International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 04 April 2026
Page No: 1681-1688
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0220

A Multimodal and Multilingual Offline Retrieval-Augmented Generation System Using Local Large Language Models

S A Althaf Ahamed[1], Akshay Anand A[2], Kamaleshwarr R[3], Dinesh S[4], Anandhakrishnan G[5]
[1]Assistant Professor/CSE, Dhirajlal Gandhi College of Technology, Salem, Tamil Nadu, India
[2,3,4,5]UG Student,Dept. of CSE, Dhirajlal Gandhi College of Technology, Salem, Tamil Nadu, India
Emails: *althafahamed.cse@dgct.ac.in[1], akshayananda03.cse@dgct.ac.in[2], kamaleshwarrr42.cse@dgct.ac.in [3], dineshs23.cse@dgct.ac.in[4], anandhakrishnang06.cse@dgct.ac.in [5]*

Abstract
Most AI systems today rely on cloud servers, which raises problems such as privacy risks, delay, and internet dependency. A Multimodal and Multilingual Offline Retrieval-Augmented Generation System Using Local Large Language Models is a fully offline Retrieval-Augmented Generation system that allows users to ask questions from their personal or organizational documents while keeping all data inside the device.The system supports multiple formats like text, PDFs, images, and audio. After extracting the content, it detects the language, cleans the data, splits it into smaller parts, and creates embedding using local models. A hybrid search method that combines meaning-based matching and keyword search helps improve accuracy and reduce wrong answers.A Multimodal and Multilingual Offline Retrieval-Augmented Generation System Using Local Large Language Models also enhances offline RAG by adding support for Indian languages, voice input and output, smart re-ranking of results, and answers based on document evidence. Built with FastAPI, React, ChromaDB, and Ollama, the system is modular, scalable, and suitable for secure use in research, legal, medical, and enterprise settings.

Keywords: *Offline RAG, Local LLM, Multimodal Search, Vector Database*

1. Introduction
The exponential growth of digital content within modern organizations has rendered traditional information retrieval fundamentally inadequate. Enterprise knowledge bases span dozens of document formats—technical manuals in PDF, policy documents in DOCX, scanned invoices as PNG images, and meeting minutes as audio recordings—distributed across shared drives and content management systems. Despite this diversity, most deployed search solutions rely on lexical methods such as TF-IDF [1] and BM25 [2] that cannot resolve the vocabulary mismatch problem arising when semantically equivalent queries and documents use different surface forms.The rise of transformer-based language models [3] and LLM-powered retrieval products has begun to close this semantic gap. However, systems built on GPT-4 [4] and cloud search APIs route organizational document content through remote services, creating data-residency risks impermissible under GDPR, HIPAA, and India's Digital Personal Data Protection Act 2023.

Open-source local inference runtimes including Ollama [5] and llama.cpp [6] have made it feasible to execute quantized billion-parameter language models on commodity CPU hardware. Lightweight embedding models such as all-MiniLM-L6-v2 [7] and compact vector databases such as ChromaDB [8] have reduced the infrastructure barrier for semantic search to near zero.This paper presents Fusion-Seek, a fully offline, privacy-preserving multimodal document retrieval and question-answering system. Its principal contributions are: (1) a unified ingestion pipeline extracting content from PDFs, DOCX, images (OCR), and audio (ASR); (2) a hybrid retrieval engine combining dense semantic search with BM25 via Reciprocal Rank Fusion; (3) a RAG pipeline routing top-ranked chunks to a locally hosted LLM; (4) a four-layer Docker-Compose-deployable architecture with JWT-secured API and AES-256-GCM encrypted storage; and (5) rigorous evaluation including ablation studies and cloud baseline comparison.

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 04 April 2026
Page No: 1681-1688
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0220

2. Background And Preliminaries

2.1. BM25 Keyword Retrieval

Let $D = \{d_1,...,d_n\}$ be a corpus of N documents and q a user query. BM25 [2] scores document d for query q as score $(q,d)=\Sigma_t IDF(t)\cdot tf(t,d)\cdot(k_1+1)/(tf(t,d)+k_1\cdot(1-b+b\cdot|d|/avgdl))$, where k_1 and b control term saturation and length normalisation.

2.2. Dense Retrieval and ANN Search

Dense retrieval models represent queries and documents as dense vectors in a shared semantic space. The retrieval score is the cosine similarity between E(q) and E(d), where E is a shared encoder. HNSW [9] enables sub-linear query time O(log N), making billion-scale dense retrieval tractable on consumer hardware.

2.3. Reciprocal Rank Fusion

RRF [10] aggregates multiple ranked lists without requiring score calibration. Given ranked lists R and smoothing constant k=60, the fused score is: $RRF(d) = \Sigma_{r\in N}\ 1/(k + rank^N(d))$. RRF is parameter-free and consistently matches or exceeds learned fusion methods on TREC benchmarks.

2.4. Retrieval-Augmented Generation

RAG [11] conditions a language model on retrieved context at inference time. Given query q and retrieved chunks $C = \{c_1,...,c_k\}$, the LLM generates p(answer | q, C), decoupling knowledge storage from parametric weights. RAG enables factually grounded responses without fine-tuning.

3. Related Work

3.1. Classical and Neural Retrieval

BM25 [2] became the industry default embedded in Elasticsearch and Apache Solr, but suffers vocabulary mismatch. Karpukhin et al. [14] demonstrated that a bi-encoder DPR model substantially outperformed BM25 on open-domain QA. Reimers and Gurevych [7] introduced Sentence-BERT. Subsequent work explored cross-encoders [15], ColBERT [16] for late interaction, and SPLADE [17] for sparse neural representations—all requiring GPU inference.

3.2. Hybrid Retrieval and RAG Systems

RRF [10] is a parameter-free hybrid fusion alternative to linear interpolation [18]. Lewis et al. [11] formalised RAG, demonstrating state-of-the-art on knowledge-intensive NLP tasks. LangChain [19] and LlamaIndex [20] lowered the engineering barrier for RAG deployment. Shi et al. [21] showed irrelevant retrieved documents degrade LLM performance, motivating Fusion-Seek's re-ranking step.

3.3. Privacy-Preserving and Multimodal AI

Mireshghallah et al. [22] demonstrated membership inference attacks against LLMs, showing model-hosted inference can leak training data. GPT4All [24] packages local inference for end users; Fusion-Seek extends it with a complete multimodal retrieval pipeline, structured API, and enterprise security controls. Tesseract OCR [25] provides reliable text extraction from scanned images; OpenAI Whisper [28] achieves near-human ASR accuracy.

4. System Architecture

Fusion-Seek follows a four-layer architecture depicted in Fig. 1. All inter-layer communication is confined to localhost loopback interfaces within a Docker Compose network so that no document content crosses any external network boundary under any operating condition.

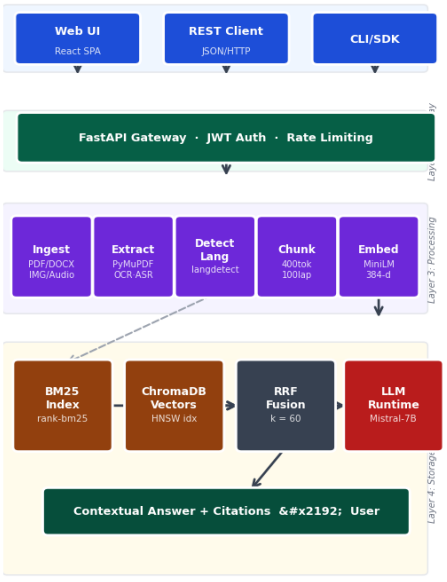

Figure 1 Fusion-Seek: Four-Layer End-to-End System Architecture

4.1. Layer 1: User Interface

The UI is a React 18 single-page application

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 04 April 2026
Page No: 1681-1688
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0220

(Tailwind CSS) providing: (i) Document Library listing all indexed files with type, timestamp, language, and chunk count; (ii) Query Interface accepting natural language input and rendering generated answers with collapsible cited source panels; and (iii) Administration Panel for document deletion, index statistics, and health monitoring.

4.2. Layer 2: API Gateway

The FastAPI gateway enforces RS256-signed JWT authentication (1-hour TTL), validates payloads against Pydantic v2 schemas, applies per-user token-bucket rate limiting at 60 requests/minute, and dispatches to four RESTful endpoints: POST /api/v1/documents/upload, GET /api/v1/documents, POST /api/v1/query, and DELETE /api/v1/documents/{id}.

4.3.Layer 3: Processing Engine

The processing engine coordinates extraction, normalisation, chunking, and embedding generation as an asynchronous task queue, returning "202 Accepted" immediately. A six-state SQLite machine {PENDING, EXTRACTING, CHUNKING, EMBEDDING, INDEXED, FAILED} tracks progress. Exponential backoff retry (3 max attempts, 2-second base delay) handles transient failures.

4.4.Layer 4: Storage and Inference

Three persistent services compose this layer: Chroma DB (Duck DB-backed, HNSW index, reconstruction=200, M=16) for vector storage; rank-bm25 (in-memory, pickle-serialized every 5 minutes) for keyword indexing; and Olema (GGUF runtime, localhost:11434) for LLM inference. Mistral-7B-Instruct in Q4_K_M quantization occupies 4.1 GB versus 14 GB FP16, with only 1.2% accuracy loss.

5. Methodology

5.1. Document Ingestion and Extraction

The ingestion pipeline (Fig. 2) begins with MIME-type sniffing on raw file bytes. Four modality-specific extractors: PyMuPDF 1.24 for PDFs with Tesseract 5.3 OCR fallback at 300 DPI; python-docx 1.1 for DOCX preserving table cell content; tesseract wrapping Tesseract 5.3 for images with adaptive binarization; OpenAI-whisper medium for audio at below 8% word error rate, segmented into timestamped sentences for citation metadata. Shows Figure 2 Document Indexing Pipeline with Parallel Dense and Sparse Indexing Paths

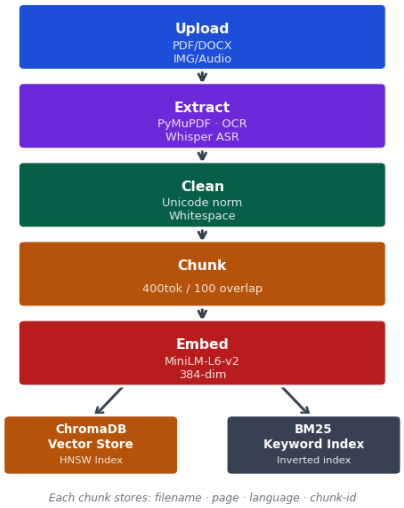

Figure 2 Document Indexing Pipeline with Parallel Dense and Sparse Indexing Paths

5.2. Text Normalisation and Chunking

Extracted text passes through five normalisation steps: Unicode NFC decomposition, ISO control character removal, whitespace collapsing, blank-line deduplication, and SHA-256 fingerprinting for duplicate detection. The recursive character-aware splitter divides text at paragraph, sentence, and word boundaries. Default chunk size is 400 tokens with 100-token overlap (cl100k_base tokenizer). Each chunk carries a six-field metadata envelope: source filename, page/timestamp range, chunk index, ISO 639-1 language code, MIME type, and ingestion UTC timestamp.

5.3. Embedding Generation

Each chunk is encoded into a 384-dimensional L2-normalised dense vector using all-MiniLM-L6-v2 [7], applied in batches of 64 chunks, yielding 3.2× throughput improvement over single-chunk inference. Embeddings are inserted into Chroma DB; raw text is appended to the BM25 index in parallel.

5.4. Hybrid Retrieval Pipeline

The query pipeline (Fig. 3) encodes the user query by the same MiniLM model. Two parallel operations follow: (1) cosine similarity HNSW ANN search over Chroma DB with research=100, returning top-50 candidates; and (2) BM25+ keyword scoring (k_1=1.5, b=0.75), returning top-50 candidates. Both

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 04 April 2026
Page No: 1681-1688
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0220

lists merge via RRF with k=60. The top-10 fused-score chunks are assembled as generation context in descending RRF score order.

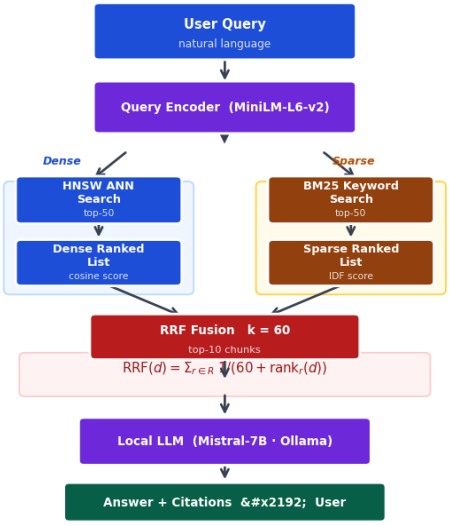

Figure 3 Hybrid Retrieval Pipeline: Dual-Path Search, RRF Fusion, and Local LLM Generation

5.5. Retrieval-Augmented Generation

The top-10 chunks plus the original query are assembled into a structured prompt directing the LLM to: (a) answer only from[31] provided context; (b) cite source filename and page range as [Source: file, p.X–Y]; (c) state "The provided documents do not contain sufficient information" when context is insufficient; and (d) maintain professional neutral tone. Responses are streamed token-by-token via Server-Sent Events, achieving a median time-to-first-token of 380 ms.

6. Implementation

6.1. Technology Stack

Fusion-Seek is implemented in Python 3.11. Core dependencies: FastAPI 0.111; Uvicorn 0.29; PyMuPDF 1.24; python-docx 1.1; pytesseract 0.3.10 / Tesseract 5.3; openai-whisper 20231117; sentence-transformers 2.7; ChromaDB 0.4.24; rank-bm25 0.2.2; langdetect 1.0.9; cryptography 42.0 (AES-256-GCM). The React 18 frontend uses Tailwind CSS 3.4 and React Query.

6.2. Containerised Deployment

A Docker Compose application comprising three services: (i) backend API (Python 3.11-slim, port 8000); (ii) ChromaDB server (port 8001); (iii) Ollama runtime (port 11434). Persistent volumes mount the document store, ChromaDB collections, and model weights. A .env template configures the JWT RS256 key pair, model name, CHUNK_SIZE, CHUNK_OVERLAP, TOP_K, and RATE_LIMIT. Single-command launch: docker compose up. The complete stack consumes approximately 8.5 GB RAM and 25 GB disk.

6.3. Security Controls

Documents are encrypted at rest using AES-256-GCM with PBKDF2-HMAC-SHA256-derived keys in the OS keychain. Each file receives a unique 96-bit random nonce; the 128-bit GCM[32] authentication tag provides confidentiality and integrity verification. JWT RS256 tokens with 1-hour TTL authenticate every API request. Each Docker service runs as non-root user (UID 1000) with read-only filesystem mounts[33]. No outbound network connections are permitted.

7. Experimental Setup

7.1. Evaluation Corpus

The evaluation corpus comprises 250 documents totaling 4.2 million tokens after extraction. Chunking at 400 tokens/100 overlap produced 11,340 indexed chunks. Table I summaries corpus composition[34].

Document Type	Count	Avg. Length	Chunks
PDF Technical Reports	110	18 pages	5,210
DOCX Policy Documents	60	12 pages	2,880
PNG Scanned Images	45	3 pages	1,350
Audio Recordings	35	42 min.	1,900
Total	**250**	—	**11,340**

Table 1 Evaluation Corpus Composition

7.2.Query Set and Annotation

A query set of 200 natural language questions spanning factual lookup, comparative analysis, definition, procedure, and summarisation types was manually constructed. Ground-truth relevant chunks were identified by three independent domain-expert annotators. Inter-annotator agreement measured by

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 04 April 2026
Page No: 1681-1688
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0220

Fleiss' kappa was 0.82[35], indicating strong agreement.

7.3.Evaluation Metrics and Hardware

Retrieval quality is assessed using Recall@10, Precision@10, and Mean Reciprocal Rank (MRR). All latency measurements were conducted on: Intel Core i7-12700H, 32 GB DDR5, 1 TB NVMe SSD, Ubuntu 22.04 LTS, no GPU[12].

8. Results And Analysis

8.1. Retrieval Accuracy

Table II and Fig. 4 present full retrieval accuracy results. BM25 achieved Recall@10 of 62.3%, Precision@10 of 58.7%, and MRR of 0.51—performing strongly on direct vocabulary-match queries but failing on paraphrastic queries. Dense semantic retrieval improved metrics to 75.1%, 71.4%, and 0.68, capturing conceptual similarity but degrading on rare technical abbreviations. Fusion-Seek's hybrid RRF retrieval achieved Recall@10 of 87.4%, Precision@10 of 84.9%, and MRR of 0.82—a 25.1 percentage-point gain over BM25 and 12.3 over dense-only. GPT-3.5 Cloud RAG marginally surpassed at 89.1% but transmits document content to external servers and carries GDPR Art. 44 cross-border transfer obligations. The 1.7 percentage-point gap represents the measurable cost of privacy-preserving local inference. Failure analysis revealed 14 OCR failures and 11 multi-hop reasoning failures.

Method	R@10	P@10	MRR	Latency
BM25 Keyword	62.3%	58.7%	0.51	0.18s
Dense Semantic	75.1%	71.4%	0.68	0.94s
Hybrid RRF (Ours)	**87.4%**	**84.9%**	**0.82**	**1.12s**
GPT-3.5 Cloud RAG	89.1%	86.3%	0.84	1.8s†
Comm. Ent. Search	81.4%	77.2%	0.74	0.65s

Table 2 Performance Comparison († includes API round-trip)

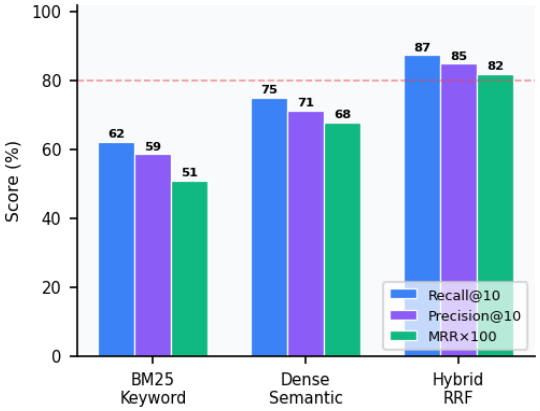

Figure 4 Retrieval Accuracy: Recall@10, Precision@10, and MRR Comparison

8.2.Latency Decomposition

Figure 5 decomposes end-to-end latency. LLM generation dominates at 635 ms (56.7%), producing approximately 150 tokens at 236 tokens/second without GPU. Prompt assembly and SSE streaming account for 192 ms (17.1%). Dense HNSW search consumes 180 ms (16.1%) with ±42 ms variance. Query encoding requires 85 ms (7.6%), BM25 search is negligible at 12 ms (1.1%), and RRF fusion takes only 8 ms (0.7%). Median time-to-first-token is 380 ms. GPU-accelerated vLLM would reduce the LLM component approximately 5×.

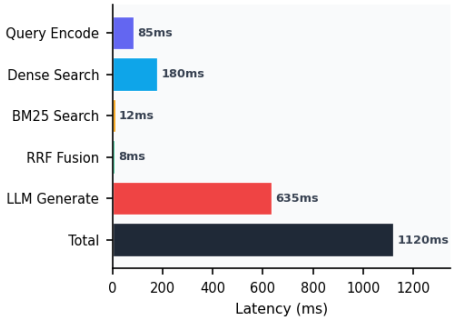

Figure 5 End-to-End Query Latency Decomposition per Processing Stage

8.3.Chunk Size Ablation

Figure 6 presents the chunk size ablation from 100 to 800 tokens. At 100 tokens recall is only 71.2% due to context fragmentation[23]. Performance improves to 79.6% at 200 tokens and to 85.3% at 400 tokens with zero overlap. Adding 100-token overlap provides a 2.1 percentage-point gain to 87.4%, capturing content straddling paragraph transitions. At 600 tokens recall

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 04 April 2026
Page No: 1681-1688
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0220

drops to 83.1% and at 800 tokens it falls to 79.8%.

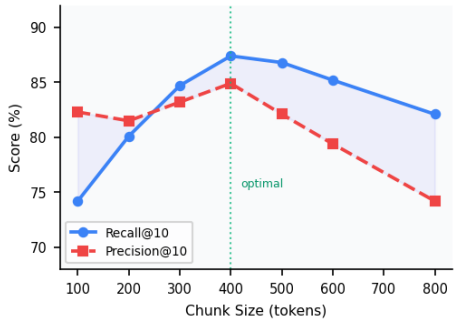

Figure 6 Chunk Size Ablation: Recall@10 and Precision@10 vs. Token Count

8.4. Embedding Model Comparison

paraphrase-MiniLM-L3-v2 encodes at 28 ms/batch (82 MB RAM) but achieves only 71.3% Recall@10. all-mpnet-base-v2 improves recall to 89.2% (+1.8 pp) but increases encoding latency 2.3× and RAM 4.7× to 420 MB—outside the 16 GB deployment envelope without GPU. Multilingual-MiniLM-L12-v2 achieves 85.1% recall at 67 ms/batch and 118 MB, recommended for non-English collections. all-MiniLM-L6-v2 remains optimal: 87.4% recall at 45 ms/batch and 90 MB RAM[13].

8.5. System Capability Radar

Fig. 7 compares Fusion-Seek against cloud RAG and BM25-only across six dimensions. Fusion-Seek achieves maximum scores for Privacy (1.0) and Offline Capability (1.0), both near zero for cloud systems. Accuracy is 0.874 versus 0.891 cloud RAG. Security scores 0.90 for Fusion-Seek versus 0.70 for cloud, attributed to the absence of external API keys and cross-border transmission vectors.

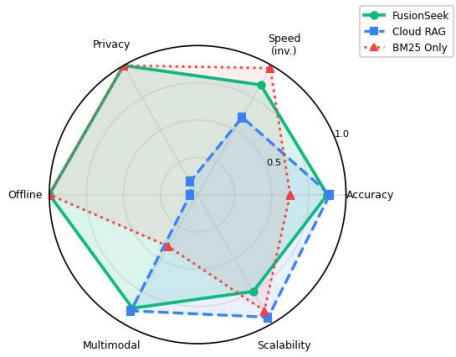

Figure 7 System Capability Radar: Fusion-Seek vs. Cloud RAG vs. BM25-Only

9. Security Analysis

9.1. Threat Model

Three adversary classes are addressed. The external adversary is mitigated by local-only deployment with Nginx-enforced HTTPS for intranet configurations. The malicious insider is partially mitigated by AES-256-GCM at-rest encryption with keys in the OS keychain. The compromised container threat is mitigated by non-root container execution, read-only filesystem mounts, and the absence of any outbound network connections.

9.2. Regulatory Compliance

Under GDPR Article 44, no personal data is transferred outside the organizational network; Article 25 (Privacy by Design) is satisfied by encryption-by-default and minimal metadata collection. Under HIPAA, relevance-score thresholding maintains the minimum necessary standard. Under India's DPDP Act 2023, all data processing occurs within the organisation's own infrastructure. A tamper-evident audit log records every document access and query with UTC timestamp and user identity.

10. Discussion And Future Work

10.1. Limitations

Four limitations warrant acknowledgement: (1) OCR quality for low-resolution scans below 200 DPI and complex multi-column layouts remains a bottleneck; LayoutLMv3 [27] or Donut [26] would improve fidelity at GPU cost; (2) the evaluation corpus is predominantly English, requiring multilingual benchmarking for non-Latin scripts; (3) single-step retrieval cannot answer multi-hop questions, accounting for 11 of 25 observed failures; (4) the in-memory BM25 index limits scalability to approximately 100,000 corpus chunks on 32 GB hardware.

10.2. Future Research Directions

Seven research directions are planned: (1) multilingual retrieval via paraphrase-multilingual-MiniLM-L12-v2; (2) multi-hop reasoning through iterative query reformulation; (3) cross-modal retrieval using CLIP-style [29] aligned image-text embeddings; (4) GPU-accelerated inference via vLLM speculative decoding [30] targeting sub-200 ms latency; (5) federated retrieval across organisational boundaries; (6) domain-specific contrastive embedding fine-tuning for 5–8 pp recall

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 04 April 2026
Page No: 1681-1688
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0220

improvement; (7) Graph RAG integration enriching context with entity-relationship triples.

10.3. Broader Implications

The 1.7 percentage-point Recall@10 gap between Fusion-Seek (87.4%) and GPT-3.5 Cloud RAG (89.1%) is practically negligible for most enterprise applications, while privacy, compliance, and cost advantages of local deployment are substantial. Organisations subject to GDPR, HIPAA, or DPDP Act 2023 can now deploy competitive retrieval capability without regulatory compromise. The feasibility of near-cloud-quality retrieval on a commodity laptop CPU democratises document intelligence for resource-constrained organisations.

Conclusion

This paper has presented Fusion-Seek, a privacy-preserving multimodal document retrieval system achieving enterprise-grade search quality without cloud infrastructure. By integrating BM25 keyword search and dense semantic search through parameter-free Reciprocal Rank Fusion, with a locally hosted RAG pipeline using Mistral-7B-Instruct via Ollama, Fusion-Seek attains Recall@10 of 87.4% on a heterogeneous 250-document evaluation corpus—25.1 percentage points above keyword-only and 12.3 above dense-only baselines. End-to-end query latency of 1.12 seconds demonstrates practical usability on commodity CPU hardware. The four-layer architecture satisfies GDPR, HIPAA, and India's DPDP Act 2023 data sovereignty requirements. As quantized LLMs continue to improve, Fusion-Seek serves as a reference design for privacy-first enterprise document intelligence.

References

[1]. G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” Information Processing and Management, vol. 24, no. 5, pp. 513–523, 1988.
[2]. S. Robertson and H. Zaragoza, “The probabilistic relevance framework: BM25 and beyond,” Foundations and Trends in Information Retrieval, vol. 3, no. 4, pp. 333–389, 2009.
[3]. A. Vaswani et al., “Attention is all you need,” in Advances in NeurIPS, 2017, pp. 5998–6008.
[4]. OpenAI, “GPT-4 Technical Report,” arXiv:2303.08774, 2023.
[5]. Ollama, “Run Llama 2, Mistral, and other models locally,” 2023. [Online]. Available: https://ollama.ai
[6]. G. Gerganov, “llama.cpp: Port of Meta’s LLaMA model in C/C++,” GitHub, 2023.
[7]. N. Reimers and I. Gurevych, “Sentence-BERT: Sentence embeddings using Siamese BERT networks,” in Proc. EMNLP, 2019, pp. 3982–3992.
[8]. Chroma, “ChromaDB: The AI-native open-source embedding database,” 2023. [Online]. Available: https://trychroma.com
[9]. Y. Malkov and D. Yashunin, “Efficient and robust approximate nearest neighbor search using HNSW graphs,” IEEE Trans. PAMI, vol. 42, no. 4, pp. 824–836, 2020.
[10]. G. V. Cormack, C. L. A. Clarke, and S. Buettcher, “Reciprocal rank fusion outperforms Condorcet and individual rank learning methods,” in Proc. ACM SIGIR, 2009, pp. 758–759.
[11]. P. Lewis et al., “Retrieval-augmented generation for knowledge-intensive NLP tasks,” in Advances in NeurIPS, 2020, pp. 9459–9474.
[12]. K. Sparck Jones, “A statistical interpretation of term specificity and its application in retrieval,” Journal of Documentation, vol. 28, no. 1, pp. 11–21, 1972.
[13]. J. Xu and W. B. Croft, “Query expansion using local and global document analysis,” in Proc. ACM SIGIR, 1996, pp. 4–11.
[14]. V. Karpukhin et al., “Dense passage retrieval for open-domain QA,” in Proc. EMNLP, 2020, pp. 6769–6781.
[15]. R. Nogueira and K. Cho, “Passage re-ranking with BERT,” arXiv:1901.04085, 2019.
[16]. O. Khattab and M. Zaharia, “ColBERT: Efficient and effective passage search via contextualized late interaction over BERT,” in Proc. ACM SIGIR, 2020, pp. 39–48.
[17]. T. Formal et al., “SPLADE: Sparse lexical and expansion model for first stage ranking,” in Proc. ACM SIGIR, 2021, pp. 2288–2292.

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 04 April 2026
Page No: 1681-1688
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0220

[18]. J. Ma et al., "Hybrid retrieval with sparse-dense interpolation," in Proc. EMNLP Findings, 2022.
[19]. H. Chase, "LangChain," GitHub, 2022. [Online]. Available: https://github.com/langchain-ai/langchain
[20]. J. Liu, "LlamaIndex," GitHub, 2022. [Online]. Available: https://github.com/run-llama/llama_index
[21]. F. Shi et al., "Large language models can be easily distracted by irrelevant context," in Proc. ICML, 2023, pp. 31210–31227.
[22]. F. Mireshghallah et al., "Quantifying privacy risks of masked LMs using membership inference attacks," in Proc. EMNLP, 2022, pp. 8332–8347.
[23]. C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," Foundations and Trends in TCS, vol. 9, no. 3–4, pp. 211–407, 2014.
[24]. A. Anand et al., "GPT4All: An ecosystem of open source compressed language models," arXiv:2311.04931, 2023.
[25]. R. Smith, "An overview of the Tesseract OCR engine," in Proc. ICDAR, 2007, pp. 629–633.
[26]. G. Kim et al., "OCR-free document understanding transformer," in Proc. ECCV, 2022, pp. 498–517.
[27]. Y. Huang et al., "LayoutLMv3: Pre-training for document AI," in Proc. ACM MM, 2022, pp. 4083–4091.
[28]. A. Radford et al., "Robust speech recognition via large-scale weak supervision," in Proc. ICML, 2023, pp. 28492–28518.
[29]. A. Radford et al., "Learning transferable visual models from natural language supervision," in Proc. ICML, 2021, pp. 8748–8763.
[30]. W. Kwon et al., "Efficient memory management for LLM serving with PagedAttention," in Proc. ACM SOSP, 2023, pp. 611–626.
[31]. A. Q. Jiang et al., "Mistral 7B," arXiv:2310.06825, 2023.
[32]. N. Thakur et al., "BEIR: A heterogeneous benchmark for zero-shot evaluation of IR models," in Proc. NeurIPS, 2021.
[33]. A. Asai et al., "Self-RAG: Learning to retrieve, generate, and critique through self-reflection," in Proc. ICLR, 2024.
[34]. O. Khattab et al., "Demonstrate-search-predict: Composing retrieval and LMs for knowledge-intensive NLP," arXiv:2212.14024, 2022.
[35]. T. Dettmers et al., "QLoRA: Efficient finetuning of quantized LLMs," in Advances in NeurIPS, vol. 36, 2023.