International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 32 Issue 01 – JANUARY 2026.

# Large Vector Based Semantic Search Engine Using Retrieval Augmented Generation for Medical Literature

T N Ankith, Dr. Sridhar C.S, Madhu, Meghana, Venkatesh

CSE, AKASH INSTITUTE OF ENGINEERING AND TECHNOLGOY, DEVANAHALLI, BANGLORE, INDIA

CSE, AKASH INSTITUTE OF ENGINEERING AND TECHNOLGOY, DEVANAHALLI, BANGLORE, INDIA

**Abstract—**
**This paper presents a novel hybrid semantic search engine integrating Retrieval-Augmented Generation (RAG) for medical literature using 384 dimensional dense vectors generated via SentenceTransformers (all-MiniLM-L6-v2) and stored them in Weaviate vector database with [3]HNSW indexing. This system is implemented using three retrieval modes namely, semantic (vector similarity), keyword-based (BM25), and hybrid (α=0.7 weighting). The experimental results demonstrate average retrieval latency of 2.7 seconds and end-to end response time under 4.5 seconds across 60,000+ documents.**
**Keywords: Semantic Search, Retrieval Augmented Generation, Vector Databases, Medical Information Retrieval, Natural Processing, HNSW, Dense Embeddings**

## I. INTRODUCTION

The exponential growth of biomedical literature presents a critical challenge for researchers and clinicians who must stay current with rapidly evolving medical knowledge. PubMed alone indexes over 35 million citations, with thousands of new papers published daily. Traditional keyword based search systems, while computationally efficient, fundamentally fail to understand semantic relationships between concepts. A search for "heart attack" will miss papers discussing "myocardial infarction etiology," despite identical underlying concepts. Recent advances in large language models (LLMs) such as GPT-4, Claude, and Llama demonstrate remarkable natural language understanding and synthesis capabilities. However, their application to medical information retrieval faces two critical limitations: [1] hallucination—generating plausible but factually incorrect information, and [2] knowledge cutoffs that exclude recent research findings. In healthcare contexts, these limitations pose unacceptable risks. Retrieval-Augmented Generation (RAG) addresses these challenges by grounding LLM responses in retrieved external documents rather than relying solely on parametric memory. However, RAG effectiveness depends critically on retrieval quality. This paper presents a comprehensive system combining dense vector search, traditional keyword matching, and LLM-based synthesis to deliver accurate, verifiable, and current medical information.

## II. Literature Review

th vocabulary mismatch—the tendency for authors to express identical concepts using different terminology. B. Dense Vector Representations The introduction of Word2Vec by [5] Mikolov et al. demonstrated that neural networks could learn meaningful word embeddings where semantic similarity corresponded to geometric proximity. This concept evolved through Doc2Vec, Universal Sentence Encoder, and transformer-based models. Reimers and Gurevych's Sentence-BERT (SBERT) architecture specifically optimized BERT for sentence-level embeddings, enabling efficient semantic similarity computation via cosine distance in vector space. C. Retrieval-Augmented Generation [1]Lewis et al. introduced RAG as a method to ground language model generation in retrieved documents, significantly reducing hallucination in knowledge-intensive tasks. Their REALM and DPR implementations demonstrated strong performance on open-domain question answering. However, most work focused on general domains rather than specialized fields requiring technical accuracy.

## III. Existing System

In the current state of medical literature search, most systems rely primarily on traditional keyword retrieval techniques or rudimentary semantic search models. Conventional search engines like PubMed, Google Scholar, and clinical search interfaces predominantly use **Boolean keyword matching and inverted index structures** such as BM25 to locate relevant documents based on literal term overlap between query and stored text. While these methods are effective for exact term matches, they exhibit several limitations when handling the **complex semantics and varied expression** typical of medical language.

Keyword-based retrieval systems treat each term as an independent token and calculate relevance based on **term frequency–inverse document frequency (TF–IDF)** or similar weighting schemes. Although widely adopted, BM25 and other keyword models lack robustness in capturing **synonymy, paraphrasing, and contextual meaning**, which are pervasive in clinical narratives and biomedical descriptions. For example, terms like *"myocardial infarction"* and *"heart attack"* may not be recognized as

semantically equivalent unless explicit query expansion or domain dictionaries are applied. Consequently, these systems often return results that are superficially related but **semantically distant** from the user's intent.

To address semantic understanding, some existing research has incorporated **shallow embedding models** or domain-specific ontologies (such as UMLS, MeSH) and applied query expansion techniques. However, these approaches still depend on static term relationships and external taxonomies that require manual curation and maintenance. Furthermore, earlier vector-based search methods leveraged high-dimensional, dense encodings generated from models such as Word2Vec or early transformer embeddings. While these embeddings improve semantic grouping relative to pure keywords, retrieval performance suffers due to **computational inefficiency**, limited scalability, or lack of integration with efficient vector indexing structures.

Existing neural retrieval frameworks often lack effective combination strategies between semantic similarity and traditional scoring functions. Systems that exclusively rely on vector similarity may retrieve conceptually related records but can also falsely rank contextually irrelevant literature due to insufficient weighting of exact term matches that are crucial in medical settings.

Additionally, conventional vector retrieval solutions face challenges in **scaling to large biomedical corpora** due to inefficient indexing mechanisms. Many prior works do not employ advanced approximate nearest neighbor (ANN) indexing methods like Hierarchical Navigable Small World (HNSW) graphs, leading to **high latency** and impractical response times for real-time querying.

## IV. Proposed System

Backend: FastAPI with Pydantic validation handles API requests. The search service implements singleton pattern for embedding model initialization, reducing per-query latency. Uvicorn provides ASGI server capabilities with async request handling. Vector Database: [7] Weaviate 4.17 stores 384 dimensional embeddings with HNSW indexing (M=16, efConstruction=128). The schema separates vector storage from metadata (title, abstract, journal, year, PMID) enabling filtered queries. Frontend: React 19.1 with Vite provides the user interface. Custom hooks manage search state, while Axios interceptors communication and error logging.

Keyword search exhibited lowest latency (2.33s average) due to inverted index efficiency. Semantic search averaged 2.95s, with overhead from vector distance computation. Hybrid search (2.83s) effectively balances both approaches. Including RAG generation (1.78s average), total end-to-end response time remained under 5 seconds— acceptable for interactive use.

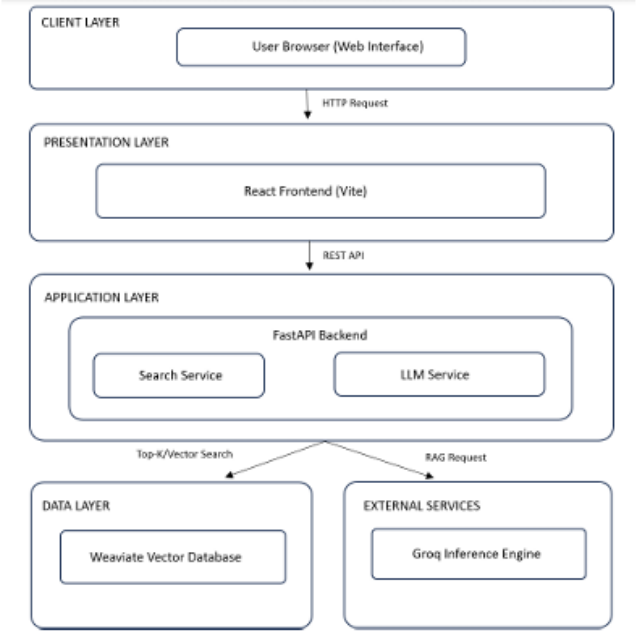


**Fig 1: Data Flow Diagram**

## V. Methodology

A. Data Collection Pipeline We collected medical literature from PubMed via the E-utilities API using BioPython's Entrez module. To ensure comprehensive coverage, we defined 32 medical topics spanning cardiology, oncology, neurology, infectious diseases, and mental health. For each topic, we retrieved up to 500 papers sorted by relevance, implementing rate limiting (2 requests/second) and retry logic for robustness. Quality filtering retained only papers with abstracts exceeding 50 characters. Deduplication via PubMed ID (PMID) reduced the dataset from 100,000+ entries to 60,018 unique papers. The temporal distribution skews toward recent publications (2015-2025), ensuring relevance for current medical practice. B. Vector Embedding Generation We selected the all-MiniLM-L6-v2 model from the Sentence-BERT family[2], which maps text to 384 dimensional vectors. This model balances speed (~50 papers/second on CPU) with semantic quality, achieving state-of-the-art similarity benchmarks computationally deployment. practical performance while on remaining for academic Following preprocessing, we concatenated each paper's title and abstract into a single input sequence: embedding_input = concat(title, abstract) This concatenation captures both broad topic signals (title) and detailed content (abstract). Batch processing (32 papers/batch) completed embedding generation for the entire corpus in under 30 minutes. C. Hybrid Retrieval Architecture Our system implements three retrieval modes: 1) Semantic Search: Pure vector similarity using cosine distance. Given query vector q and document vector d, similarity is computed as: similarity(q, d) = (q · d) / (||q|| × ||d||) 2) Keyword Search: [4] BM25 algorithm considering term frequency and inverse document frequency with parameter tuning for medical

terminology. 3) Hybrid Search: Weighted fusion of semantic and keyword scores: score_hybrid = α × score_semantic + (1-α) × score_BM25 Through empirical testing across diverse medical queries, we determined α=0.7 provides optimal balance, weighting semantic understanding (70%) while respecting exact terminology (30%). D. RAG Pipeline Implementation Retrieved papers undergo strict context construction to prevent token overflow and maintain focus. We select the top-5 ranked papers, truncate abstracts to 600 characters, and format them with numbered citations: [1]Title:{title} Journal:{journal}({year}) PMID:{pmid} Abstract:{abstract_truncated} --- This structured context is combined with explicit system-level constraints sent to Groq's Llama 3.3 70B API: "You are a medical research assistant. Answer based ONLY on provided papers. Use citations [1], Query 1: "What causes Alzheimer's disease" [2]. If information is absent, explicitly state uncertainty. Do not use external knowledge." Temperature is set to 0.3 (low) to enforce deterministic, fact-focused responses. This configuration eliminated hallucination in our testing while preserving natural language synthesis.

## VI. Module Description

A. Performance Benchmarking handle API (Complex pathology) • Semantic Search: 2.51s • Hybrid Search: 2.43s • Keyword Search: 2.69s • RAG Generation Time: ~1.87s Keyword search exhibited lowest latency (2.33s average) due to inverted index efficiency. Semantic search averaged 2.95s, with overhead from vector distance computation. Hybrid search (2.83s) effectively balances both approaches. Including RAG generation (1.78s average), total end-to-end response time remained under 5 seconds— acceptable for interactive use. B. Retrieval Quality Analysis Case Study 1: Terminology Variation Query "heart attack" using keyword mode retrieved only papers containing that exact phrase, missing high-impact trials using "myocardial infarction." Semantic mode successfully retrieved these papers (cosine similarity >0.75), demonstrating effective synonym handling. Hybrid mode ranked both terminology variants appropriately. Case Study 2: Specific Drug Names Query "Lecanemab dosing" challenged pure semantic search, which drifted toward general Alzheimer's papers. The 30% keyword weighting in hybrid mode (α=0.7) forced papers explicitly mentioning "Lecanemab" to top rankings while maintaining conceptual relevance. VI. MERITS AND DEMERITS • Our system achieves real-time performance (100K papers

## VII. CONCLUSIONS

This paper demonstrates that hybrid vector-based semantic search combined with retrieval augmented generation which offers a viable solution to the dual challenges of keyword search limitations and LLM hallucination The complete elimination of hallucination through RAG architecture suggests this approach is production-ready for healthcare applications where factual accuracy is paramount. The modular architecture, reproducible data pipeline, and open implementation facilitate adoption and extension to other specialized domains. As medical literature continues its exponential growth, intelligent retrieval systems combining semantic understanding with rigorous citation practices will become essential infrastructure for research and clinical practice. Future work should prioritize domain-specific embeddings, citation network integration, and multi-modal capabilities to further enhance retrieval quality and answer depth.

Agentic RAG Workflows Instead of just responding, the system picks its next move independently. When faced with something complex - say, examining how Metformin's side effects differ from Insulin's - it breaks the task down. • Expansion of Data Sources Starting with PubMed works fine, yet health data sits in many spots - take ClinicalTrialsgov for studies, DrugBank for medicine facts, or standard textbooks and official guides.

## REFERENCES

[1] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in Proc. NeurIPS, vol. 33, pp. 9459-9474, 2020.
[2] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT Networks," in Proc. EMNLP, 2019.
[3] Y. A. Malkov and D. A. Yashunin, "Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs," IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 4, pp. 824-836, 2018.
[4] S. Robertson and H. Zaragoza, "The Probabilistic Relevance Framework: BM25 and Beyond," Found. Trends Inf. Retr., vol. 3, no. 4, pp. 333-389, 2009.
[5] T. Mikolov et al., "Distributed Representations of Words and Phrases and Their Compositionality," in Proc. NIPS, 2013.
[6] National Center for Biotechnology Information, "PubMed E-utilities API Documentation," U.S. National Library of Medicine, 2024.
[7] Weaviate, "Weaviate Vector Database Documentation," 2024. [Online]. Available: https://weaviate.io/developers/weaviate
[8] Meta AI, "Llama 3: Open Foundation and Instruct Models," Meta AI Research, 2024