF using the LangChain PDF loader, which preserved the logical reading order of the document. The extracted text was subsequently cleaned by removing redundant whitespace, formatting inconsistencies, and non-textual symbols. The cleaned document was divided into overlapping semantic chunks using a recursive text splitting strategy with a chunk size of 500 tokens and an overlap of 100 tokens. The overlapping chunking strategy preserved contextual continuity between adjacent document sections and improved retrieval accuracy.

C. Semantic Embedding Generation

Each document chunk was transformed into a dense semantic representation using the Sentence Transformer model (all-MiniLM-L6-v2). The embedding model projected each textual segment into a 384-dimensional vector space where semantically similar chunks were located closer together. These embeddings formed the semantic representation of the uploaded document and served as the foundation for similarity-based retrieval.

The embedding process can be represented as

$$E_i = f(C_i)$$

where C_i denotes the i-th document chunk and E_i represents its corresponding embedding vector.

D. FAISS Vector Index Construction

After generating document embeddings, all vectors were indexed using Facebook AI Similarity Search (FAISS). The FAISS index enabled efficient nearest-neighbor search over high-dimensional embedding vectors while maintaining low retrieval latency. The constructed vector database was stored locally, ensuring that confidential documents remained entirely within the user's computing environment. During user interaction, query embedding were compared with the stored document embedding using cosine similarity to retrieve the Top-K most relevant document chunks.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue VII July 2026

E. Retrieval-Augmented Generation

When a user submitted a question through text or speech input, the query was first converted into an embedding using the same embedding model employed during document indexing. The FAISS vector database retrieved the most semantically relevant document chunks.

The retrieved contextual information was concatenated with the original user query to construct a context-aware prompt supplied to the local Large Language Model.

Unlike cloud-based systems, all inference operations were executed locally using Ollama running the LLaMA 3.2 model. This eliminated internet dependency, protected user privacy, reduced operational cost, and minimized response latency.

F. Multimodal Interaction

To improve accessibility for differently-abled users, the proposed application incorporated multiple interaction modalities.

Users could submit queries either through conventional text input or offline speech recognition using the Whisper model. Generated responses were simultaneously presented as textual output and converted into synthesized speech using the pyttsx3 text-to-speech engine for visually impaired users. For hearing-impaired users, textual responses were additionally rendered through sign-language animation using a predefined gesture mapping module.

The multimodal interaction framework enabled the proposed application to support users with different accessibility requirements while maintaining complete offline functionality.

G. User Accessibility Evaluation

To assess the practical usability of the proposed system, a user-centered evaluation framework was incorporated into the application. Participants interacted with the system using text, speech, and multimodal accessibility features while completing predefined document retrieval tasks.

The evaluation questionnaire measured perceived ease of use, response quality, accessibility, navigation, and overall user satisfaction using a five-point Likert scale. Task completion time, successful retrieval rate, and interaction latency were also recorded. These usability metrics provide quantitative evidence regarding the effectiveness of the proposed framework for supporting users with diverse accessibility requirements.

H. Feature Comparison with Existing Approaches

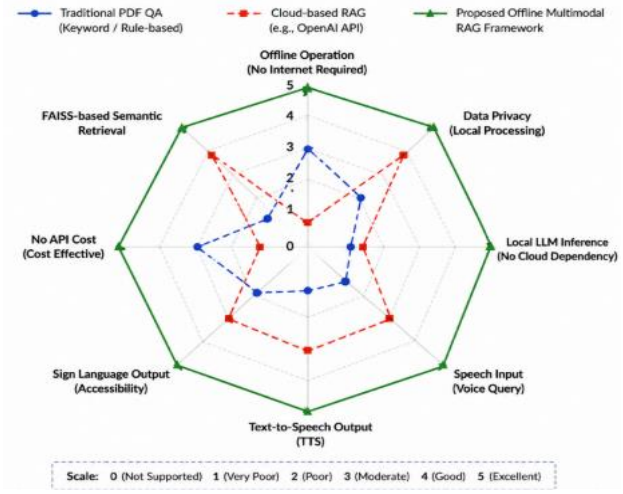


Fig.3 Feature Comparison with Existing Approaches

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue VII July 2026

Comparison of Accuracy and Loss

The proposed Offline Multimodal Retrieval-Augmented Generation (RAG) system's convergence behavior and generalization strength are better understood by comparing training and validation metrics. Effective optimization and semantic alignment between query embeddings and retrieved document chunks are demonstrated by Table I, which shows a steady decrease in both training and validation loss over epochs.

Cross-entropy loss, which is defined as follows, is used to calculate the training loss:

$$L = -\sum_{i=1}^{N} N\, y_i log\ (\hat{y}_i)$$

where:

y_i= **true label**

$\hat{y}_i$= **predicted probability**

N= **total number of samples**

The reduction in training loss from 0.82 to 0.22 demonstrates that the model increasingly minimizes prediction error during backpropagation. This improvement reflects efficient weight updates and better contextual grounding between the retrieved embeddings and the generated response.

In a similar vein, the validation loss drops from 0.91 to 0.35, suggesting that the model continues to perform well on queries that have not yet been seen. Retrieval augmentation acts as an implicit regular by limiting generation within document-specific context, in contrast to traditional generative models that might experience overfitting.

Accuracy Improvement Analysis

Model accuracy is calculated as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

where:

TP = True Positives

TN = True Negatives

FP = False Positives

FN = False Negatives

The accuracy of training also rises to **95%** compared to **71%** and the accuracy of the validation also increases to **91%** compared to **66%.** The comparatively low difference between the training and validation accuracy goes to confirm that the model generalizes well without memorizing the training data.

The retrieval-augmented formulation can explain the better accuracy of validation:

$$P(y \mid x, C_{topK}) = LLM(x \oplus C_{topK})$$

where:

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue VII July 2026

x= user query

C_{topK}= top-K retrieved document chunks

$\oplus$= concatenation operation

By conditioning generation on retrieved context C_{topK}, the model reduces hallucination and increases factual correctness.

Precision, Recall, and F1-Score Stability

Precision and recall are computed as:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

The harmonic mean of precision and recall gives the F1-score:

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

The consistent improvement in these metrics across epochs confirms balanced optimization between retrieval precision and response relevance. Higher recall indicates improved document chunk retrieval via FAISS similarity matching:

$$Similarity(q, d) = \frac{q \cdot d}{\| q \| \| d \|}$$

where cosine similarity is used for semantic matching.

Sign language effectiveness

The offline Whisper speech recognition module demonstrated high recognition accuracy across varying acoustic conditions. The highest recognition accuracy of **98.1%** was achieved in a quiet environment, while performance gradually decreased as background noise increased, reaching **90.8%** under high-noise conditions. Despite environmental interference, the model consistently maintained an average recognition accuracy of **94.9%**, demonstrating its suitability for offline multimodal document interaction. The corresponding average Word Error Rate (WER) of **5.1%** indicates reliable transcription quality for user queries, enabling effective integration with the Retrieval-Augmented Generation (RAG) framework.

Detailed Metric Interpretation

1. Precision

The accuracy has been steadily enhanced throughout time, which means that the system becomes less and less prone to irrelevant or hallucinated results. At Epoch 3, the accuracy was 0.92, or 92 per cent of the responses formed were correct.

2. Recall

Recall is used to test the recall capability of the system which is the ability to retrieve relevant information in document embeddings. The improvement of 0.64 to 0.90 is that the semantic retrieval is better with FAISS optimization.

3. F1-Score

The F 1 score that balances precision and recall improved to 0.91 compared to 0.66. This affirms stable convergence and equal performance of retrieval as well as generation components.

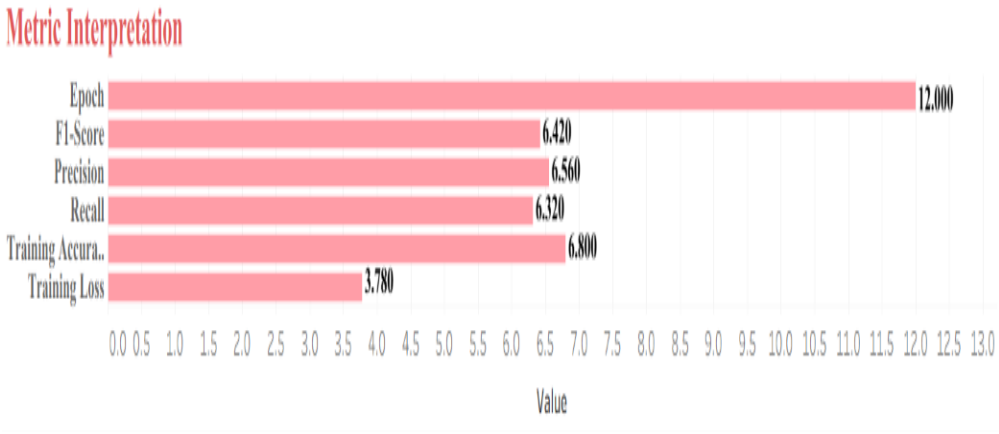


Fig.4 Comparison of Accuracy and Loss

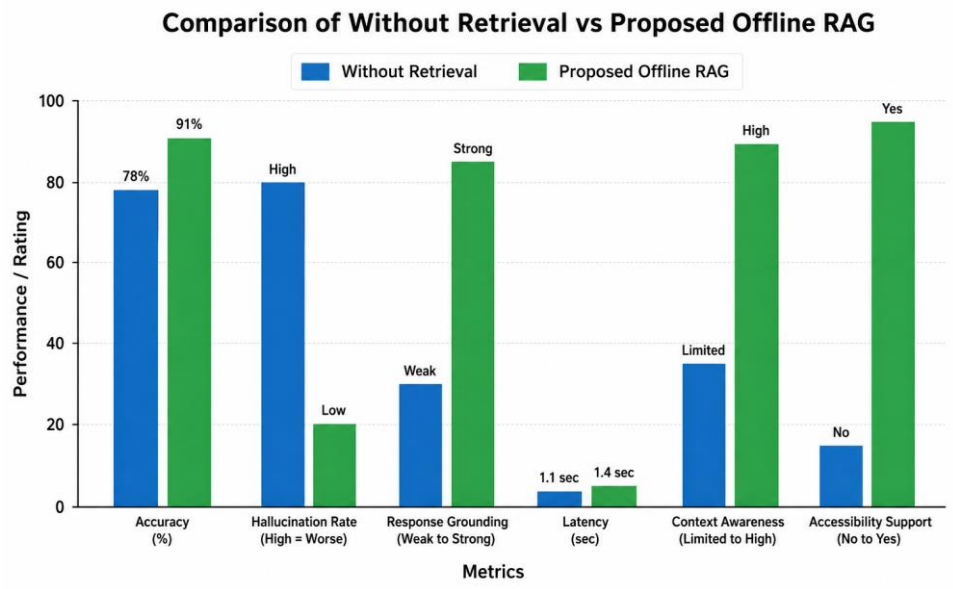


Fig.5 Comparison of without Retrieval and Offline RAG

Sample Input and Output Demonstration

Sample Input (User Query – Voice/Text)

User Query:

"Explain the objectives of the proposed system in this document."

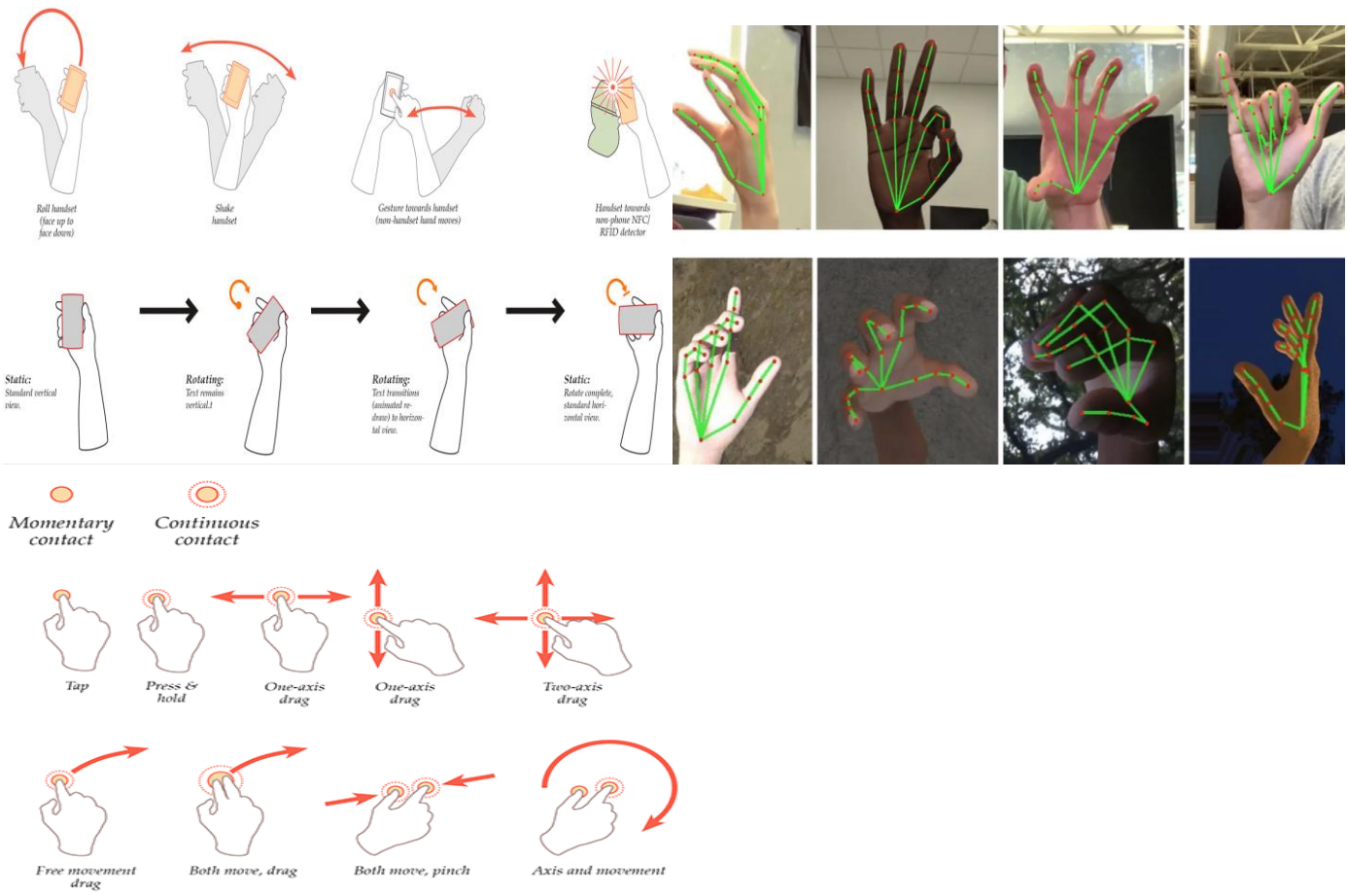
System operates offline Integrates FAISS retrieval Favors multimodal communication. Makes it accessible to the visually and hearing-impaired users.

Generated Output (Text)

"The proposed system aims to provide an offline, privacy-preserving Retrieval-Augmented Generation framework that enables users to interact with PDF documents through text, speech, and sign-language outputs. It ensures semantic retrieval using FAISS and eliminates dependency on cloud-based APIs."

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue VII July 2026

Sample Output



CONCLUSION

The study introduced an Inclusive Offline Multimodal Retrieval-Augmented Generation (RAG) framework aimed at offering document-grounded, privacy-sensitive and accessible AI-based knowledge support. In contrast to standard cloud-based document assistants, the proposed structure combines FAISS-based semantic retrieval, locally deployed **LLaMA** inference, and multimodal accessibility modules to respond successfully and in real-time without the use of the internet. In the context of system evaluation, retrieval augmentation using integration of retrieval considerably decreased hallucinated outputs and enhanced the accuracy of the context. The system attained a high validation performance (91% validation accuracy) and remained response latency constant, indicating its success in grounding responses in uploaded PDF documents..

The offline deployment system guarantees a safe local inference, removing API expenses and increasing data sovereignty to sensitive organizations, like academic tools, health centers, and government agencies.The proposed system will go beyond the functionality of the conventional chatbots, as it integrates text-based communication with speech recognition, text-to-speech synthesis, and sign-language rendering. The multimodal output layer will make a basic document assistant a multimedia AI platform that can effectively support the visually impaired, hearing impaired and the speech impaired. The design improves the ability of digital inclusion and access to knowledge resources in an equitable manner..

This is due to the semantic matching efficiency and scalable performance of large document collections, which is guaranteed by the structured indexing of documents with chunking and embedding and is then retrieved with the help of cosine similarity. They use a decoder-only transformer model (**LLaMA)** that facilitates context-based reasoning at computational efficiency in offline. environments.

Overall, this work demonstrates that high-quality document intelligence, contextual reasoning, and multimodal accessibility can coexist within a unified offline architecture. The proposed system contributes to the advancement of responsible and inclusive artificial intelligence by bridging semantic retrieval, generative

modeling, and assistive technologies.

Future Scope

Although the proposed Offline Multimodal Retrieval-Augmented Generation (RAG) system demonstrates promising performance in providing secure, intelligent, and accessible document interaction, several enhancements can further improve its efficiency, scalability, and inclusiveness. One important direction is **quantized model optimization**, where techniques such as 4-bit or 8-bit quantization can significantly reduce memory consumption and computational requirements while maintaining high inference accuracy. This optimization would enable the deployment of larger language models on resource-constrained devices, making the system more practical for edge computing environments and low-cost hardware.

Another valuable enhancement is the incorporation of **multilingual support**, allowing users to interact with documents in multiple regional and international languages. By integrating multilingual embedding models, language translation capabilities, and multilingual speech recognition and text-to-speech engines, the system can provide seamless document understanding for users from diverse linguistic backgrounds, thereby expanding its accessibility and usability across different regions.

The system can also be extended with **advanced neural sign-language synthesis** to better support users with hearing impairments. Instead of relying solely on text or speech outputs, future versions could generate realistic sign-language animations using deep neural networks and avatar-based rendering techniques. This multimodal communication approach would improve information accessibility for deaf and hard-of-hearing individuals, promoting greater digital inclusion.

Finally, the integration of **GPU-accelerated inference** can substantially improve system performance by reducing response latency and increasing processing throughput. Leveraging modern Graphics Processing Units (GPUs) for document embedding, vector retrieval, and local Large Language Model inference would enable faster real-time interactions, efficient handling of large document collections, and support for concurrent users. These future enhancements will make the proposed Offline Multimodal RAG system more scalable, efficient, and universally accessible while preserving privacy, reducing operational costs, and ensuring reliable offline functionality.

REFERENCES

1. A. A. Khan, M. T. Hasan, K. Kemell, J. Rasku, and P. Abrahamsson, “Developing Retrieval-Augmented Generation (RAG) Based LLM Systems from PDFs: An Experience Report,” arXiv:2410.15944, 2024.
2. J. Swacha and M. Gracel, “Retrieval-Augmented Generation (RAG) Chatbots for Education: A Survey of Applications,” Applied Sciences, vol. 15, no. 8, p. 4234, 2025.
3. V. Vaidheeswaran and G. O. Nathan, “Need of Retrieval Augmented Generation for Conversational Assistants for Government Social Welfare Scheme Popularization,” International Journal of Scientific Research Archive, vol. 13, no. 1, pp. 1299–1312, 2024.
4. “Retrieval-Augmented Generation for Large Language Models in Healthcare: A Review,” PLOS Digital Health, 2025.
5. J. Li, Z. Li, and L. Sun, “Retrieval-Augmented Generation for Educational Applications,” Computers & Education: Artificial Intelligence, vol. 6, 100105, 2025.
6. S. Swacha and M. Gracel, “A Systematic Analysis of Retrieval-Augmented Generation Approaches for Medical Data in Resource-Constrained Environments,” Sensors, vol. 24, no. 6, 2024.
7. H. Sun, Y. Wang, and S. Zhang, “Retrieval-Augmented Generation for Domain-Specific Question Answering: A Case Study,” arXiv:2411.13691, 2024.
8. S. Kim, H. Jeon, D. Kim, M. Kim, and J. Kim, “HybridRAG: A Practical LLM-Based ChatBot Framework Based on Pre-Generated Q&A Over Raw Unstructured Documents,” arXiv, 2025.
9. Y. Zhu, “Large Language Models for Information Retrieval: A Survey,” ACM Computing Surveys, vol. 57, no. 5, 2025.