Faculty of Engineering Artificial Intelligenc

Department of Artificial Intelligence

& Data Science



An intelligent Arabic medical chatbot

(RAG)

A senior project report – Submitted in Fulfillment of the

Requirements for a Bachelor's Degree

in Computer Engineering

Prepared by

Rshdy ali

Supervised by

Dr. Majida Albakoor

Eng. Ahmad Shekha

2026-2025

الجامعة السورية الخاصة
SYRIAN PRIVATE UNIVERSITY

كلية الهندسة، قسم الذكاء الاصطناعي

قسم الذكاء الاصطناعي وعلوم البيانات

شات بوت طبي عربي ذكي

(RAG)

تقرير مشروع التخرج - مقدم استيفاءً لمتطلبات الحصول على درجة البكالوريوس

في هندسة الحاسوب

إعداد الطالب:

رشدي علي

بإشراف

د. ماجدة البكور

م. احمد شيخه

2026-2025

Supervisor Certification

I Certify that the preparation of this project entitled **An intelligent Arabic medical chatbot (RAG)**

Prepared by **Rshdy ali** was made under my supervision at Faculty of Engineering Artificial Intelligenc

in partial Fulfillment of the Requirements for the Degree

of Bachelor of Artificial Intelligence Engineering.

Name: .....

Signature: .....

Name: .....

Signature: .....

Abstract

This project showcases the design and implementation of an expert system for medical consultations in Arabic, aiming to provide accurate and reliable answers drawn from a massive database containing over 700,000 medical records. The content is presented in a format (RAG). Arabic medical content faces significant challenges due to the accuracy of general linguistic models (LLMs) and their tendency towards hallucination or language mixing. To address these challenges, an integrated engineering infrastructure based on Retrieval-Augmented Generation (RAG) technology has been developed.

The system features a hybrid search engine that combines semantic search using the Multilingual-E5 embedding model with traditional textual search (BM25) to ensure the accuracy of medical terminology. The project also includes an advanced data pipeline that filters out linguistic noise and personal narratives to focus the search on pure medical facts.

The system's performance was evaluated through a blind test using multiple scientific metrics, including semantic similarity (which achieved an average accuracy exceeding 72%).

الملخص

يستعرض هذا المشروع تصميم وتنفيذ نظام خبير للاستشارات الطبية باللغة العربية، يهدف إلى تقديم إجابات دقيقة وموثوقة مستمدة من قاعدة بيانات ضخمة تحتوي على أكثر من **700,000سجل طبي** .يواجه المحتوى الطبي العربي تحديات كبيرة في دقة النماذج اللغوية العامة (LLMs) وميلها للهلوسة (Hallucination) أو خلط اللغات. لمعالجة هذه التحديات، تم بناء بنية تحتية هندسية متكاملة تعتمد على تقنية Retrieval-Augmented Generation (RAG).

يتميز النظام باستخدام محرك بحث هجين (Hybrid Search) يدمج بين البحث الدلالي (Semantic Search) باستخدام نموذج التضمين **Multilingual-E5**، والبحث النصي التقليدي (**BM25**) لضمان دقة المصطلحات الطبية.

كما يتضمن المشروع خط معالجة بيانات متقدم يقوم بتصفية الضوضاء اللغوية و تنظيف النصوص و التركيز البحث على الحقائق الطبية الصرفة.

تم تقييم أداء النظام من خلال اختبار أعمى باستخدام مقاييس علمية متعددة شملت Semantic Similarity والذي حقق متوسط دقة يتجاوز 72%

رسالة الشكر

إلى أبي

أبي لم يكن نبيًّا،
ولكن... عليه السلام.

سلامٌ على قلبٍ علّمني أن أتوكل على الله قبل أن أتوكل على نفسي،
وسلامٌ على كتفٍ حمل عني ثقل الأيام دون أن يشتكي،
وسلامٌ على دعاءٍ صامتٍ كان يصلني وأنا لا أعلم.

أبي، كنتَ السبب بعد الله في ثباتي، وما وصلتُ إليه اليوم
هو ثمرة تعبك، وصبرك، وإيمانك بي.

اللهم احفظ أبي،
واجعل ما قدمه لي في ميزان حسناته، واجزه عني خير الجزاء
كما ربّاني صغيرًا وساندني كبيرًا.

إلى أمي

أمي...
ليست مجرد كلمة، بل وطنٌ حين يضيق العالم.

أنتِ الدعاء الذي لا يُرد، والتعب الذي لم يشتكِ،
والحب الذي سبق أحلامي واحتواها.

كل خطوة نجاح خطوتها
كانت ممهورة بصبرك،
وكل إنجاز أحمله
تحته قلبك الذي لم يتعب من العطاء.

إن كان لهذا العمل قيمة،
فأعظم قيمته أنكِ خلفه.
فشكرًا بحجم قلبك...
وشكرًا أكثر مما تسعه الكلمات.

إلى عمي العزيز **رشيد**، وإلى **مرة عمي** الغالية..

لا يمكنني أن أغفل عن صاحب الفضل والقلب الكبير.

شكراً لك يا عمي على وقفتك التي كانت لي سنداً، وعلى كلماتك التي لطالما أعطتني الثقة للمضي قدماً.

وشكراً لكِ يا زوجة عمي على دعواتك الصادقة، وعلى روحك الطيبة التي كانت تملأني بالتفاؤل في كل مرة نلتقي فيها.

أنتما لم تكونا مجرد أهل، بل كنتما عائلة آمنت بي وبحلمي. أسأل الله أن يحفظكما لي، وأن يديمكما تاجاً فوق رأسي.

إلى أخي المهندس حسين

شكرًا لأنك كنت دائمًا مثالًا للاجتهاد والمسؤولية.
وجودك في حياتي مصدر فخر ودافع،
ونصائحك ودعمك تركت أثرًا جميلًا لا يُنسى.

إلى أخي المحامي جهاد

شكرًا لوقوفك الدائم إلى جانبي،
ولحكمتك التي تمنحني الطمأنينة،
ولكلماتك الصادقة التي تأتي في وقتها.
أعتز بك أخًا وسندًا.

إلى **روبين شيرو**..

إلى العزيزة المحامية **روبين شيرو** ..مع ختام مشروعي، لا يسعني إلا أن أعبر عن امتناني الكبير لوجودكِ في حياتي.

لقد استلهمت من شغفكِ وإصراركِ الكثير من القوة.

شكراً لأنكِ كنتِ خير داعم وخير مستشار، وشكراً لقلبكِ الكبير الذي كان دائماً يمنحني الأمان.

إلى أخي الغالي، الدكتور مصطفى مسلم..

أكتب لك هذه الكلمات ومعها فيضٌ من الفخر الذي لا تسعه السطور

فمع انتهاء مشروعي، أجد أن شكرك واجبٌ لا يكتمل النجاح بدونه.

إلى عائلة الأصول والمواقف النبيلة.. عائلة عمي الغالي (أبو جهاد)..

مع انتهاء مشروعي وبلوغ لحظة النجاح، يفيض قلبي بالامتنان لكم.

لا أجد كلمات توفيكم حقكم، فقد كنتم لي العائلة التي أستمد منها الدعم والتشجيع، وكنتم خير سندٍ بدعواتكم الصادقة وكلماتكم التي كانت تمنحني القوة للاستمرار.

شكراً لعمي (أبو جهاد) على قلبه الكبير، ولعائلتكم الكريمة فرداً فرداً على تلك الروح الطيبة التي أشعرتني دائماً أنني لست وحدي في هذا الطريق.

نجاحي اليوم هو ثمرة لمحبتكم ووجودكم بجانبي، وأسأل الله أن يديم عزكم ويجعل بيوتكم دائماً منبعاً للجود والأفراح

(الدكتور جهاد مسلم , المهندس محمد مسلم , السيد خلوسي مسلم , الغالية مريم مسلم , الغالية نازلية عبدو , المهندسة جمانا شيخو , المحامية اوريفان حسون).

الدكتورة ماجدة البكور

إلى سعادة الدكتورة **ماجدة البكور** ..أتقدم لكِ بخالص الشكر وعميق العرفان على إشرافكِ الكريم؛ فقد كانت توجيهاتكِ الحكيمة هي المنارة التي أضاءت لي دروب البحث، وملاحظاتكِ الدقيقة هي الصقْل الذي ارتقى بهذا العمل إلى مستواه المنشود. شكراً لصبركِ الذي لا ينضب، ولعلمكِ الذي لم تبخلي به، فلكِ مني كل التقدير والامتنان.

المهندس أحمد الشيخة

الأخ العزيز المهندس **أحمد الشيخة** ..أقف اليوم بامتنان كبير لأعبر لك عن خالص شكري وتقديري؛ فلم تكن مجرد زميل أو مسؤول، بل كنت الداعم والمحفز الأول طوال رحلتي معك. إن بصمتك الواضحة وتوجيهاتك المستمرة كانت حجر الزاوية في إتمام هذا العمل ونجاحه. شكراً لأنك كنت خير سند، ومثالاً يُحتذى به في العطاء المهني.

Contents

Prepared by 1

Supervised by 1

Introduction 11

1.1 General Background of the Project 11

1.2 Project Description 11

1.3 Project Importance 11

1.4 Project Objectives 12

1.5 Project Implementation Methodology 12

1.6 Project Scope 13

1.7 Gantt Chart 14

Previous Studies 17

2.1 Introduction 17

2.2 Studies Related 17

2.3 Summary of previous studies 20

System Architecture and Methodology 22

3.1 Retrieval-Augmented Generation (RAG) Framework 22

3.2 What Is Retrieval-Augmented Generation (RAG)? 22

3.3 Combining Internal, External Resources 23

3.4 How RAG Works 23

3.5 Keeping Sources Current 24

3.2 Hybrid Search Drive 26

3.2.1 Semantic Search using the Multilingual-E5 Model 26

3.2.2 Lexical Search using the BM25 algorithm 26

3.2.3 Hybrid Search and Reciprocal Rank Fusion (RRF) 27

3.2.4 Fuzzy Matching 31

3.2.5 Query Rewriting (QR) 31

3.3 Cross-Encoder Reranker 32

3.3.1 Definition : 32

3.3.2 How It Works 33
3.3.3 Project Significance: 34
3.3.4 Technical Benefit: 34
5 Generation & LLM Inference 36
5.1 Hugging Face Inference API 36
5.2 The Big Language Model: DeepSeek-V3 36
5.3 Medical Prompt Engineering 37
5.4 The Generation Process 37
Data set & Pre-processing 41
4.1 Data Engineering & Pre-processing 41
4.2 Knowledge Base Indexing & Vector Database 42
Evaluation methodology 45
5.1 Semantic Similarity 45
5.2 BLEU (Bilingual Evaluation Understudy) 46
5.3 ROUGE (Recall-Oriented Understudy for Gisting Evaluation) 46
5.4 Precision 47
5.5 Recall 47
5.4 MRR (Mean Reciprocal Rank) 48
Results & Performance Analysis 50
6.1 Core Success Metrics 51
6.2 Retrieval Quality Analysis 51
6.3 Language Generation Scores Assessment 52
Challenges & Solutions 56
7.1 Challenges and solutions 56
7.2 Future Recommendations 57
7.3 Conclusion 57
Bibliography 58

Figure 1Gantt chart 15
Figure 2 retrieval-augmented generation, LLMs embedding and reranking models, storing knowledge in a vector database 23
Figure 3 Retrieval-augmented generation combines LLMs with embedding models and vector databases. 24
Figure 4 hybrid search work 27
Figure 5 Sort documents 28
Figure 6 Assign rank positions 29
Figure 7 Apply the RRF formula 29
Figure 8 Add rank contributions 30
Figure 9 Query Rewriting 32
Figure 10 Reranker work 33
Figure 11 hybrid and reranking logic 35
Figure 12 query processing sequence 38
Figure 13 query processing flow 39
Figure 14 indexing ang database flow 43
Figure 15 Semantic similarity equation 45
Figure 16 BLEU equation 46
Figure 17 ROUGE equation 47
Figure 18 Precision equation 47
Figure 19 Recall equation 48
Figure 20 MRR equation 48
Figure 21 RAG metrics 50
Figure 22 streamlit system interface 53
Figure 23 Test 1 53
Figure 24 Test 2 54
Figure 25 Test 3 54

Table 1 Study Summary..... 20
Table 2 challenges and solution..... 56

Chapter One: Introduction

Introduction

1.1 General Background of the Project

In recent years, Large Language Models (LLMs) have experienced tremendous growth, but they still face significant challenges in sensitive fields such as medicine, especially when dealing with Arabic. The problem lies in the phenomenon of "technological hallucination," where a model might confidently present incorrect medical information. This has highlighted the need to integrate linguistic models with reliable and up-to-date data sources.

This is known as the RAG technique, to ensure that the medical answer is based on scientific reference and not merely on linguistic probabilities.

1.2 Project Description

An intelligent medical consultation system based on a sophisticated Retrieval-Augmented Generation (RAG) infrastructure. The system processes user inquiries in Arabic, then searches a huge medical database (containing more than 700,000 records) to extract relevant information, and reformats it using an advanced language model (DeepSeek-V3) ,The system is equipped with a hybrid search engine that combines semantic understanding and precise text matching (BM25).

1.3 Project Importance

The project's importance is evident in several key points:

- ✓ Bridging the digital divide: the scarcity of reliable and interactive Arabic medical content.
- ✓ Accuracy and safety: reducing the risk of misinformation by restricting the model to a specific medical context.
- ✓ Decision support: It can act as an initial aid for patients, guiding them or providing them with sound preliminary information before consulting a doctor.
- ✓ Handling big data: The ability to retrieve information from among hundreds of thousands of records in a matter of seconds.

1.4 Project Objectives

Objectives:

1. Building a vector database capable of storing and processing 700,000 medical documents.

2. Developing a hybrid search engine that balances understanding meaning with purely medical terminology.

3. Designing a rigorous prompt engineering that prevents language mixing and enforces the use of Modern Standard Arabic.

4. Creating an automated evaluation system to measure the quality of responses using scientific metrics (Recall , Precision, Semantic Similarity).

1.5 Project Implementation Methodology

The Methodology consists of five key phases:

Phase 1: Data Pre-processing & Engineering

1. **Cleaning:** Removal of duplicates (0.03%) and null values.

2. **Sampling:** Applying **Stratified Sampling** to maintain the proportional representation of 90 medical categories.

3. **Chunking:** Splitting long medical texts (some reaching 32k characters) into optimized chunks of 512 words with a 128-word overlap to preserve context.

Phase 2: Knowledge Base Construction (Indexing)

1. Transforming text chunks into high-dimensional vectors using the Multilingual-E5 model.

2. Storing and indexing these vectors in **ChromaDB** for sub-second similarity searches.

Phase 3: The Retrieval Pipeline (Two-Stage Search)

Developing the RRF algorithm to integrate semantic and textual search results.

1. Vector search
2. BM25

Phase 4: LLM Integration & Prompt Engineering

1. Connecting the system to hugging face DeepSeek-V3.

2. Designing a **Strict System Prompt** that forces the model to use only the provided context and prevents it from providing unauthorized medical prescriptions.

Phase 5: Deployment & Validation

1. Building the **Streamlit** dashboard.

2. Conducting **System Debugging** to monitor GPU/CPU performance and ensure "Zero-Hallucination" during medical queries.

1.6 Project Scope

The scope of this project encompasses the design and development of an automated medical inquiry system. It involves processing a large-scale dataset of **700000 medical records** across **90 different specialties**. The project includes building a high-performance **Vector Database (ChromaDB)**, implementing a **two-stage retrieval logic**, and deploying a user-friendly **Streamlit** web interface.

The system is designed to operate locally to ensure maximum data privacy and security.

Key Inclusions :

- ✓ **Data Layer:** Cleaning, chunking, and stratified sampling of 700k+ documents.
- ✓ **AI Engine: Integration of Multilingual-E5 for embeddings and** DeepSeek-V3 **for response**
- ✓ **Generation Optimization:** Implementation of hybrid search and a **Cross-Encoder Re-ranker** for clinical accuracy.
- ✓ **UI/UX:** Developing a Bilingual-ready interface with RTL support for Arabic users.

1.7 Gantt Chart

A Gantt chart is a type of bar chart that illustrates a project timeline. The horizontal axis represents the schedule (in days, weeks, or months), while the vertical axis represents the tasks to be completed. Each task appears as a bar that runs from the start date to the end date, providing an immediate visual overview of the workflow.

Software projects are characterized by their complex nature and numerous overlapping technical details, and this is where the importance of the Gantt chart comes to the fore in the following points:

1. Credential Management

In programming, code cannot be tested before it is written, nor can it be written before the database is designed. A Gantt chart helps define the relationships between tasks, so each programmer knows when their colleague must finish a particular part before they can begin the next.

2. Effective Resource Allocation

The chart prevents tasks from piling up on a single programmer at a given time. Through visual observation, the project manager can see if the "UI developer" is overloaded with tasks in the third week, while the "server developer" has free time, allowing for task redistribution and avoiding bottlenecks.

3. Identifying the Critical Path

The chart helps identify the essential tasks that, if delayed by even one day, will delay the entire project. This helps the development team focus on the highest priorities and avoid wasting time on secondary features that could compromise the final delivery date.

4. Tracking Actual Progress Against the Plan

A Gantt chart provides a way to compare what has actually been accomplished with what was planned. In software projects that rely on "continuous delivery," the chart makes it easy to see if

the team is on track or needs to adjust its course.

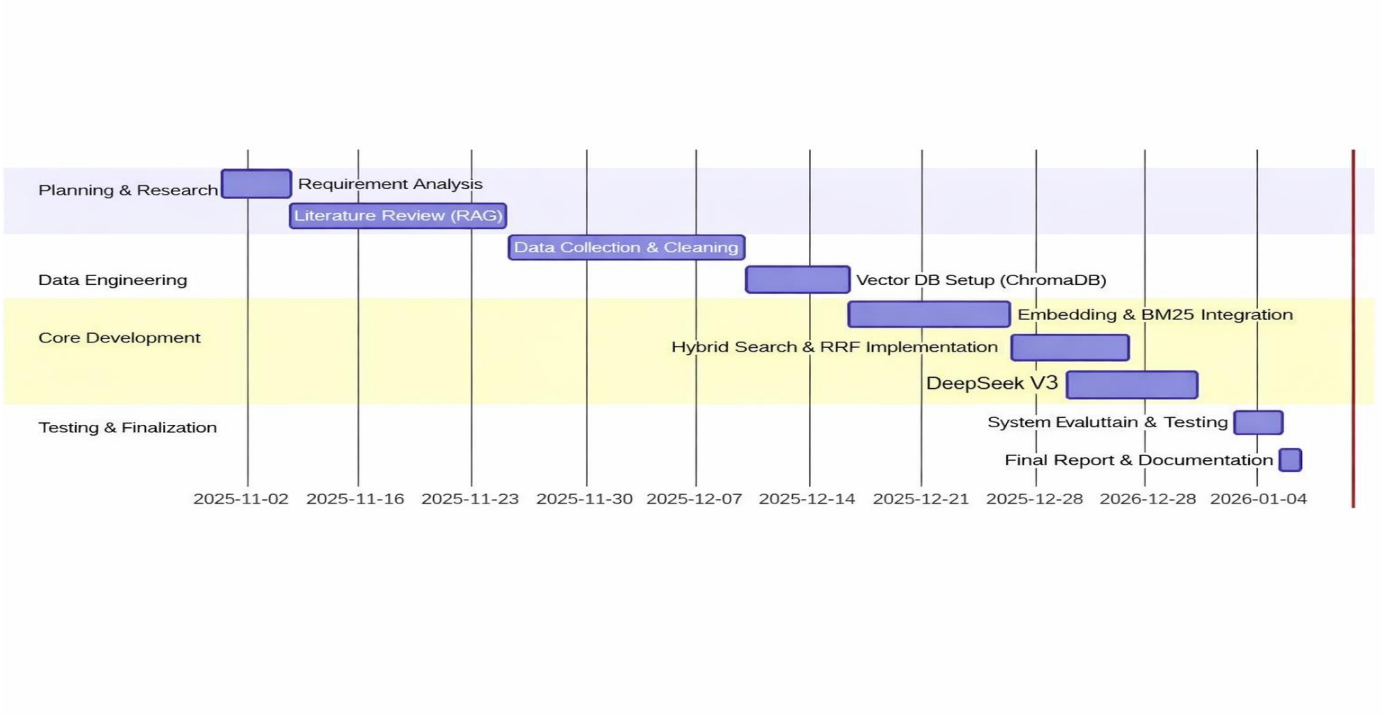


Figure 1Gantt chart

Chapter Two: Previous Studies

Previous Studies

2.1 Introduction

Previous studies are the foundation upon which any successful scientific project is built. They provide a comprehensive overview of research achievements in a specific field and help identify gaps and areas that need development. In this chapter, we will review the most prominent research related to summarization, classification, recommendation, and question-and-answer systems in the news field, with a focus on applications supporting the Arabic language.

2.2 Studies Related

Study title 1: Retrieval-Augmented Generation for Large Language Models

Methodology

This paper adopts a comprehensive survey methodology for the evolution of RAG systems, dividing it into three main phases:

- Naive RAG: This follows a simple linear process consisting of retrieval and generation. Relevant documentation is searched for and integrated into the model's prompt.
- Advanced RAG: This introduces improvements both before and after retrieval. This includes re-ranking results and expanding queries to ensure the quality of retrieved information.
- Modular RAG: This provides greater flexibility by adding specialized modules such as a recurring search module, a routing module, and a modification module, which allow the system to switch between different functions based on the type of question (Yun & Xinyu, 2023).

Obstacles and Challenges

- Retrieval Precision: The risk of retrieving irrelevant or contradictory information, which can mislead the model.
- Context Window: As the amount of retrieved information increases, the model may struggle to process all the details or may ignore information in the middle of the text.

Study title 2：MedRAG: Enhancing Retrieval-augmented Generation with Knowledge Graph-Elicited Reasoning for Healthcare Copilot

1. Methodology

The research presents MedRAG, an advanced framework that integrates Retrieval-Assisted Text Generation (RAG) technology with Knowledge Graphs (KG) to improve the accuracy of medical diagnoses.

The methodology consists of three main pillars:

- Knowledge Graph Construction (KG Construction): This involves creating a four-tiered hierarchical structure (symptoms, diseases, medical categories, and relationships) that focuses specifically on "diagnostic differences" between diseases with similar symptoms.
- Hybrid Retrieval: The system not only retrieves similar electronic health records (EHRs) but also extracts inference pathways.
- Proactive diagnostic questioning

Obstacles & Challenges

- Complexity of graph construction: Building hierarchical medical graphs requires high accuracy and significant effort to ensure the comprehensiveness of the relationships between diseases and symptoms.
- Real-time inference challenges: Integrating search across large databases with graph inference increases computational demands. The need for structured data: System success depends heavily on the quality of electronic health records (EHRs) (Chunyan, 2025).

Study title 3：RGAR: Recurrence Generation-augmented Retrieval for Factual-aware Medical Question Answering

Methodology

The research presents a framework called RGAR, a system based on "repeated generation-assisted retrieval." The methodology aims to address the information gap between medical questions and theoretical knowledge through the following steps:

- Factual Knowledge Generation: Instead of direct searching, the model first generates "medical facts" or concepts relevant to the question based on the internal knowledge of the Large Language Model (LLM) (Deyu & Wenwen , 2025).

- Recurrence Retrieval: The system uses these generated facts as optimized queries to search large medical databases. This process is repeated to refine the results and ensure the retrieval of the most relevant texts for the clinical case.
- Integration & Generation: Facts extracted from patient records (EHRs) are integrated with the retrieved methodological knowledge to guide the model in formulating the final answer.

Obstacles & Challenges

- Resource and time consumption: Because the process relies on repeated generation before retrieval, it consumes more time.
- Risk of generating initial false facts: The success of the retrieval depends heavily on the quality of the facts initially generated by the model.
- Formatting complexity: The system requires careful adjustment to ensure that retrieved texts do not exceed the form's "Context Window," especially when dealing with long and complex medical texts.

Study title 4: Biomedical Literature Q&A System Using Retrieval-Augmented Generation (RAG)

Methodology

This research focuses on building a Q&A system specialized in biomedical literature using the RAG technique:

- Data Source Compilation: Integrating diverse sources, including PubMed articles, medical encyclopedias, and curated Q&A datasets, to ensure comprehensive knowledge (Yen Chih & Shreyash, 2024).

- Retrieval Pipeline: Using MiniLM models to create semantic embeddings.

- Deploying the FAISS database for rapid vector searches to find the most relevant information.
- Answer Generation: Using the Mistral-7B-v0.3 model after fine-tuning.

Obstacles & Challenges

- Handling complex terminology: Despite improvements, models still sometimes struggle to accurately distinguish between highly specialized biomedical terms that may sound similar but have different meanings.

- Limited resources: Although QLoRA is used, processing and storing large volumes of PubMed articles still requires a robust vector management infrastructure.

2.3 Summary of previous studies

We summarize the studies in the following table, which shows the most important large language models and the embedding models used.

Table 1 Study Summary

large language models	embedding models
GPT-4 **GPT-3.5-Turbo**	**MimiLM-L6-v2**
Mistral-7B-v0.3	**multilingual-e5-large-instruct**
Llama2-40B	**SBERT(sentence-BERT)**

Chapter three: System Architecture and Methodology

System Architecture and Methodology

3.1 Retrieval-Augmented Generation (RAG) Framework

What Is Retrieval-Augmented Generation, aka RAG?

Retrieval-augmented generation is a technique for enhancing the accuracy and reliability of generative AI models with information from specific and relevant data sources.

How It Got Named 'RAG'

Patrick Lewis, lead author of the 2020 paper that coined the term, apologized for the unflattering acronym that now describes a growing family of methods across hundreds of papers and dozens of commercial services he believes represent the future of generative AI.

We definitely would have put more thought into the name had we known our work would become so widespread, Lewis said in an interview from Singapore, where he was sharing his ideas with a regional conference of database developers.

We always planned to have a nicer sounding name, but when it came time to write the paper, no one had a better idea," said Lewis, who now leads a RAG team at AI startup Cohere.

3.2 What Is Retrieval-Augmented Generation (RAG)?

Retrieval-augmented generation is a technique for enhancing the accuracy and reliability of generative AI models with information fetched from specific and relevant data sources.

In other words, it fills a gap in how LLMs work. Under the hood, LLMs are neural networks, typically measured by how many parameters they contain. An LLM's parameters essentially represent the general patterns of how humans use words to form sentences.

That deep understanding, sometimes called parameterized knowledge, makes LLMs useful in responding to general prompts. However, when users need authoritative, source-grounded answers rather than broad knowledge alone, RAG can provide the necessary depth and accuracy.

3.3 Combining Internal, External Resources

Lewis and colleagues developed retrieval-augmented generation to link generative AI services to external resources, especially ones rich in the latest technical details.

The paper, with coauthors from the former Facebook AI Research (now Meta AI), University College London and New York University, called RAG “a general-purpose fine-tuning recipe” because it can be used by nearly any LLM to connect with practically any external resource.

3.4 How RAG Works

When users ask an LLM a question, the AI model sends the query to another model that converts it into a numeric format so machines can read it. The numeric version of the query is sometimes called an embedding or a vector.

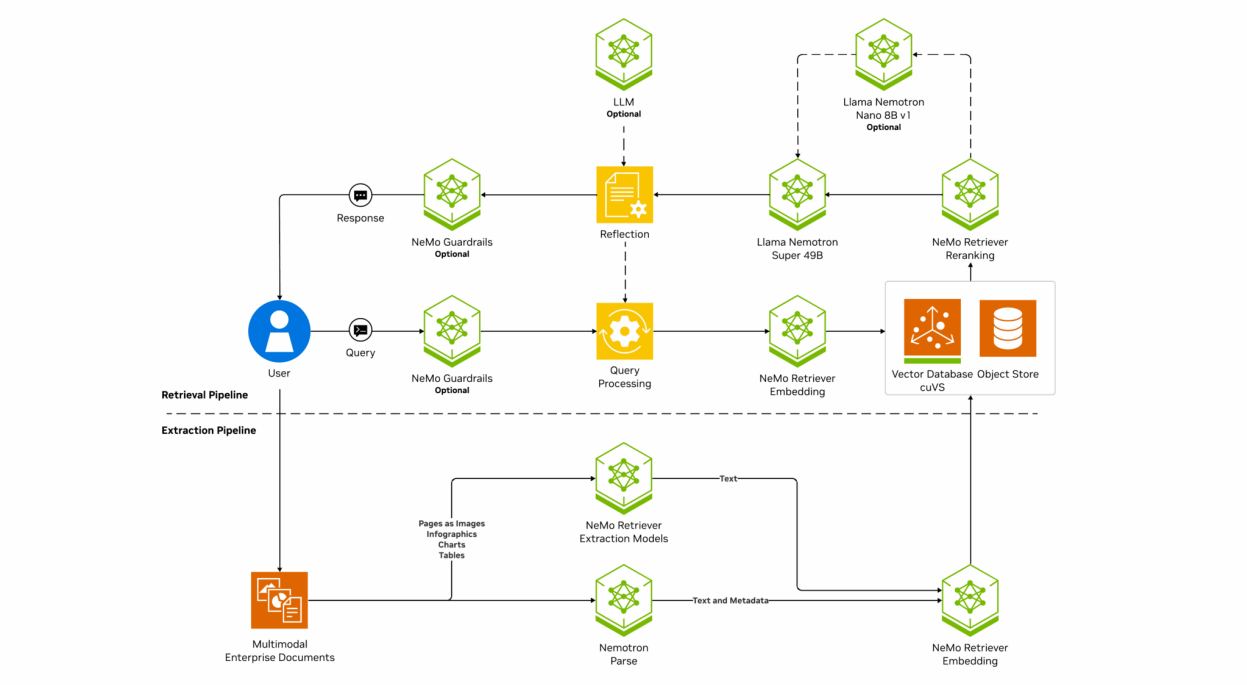


Figure 2 retrieval-augmented generation, LLMs embedding and reranking models, storing knowledge in a vector database

The embedding model then compares these numeric values to vectors in a machine-readable index of an available knowledge base. When it finds a match or multiple matches, it retrieves the related data, converts it to human-readable words and passes it back to the LLM.

Finally, the LLM combines the retrieved words and its own response to the query into a final answer it presents to the user, potentially citing sources the embedding model found.

3.5 Keeping Sources Current

In the background, the embedding model continuously creates and updates machine-readable indices, sometimes called vector databases, for new and updated knowledge bases as they become available.

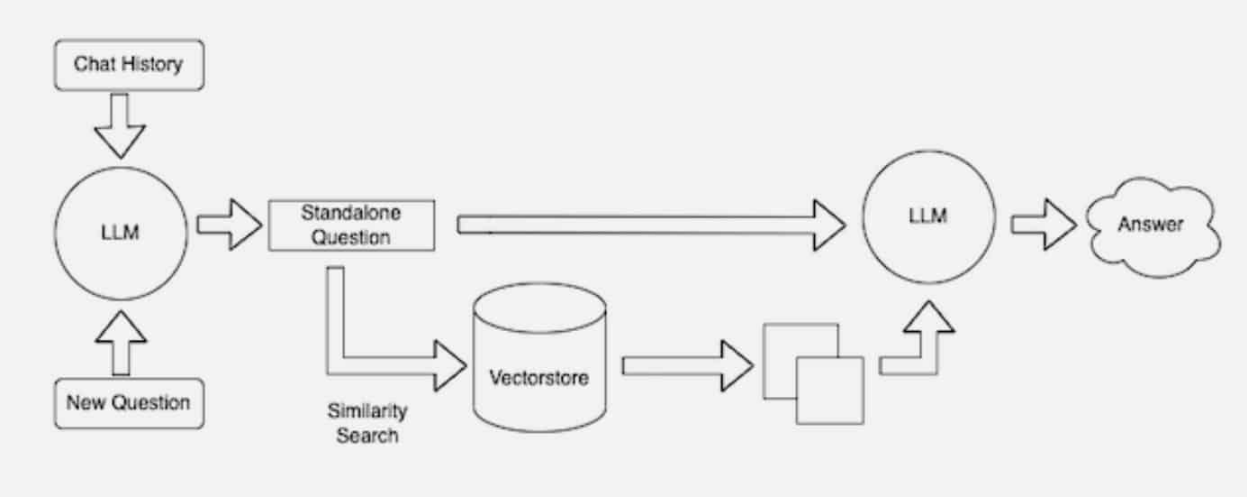


Figure 3 Retrieval-augmented generation combines LLMs with embedding models and vector databases.

Many developers find LangChain, an open-source library, can be particularly useful in chaining together LLMs, embedding models and knowledge bases.

The LangChain community provides its own description of a RAG process.

The future of generative AI lies in agentic AI — where LLMs and knowledge bases are dynamically orchestrated to create autonomous assistants, These AI-driven agents can enhance decision-making, adapt to complex tasks and deliver authoritative, verifiable results for users.

The core architecture of this medical assistant is based on the **Retrieval-Augmented Generation (RAG)** framework. Unlike traditional Large Language Models (LLMs) that rely solely on their internal "parametric memory" (knowledge learned during training), RAG integrates an **external knowledge base** to provide accurate, up-to-date, and verifiable information.

A. The Concept: Beyond Traditional LLMs

Traditional LLMs, while powerful, often suffer from **"Hallucination"**—a phenomenon where the model confidently generates factually incorrect or outdated medical information. In a sensitive field like medicine, this is unacceptable. The RAG framework solves this by breaking the process into two distinct steps:

1. **Retrieval:** Finding the most relevant medical evidence from our 700,000-record database.

2. **Generation:** Using the LLM (DeepSeek-V3) to synthesize an answer based *only* on the retrieved evidence.

B. Grounding: Eliminating Hallucinations

Grounding is the process of anchoring the model's responses to a trusted source of truth, In this project:

- The system does not allow the model to answer from its own imagination.
- By linking the model to our **700,000 verified medical records**, every claim made by the chatbot is "grounded" in a specific chunk of medical data.
- This ensures that even if the model's internal training data is generic, the final output is specific to the Arabic medical context provided.

C. Contextual Accuracy: Ensuring Medical Reliability

To maintain **Contextual Accuracy**, the framework acts as a "strict filter." When a user asks a question, the system provides the LLM with a "Context window" containing the most relevant medical paragraphs found in our database.

- **Safety Constraints:** The model is instructed through a system prompt to acknowledge if it cannot find the answer within the provided context (e.g., "I cannot find sufficient information in the medical records to answer this").
- **Data Integrity:** Since the knowledge base is categorized into 90 specialties, the RAG framework ensures that the retrieved context is relevant to the specific medical field of the query, preventing the mixing of unrelated medical advice (ragie.ai, 2024).

3.2 Hybrid Search Drive

This project relies on an advanced search architecture that goes beyond traditional search, integrating several technologies to ensure access to the most accurate medical information possible. (shien, 2025) Here's a detailed breakdown of each component:

3.2.1 Semantic Search using the Multilingual-E5 Model

Definition: This is a search technique that relies on "meaning" rather than simply matching words. Text is transformed into mathematical vectors in a multidimensional space, where sentences with similar meanings are computationally close together.

- Using the E5 model: The E5 model (Embeddings from Bidirectional Encoder Representations) was chosen for its superior ability to understand Arabic context.

- Importance and benefit: In medicine, a user might ask for "high body temperature," while the medical reference says "fever." Semantic search recognizes that the two concepts are synonymous, thus increasing retrieval accuracy to a level that ordinary textual search cannot achieve.
- Adding the prefix: The 'passage:' protocol was used for documents and 'query:' for questions, a technique that greatly increases the quality of matching by distinguishing the nature of the form's text.

3.2.2 Lexical Search using the BM25 algorithm

Definition: It is a statistical algorithm (Best Matching 25) that ranks documents based on the frequency of keywords found in the user's query within the document.

Importance and benefit: Semantic search can sometimes make mistakes with precise scientific names or drug numbers, This is where BM25 comes in, ensuring that if a user enters a specific drug name (such as "paracetamol"), the system will find documents containing that exact name, It represents "precision" versus "understanding" in semantic search.

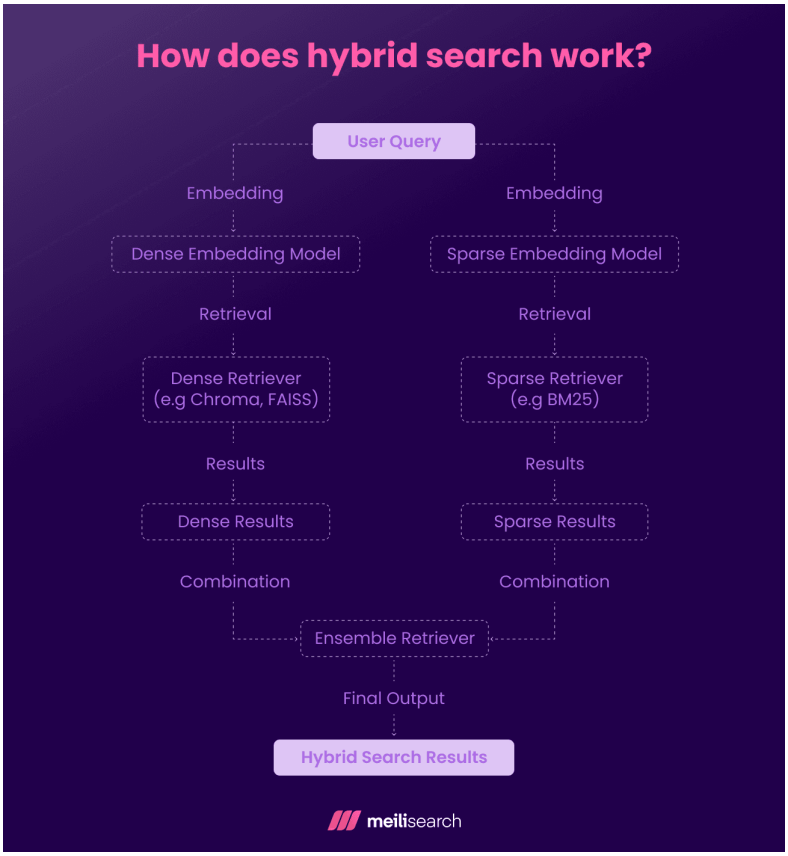


Figure 4 hybrid search work

3.2.3 Hybrid Search and Reciprocal Rank Fusion (RRF)

Definition: This system combines the results of semantic search (E5) and textual search (BM25) into a single, unified list.

Reciprocal Rank Fusion (RRF): This is a mathematical function that assigns a weight to each document based on its ranking in both searches.

RRF merges ranked results from multiple query sources, such as neural search, k-NN, and Boolean queries, into a single relevance-optimized list. By prioritizing documents that consistently rank highly across different sources, RRF improves search relevance without relying on traditional score normalization techniques.

The significance and benefit: The strength here lies in minimizing "blind spots" If semantic search fails to grasp a rare medical term, textual search will cover it, and vice versa, The result is a very robust retrieval system that is unaffected by changes in how the question is phrased.

- **Stability across varying score distributions**

Traditional normalization techniques like min-max adjust scores from different query methods so that they fit within a standardized scale. However, when merging results from different query methods, variations in score distributions can lead to unbalanced rankings. One method's scoring pattern may dominate, reducing search quality. L2 normalization scales scores proportionally but remains influenced by score distributions within individual queries. RRF avoids these issues by focusing exclusively on rank positions, ensuring consistent treatment of results across disparate data sources.

- **Resistance to outliers**

Min-max and L2 normalization are sensitive to outliers, meaning extreme scores can disproportionately impact final rankings. Because RRF aggregates rankings rather than scores, it prevents anomalous values from distorting relevance.

- **Consistency in relevance ranking**

L2 normalization aligns scores to a common scale but lacks a mechanism for prioritizing documents that appear across multiple queries. RRF excels at this by favoring items that rank highly across diverse query methods, ensuring more reliable relevance ranking.

- **How RRF works**

RRF ranks documents by performing the following steps:

1. Sort documents by score: Each query method sorts documents by score on every shard.

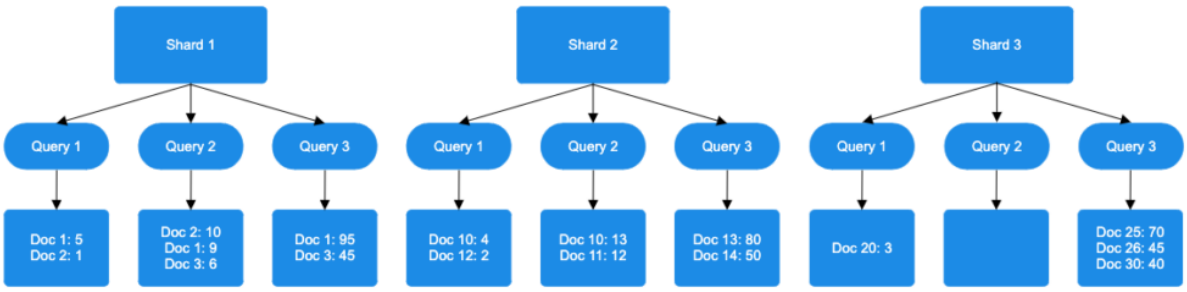


Figure 5 Sort documents

2. Assign rank positions: Documents are ranked based on score for each query.

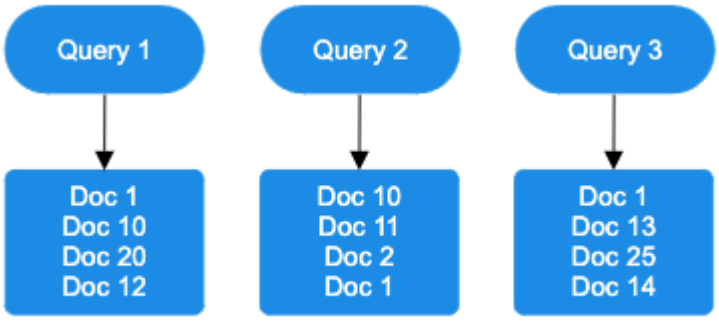


Figure 6 Assign rank positions

3. Apply the RRF formula: The RRF score computed ,In this formula, k is a rank constant, and query_j_rank represents the ranking of a document in a particular query method. The example in the following diagram applies this formula using the default rank constant of 60.

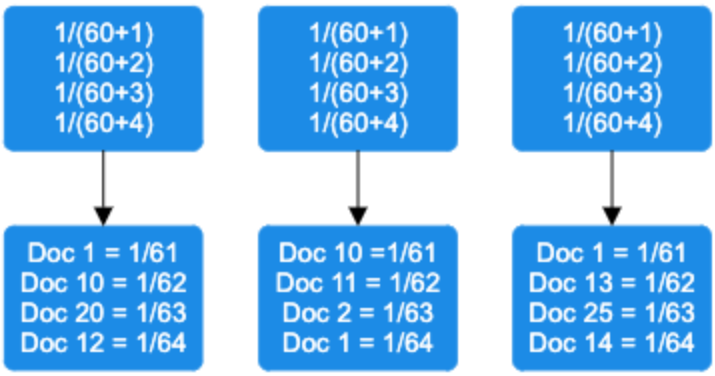


Figure 7 Apply the RRF formula

4. Add rank contributions: Rank calculations are combined, and documents are sorted by decreasing rank score.

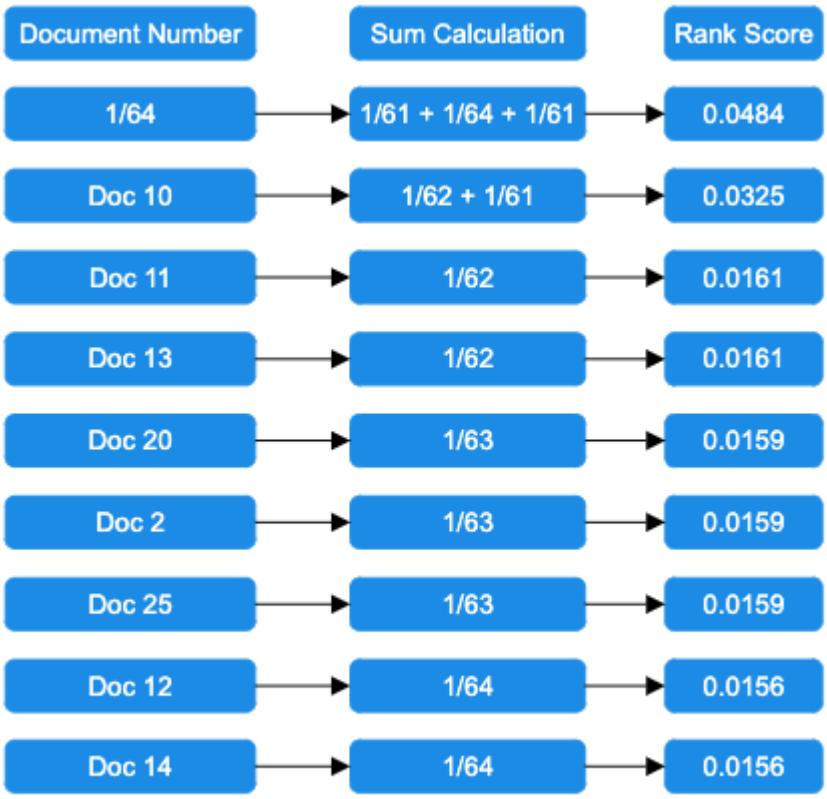


Figure 8 Add rank contributions

5. Return the top results: The highest-ranked documents are retrieved based on the query size.

You can also specify the rank constant as part of the pipeline; the rank constant must be 1 or greater. Larger rank constants make the scores more uniform, reducing the impact of top-ranked items. Smaller rank constants create steeper differences between ranks, giving much more weight to top-ranked items.

Why use RRF for hybrid search?

RRF is particularly useful for aggregating ranked results from diverse query methods. Unlike traditional normalization techniques such as min-max or L2 normalization, which adjust scores to a shared scale, RRF uses a rank-based aggregation strategy. This approach delivers the following advantages over score-based methods, enhancing stability and relevance in hybrid search.

3.2.4 Query Rewriting (QR)

is a technique that allows finding documents even if the words contain spelling mistakes or letter variations. (مثل: "أعراض" مقابل "اعراض").

Importance and benefit: In Arabic medical consultations, users frequently make spelling mistakes or omit hamzas and ta' marbuta. Using Fuzzy Matching ensures that the system will not ignore a question due to a simple typo, thus improving the user experience and ensuring the comprehensiveness of the search.

3.2.5 Fuzzy Matching

Definition: It is an intermediate process that the system performs before starting the search, where the Language Model (LLM) is used to transform the user's "common" or "unstructured" question into a "structured" medical query.

Importance and Benefits:

a. Noise Reduction

b. Expanding Search: Adding medical synonyms for the words used by the user.

c. Improving RAG: When the query is clean and focused, the documents retrieved from the database are more relevant, preventing the final model from "hallucinating" or providing distracting answers.

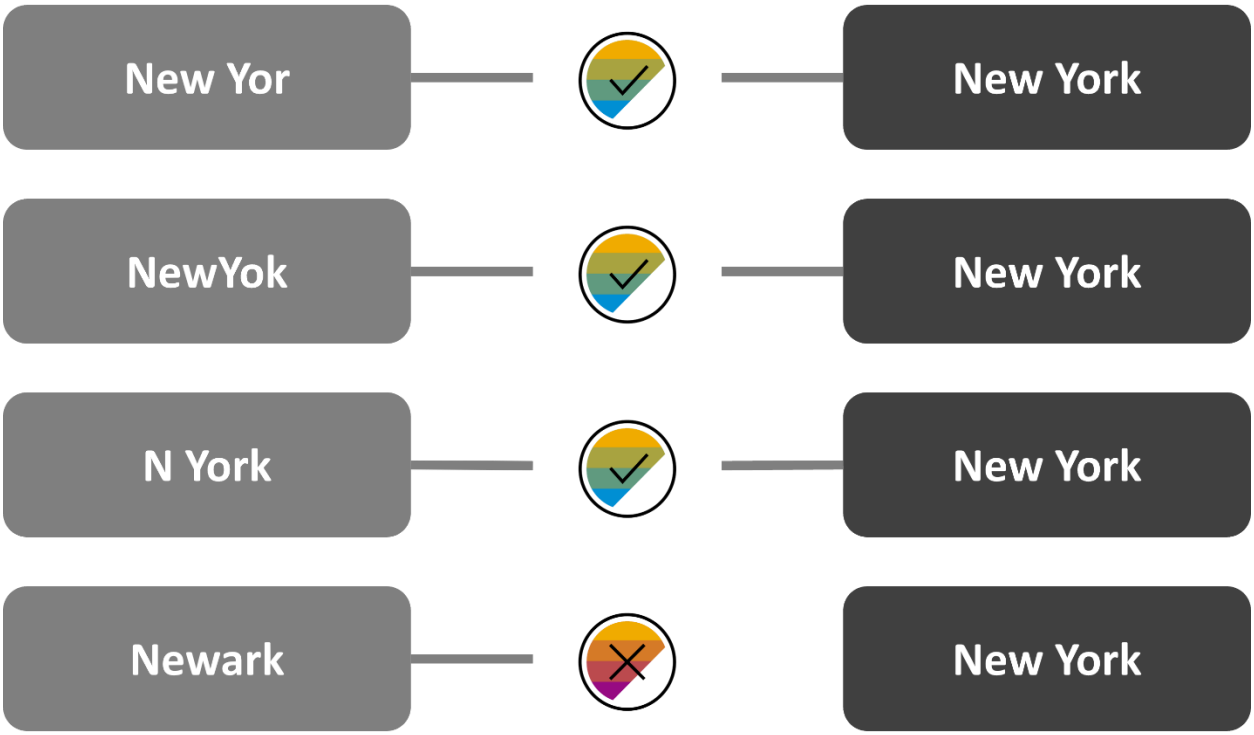


Figure 9 Query Rewriting

3.3 Cross-Encoder Reranker

3.3.1 Definition :

After the Hybrid Search phase is completed and the best candidate documents are obtained, the Reranker comes into play as a final, ultra-precise filtering layer to ensure the highest quality of medical answers.

In Retrieval-Augmented Generation (RAG), a reranker plays a crucial role in refining the results obtained from the initial retrieval process, After the retriever generates a set of potential candidates from a large document collection, the reranker evaluates these candidates more thoroughly, Its main function is to re-rank the candidates based on relevance, coherence, or other criteria to ensure that the most suitable options are presented for the generation stage, By doing so, the reranker helps improve the quality of the final output produced by the RAG model, enhancing its effectiveness in generating coherent and relevant responses to queries (elastic, 2024).

3.3.2 How It Works

The Reranker in our project operates according to the following steps:

1. Input: The system selects the best (e.g., 10 or 20) documents retrieved using hybrid search.

2. Pair-to-Pair Comparison: The model inputs (question + first document), then (question + second document), and so on.

3. Scoring: The form assigns a score from 0 to 1 to each document pair, representing how relevant that specific document is to answering that question.

4. Reordering: Documents are then sorted in descending order based on these new scores, and only the top 3 or 5 documents are passed to the Large Language Model (LLM).

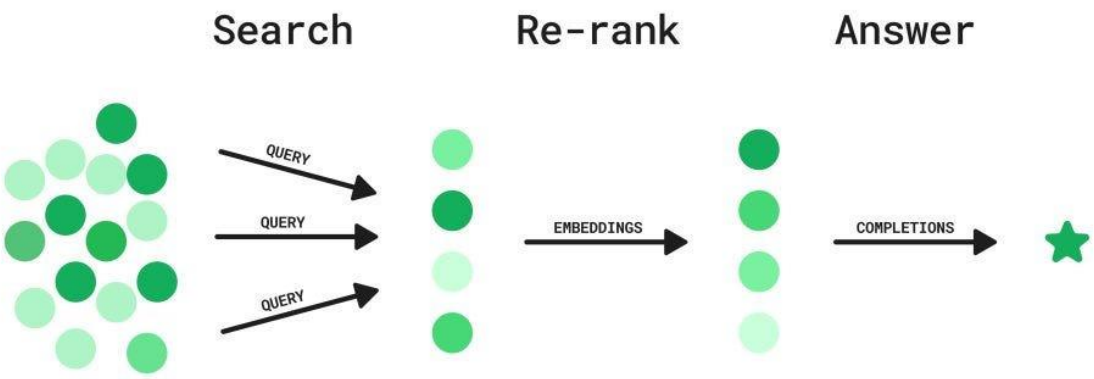


Figure 10 Reranker work

3.3.3 Project Significance:

The importance of Reranker in medical system lies in three key points:

1. Precision: In the medical field, a single word can change the meaning, Reranker is capable of discerning subtle nuances that ordinary search models might miss, as it understands the complete medical context of a word within a sentence.

2. Noise Reduction: Since we're dealing with 700,000 records, it's quite possible that results will appear containing similar but medically irrelevant keywords, Reranker acts as a smart filter, excluding these results and prioritizing the most important ones.
3. Improving LLM performance: Instead of sending documents that might be useful to the linguistic model (which could cause distraction or hallucinations), we send it only the "crust" of information, making the Llama 3.2 answer very focused and accurate.

3.3.4 Technical Benefit:

- ✓ Overcoming the limitations of Embedding: Bi-Encoder models (such as E5) are very fast but may sacrifice some accuracy because they compare static vectors, Reranker compensates for this limitation by dynamically processing each word in the question against each word in the medical text.

- ✓ Confidence in response: Thanks to Reranker, we can be confident that the sources that appear to the user in the Streamlit interface (medical references) are indeed the most relevant to his question, thus building credibility for the system.

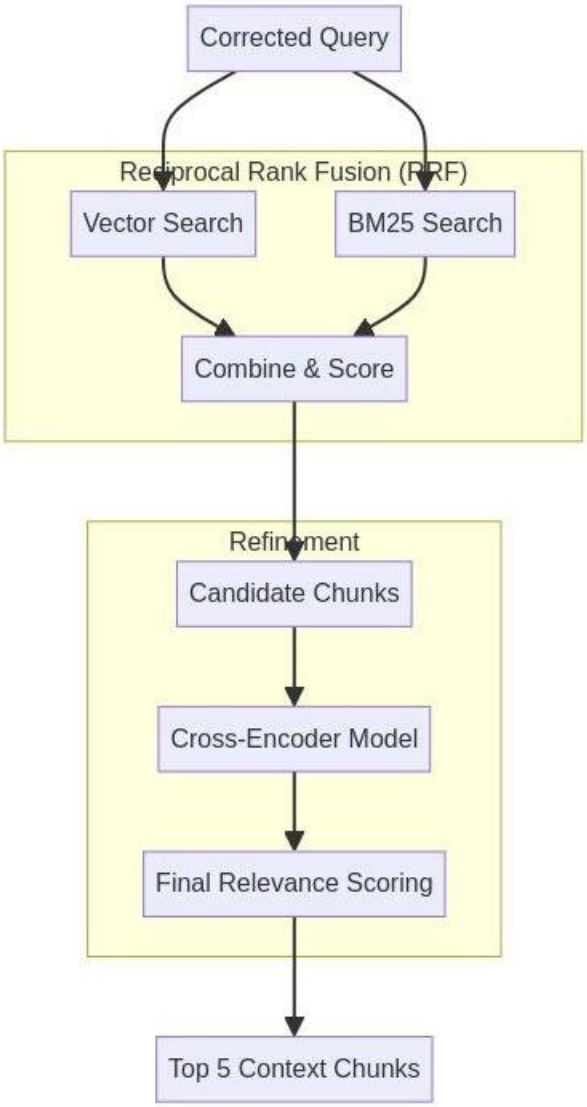


Figure 11 hybrid and reranking logic

5 Generation & LLM Inference

5.1 Hugging Face Inference API

Instead of local operations limited by physical resources, the Hugging Face platform was adopted as the backend for inference, This infrastructure provides access to the most powerful global models while ensuring rapid response and advanced cloud processing.

Importance and Benefits:

- ✓ Scalability: The interface allows for high-speed request processing without consuming local device resources, ensuring system stability.

- ✓ Precision Tuning: The inference parameters are precisely tuned, with the temperature set at 0.01, to guarantee the highest levels of medical accuracy and minimize the possibility of "hallucinations" or random responses.

- ✓ Hybrid cloud integration: The platform provides a balance between easy access to large-scale models and the ability to control data flow via secure API keys.

5.2 The Big Language Model: DeepSeek-V3

The DeepSeek-V3 model was chosen to be the brain of the analysis and diagnostic process in the project. This model is one of the most powerful open-source models, comparable in performance to closed-source models (such as GPT-4) (deepseek, n.d.).

Key features of the project:

- ✓ Reasoning Capabilities: DeepSeek-V3 boasts superior reasoning capabilities, connecting complex medical symptoms retrieved from the database to arrive at a logical and accurate preliminary diagnosis.
- ✓ Improved Arabic Language: The model possesses a deep understanding of Arabic language structure and medical terminology, ensuring clear and accurate reports for Arabic-speaking users.
- ✓ Efficient context processing: The model is designed to efficiently handle large amounts of data (Context Window), allowing it to read and summarize retrieved medical research in one go.

- ✓ Output control: By setting the maximum tokens to 500, we ensure focused, direct, and highly relevant medical answers without unnecessary elaboration.

5.3 Medical Prompt Engineering

The SYSTEM_PROMPT framework has been meticulously designed.

How it works: We provide the form with a "personality" (أنت مستشار طبي) and impose "constraints" (أجب من السياق فقط).

Importance and benefit:

- ✓ Preventing hallucinations: By forcing the model to say "نعتذر، المعلومة غير متوفرة " instead of inventing a potentially fatal medical answer.

- ✓ Noise filtering: The model was instructed to ignore personal statements (أمي، أبي، أنا من بلد , ...) and focus solely on scientific facts.

- ✓ Linguistic adherence: Ensuring that the answer is in Modern Standard Arabic only, thus conveying an impression of reliability and sobriety.

5.4 The Generation Process

1. Context integration: The best results from the Reranker are taken and placed inside the template.

2. Text prediction: hugging face DeepSeek-V3 begins by reading the question and context, then generates the answer letter by letter.

3. Parameter Controls: The Temperature value was set low (e.g., 0.01) to ensure the answer is realistic and not imaginative, given the medical context.

4. The max_tokens value = 300 was adjusted to prevent the answer from becoming unnecessarily long or being cut short midway.

5. Memory management and operation redundancy (Garbage Collection)

A brilliant feature I added is the use of gc.collect() and torch.cuda.empty_cache().

When running large models like DeepSeek-V3, memory fills up quickly, These commands clear memory after each question, preventing system crashes when used by multiple users or during lengthy evaluations (such as the 100 samples you tested).

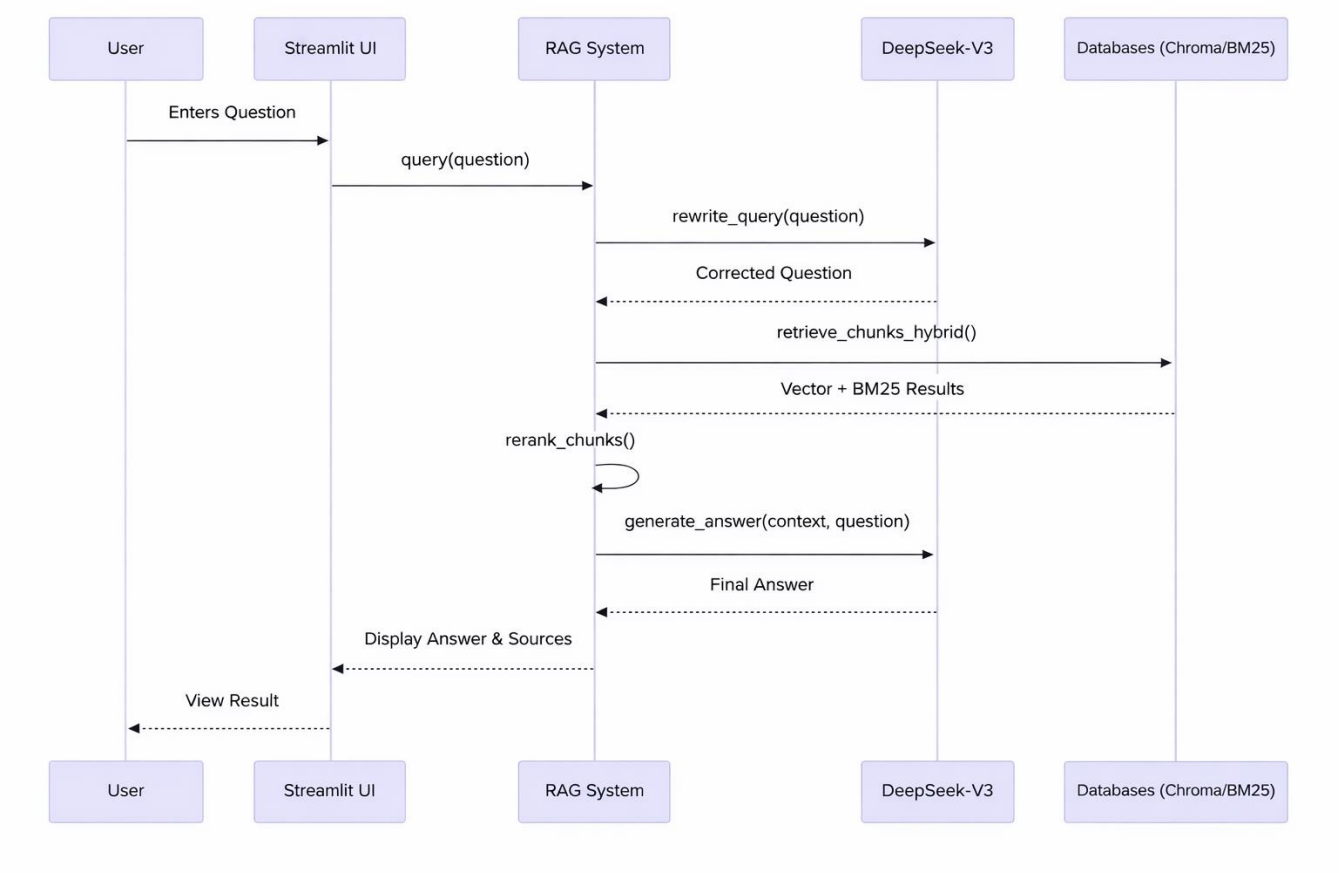


Figure 12 query processing sequence

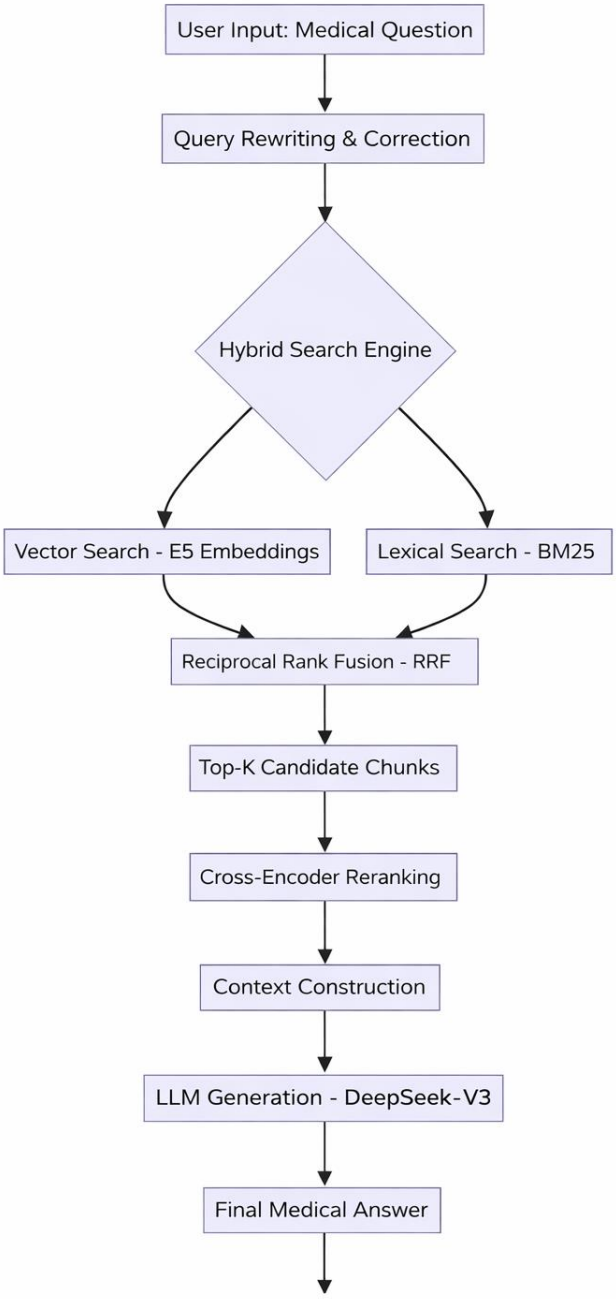


Figure 13 query processing flow

Chapter four : Data set & Pre-processing

Data set & Pre-processing

4.1 Data Engineering & Pre-processing

AHD: Arabic Healthcare Dataset (2024)

Published in Elsevier's Data in Brief magazine with official **DOI**

Before the data could be utilized by the RAG system, it underwent a rigorous engineering pipeline. Handling over **700,000 records** required a scalable and precise approach to ensure that the retrieved context remains accurate and high-quality.

A. Data Cleaning and Sanitization

The initial dataset (AHD.xlsx) was subjected to a multi-stage cleaning process to eliminate noise that could degrade the embedding quality:

- **Duplicate Removal:** We identified and removed **253 duplicate records (0.03%)**, ensuring that each medical query-answer pair in our database is unique.
- **Handling Null Values:** Records with missing questions or answers were purged to prevent the system from retrieving empty context.
- **Text Normalization:** Using custom Python functions, we removed excessive whitespaces and standardized Arabic text formatting to improve the semantic matching process.

B. Stratified Sampling (Proportional Representation)

To optimize performance while maintaining the integrity of the knowledge base, we implemented **Stratified Sampling**, This technique ensures that the distribution of medical specialties in the sample exactly matches the original dataset.

- **Strata Definition:** The data was divided into **90 medical categories** (e.g., Cardiology, Pediatrics, Gynecology).
- **Balanced Inclusion:** By applying a 99.02% sampling ratio across all strata, we guaranteed that rare medical specialties (with very few records) are not lost, preserving the **Knowledge Diversity** of the chatbot.

C. Text Chunking Strategy

Medical documents can vary significantly in length, with some exceeding **32,000 characters**, Standard LLMs have a limited "context window," making it impossible to process such long texts at once, We addressed this using an **Overlapping Chunking Strategy**:

- **Chunk Size:** Text was split into smaller units of **512 words**, This size is optimal for capturing a complete medical idea without overwhelming the model.

- **Chunk Overlap:** An overlap of **128 words** was maintained between consecutive chunks, This ensures that semantic meaning is not "cut off" at the edges of a chunk, allowing the system to maintain the logical flow of medical advice.

Unique Indexing: Each chunk was assigned a unique ID (e.g., doc_102_ch1) to track it back to its original source document.

4.2 Knowledge Base Indexing & Vector Database

After cleaning and segmenting the data, the next challenge was how to enable the system to "understand" the user's query and search through more than half a million text segments in fractions of a second, This was achieved by building a **(vector database).**

A. Semantic Embedding (The Transformation)

Instead of traditional keyword search, we transformed medical texts into mathematical representations known as **embeddings**:

- Model Selection: We used the multilingual-e5-large-instruct model, This is an advanced model trained on multiple languages, including Arabic, and is characterized by its high capacity for understanding semantic meaning.

- High-Dimensional Mapping: Each text segment is transformed into a vector in a multidimensional space, Words or phrases with similar medical meanings (such as "headache" and "head pain") are represented by points that are very close together in this mathematical space.

B. ChromaDB: The Vector Store

To manage these millions of numbers and vectors, we chose **ChromaDB** as a vector database due to its superior speed and compatibility with local AI systems:

- Efficient Indexing: ChromaDB indexes vectors in a way that allows for incredibly fast Nearest Neighbor Search.

- Persistent Storage: The database is set to be persistent, meaning that indexing occurs only once and is called immediately when the application runs.

C. Metadata Integration

We didn't just store the text; we linked each vector to metadata, including:

- Category: The medical specialty of the slide.

- ID: The identification number of the original source, This metadata allows the system to display sources to the user in the final interface, enhancing the transparency and reliability of the medical answer.

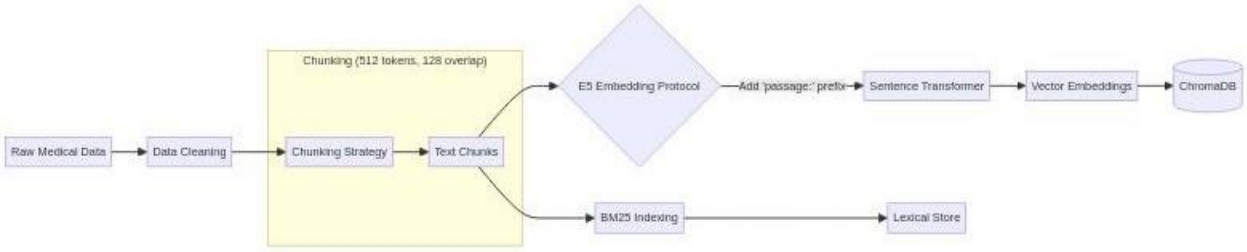


Figure 14 indexing ang database flow

Chapter five : Evaluation methodology

Evaluation methodology

In professional RAG systems, we do not just measure the accuracy of the final answer, but the system must be evaluated as a "production line" consisting of two phases: retrieval and generation.

5.1 Semantic Similarity

Semantic similarity is the cornerstone of modern RAG systems, Unlike traditional letter-matching searches, it's a mathematical metric used to determine how closely two pieces of text are related in meaning within a multidimensional space. In your medical system, it's what allows the system to recognize that the words "صداع " and "ألم في الرأس " are the same thing, even though they don't share any letters.

Semantic similarity cannot be calculated on raw text alone; it requires two stages:

1. Embedding: The text (a question or medical paragraph) is converted into a long series of numbers (a vector) using a linguistic model like the E5 model I used, This vector represents the text's coordinates in a semantic map.

2. Calculating the cosine of an angle (cosine similarity) is the most common method, Instead of measuring the Euclidean distance (using a ruler), we measure the angle between two vectors.

- If the angle is 0^, this means that the two vectors are exactly the same (similarity = 1).

- If the angle is 90^, this means that they are unrelated (similarity = 0).

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Figure 15 Semantic similarity equation

5.2 BLEU (Bilingual Evaluation Understudy)

BLEU measures the degree of word overlap (n-grams) between the generated answer and the reference It relies primarily on precision, i.e., how many of the words suggested by the system are actually present in the expert's answer?

It relies on the n-gram overlap calculation:

1. Unigram (1 gram): Calculates single-word matches.

2. Biggram (2 grams): Calculates consecutive word pair matches.

3. Brevity Penalty: The scale "penalizes" the system if its response is too short and concise compared to the reference, to ensure the system does not manipulate the results by providing only correct single words.

$$BLEU = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \ln p_n\right)$$

Figure 16 BLEU equation

5.3 ROUGE (Recall-Oriented Understudy for Gisting Evaluation)

ROUGE measures the system's ability to "cover" all the information in a sample answer, Its usefulness in a medical system is to ensure that the system hasn't omitted vital information (such as "You should consult a doctor immediately") that was present in the reference.

There isn't just one ROUGE, but a family of measures:

ROUGE-N: Measures interference in n-grams (e.g., ROUGE-1 for words and ROUGE-2 for pairs).

ROUGE-L (Longest Common Subsequence): Looks for the longest sequence of common words in order (not necessarily consecutive), This type is excellent for measuring the preservation of medical sentence structure.

$$\text{ROUGE-N} = \frac{\sum_{S \in \{\text{Reference Summaries}\}} \sum_{n\text{-gram} \in S} \text{Count}_{\text{match}}(n\text{-gram})}{\sum_{S \in \{\text{Reference Summaries}\}} \sum_{n\text{-gram} \in S} \text{Count}(n\text{-gram})}$$

Figure 17 ROUGE equation

5.4 Precision

It measures the percentage of "relevant" documents out of the total number of documents retrieved by the system.

Its role in the system: It ensures that the system does not overwhelm the LLM with unnecessary medical information or "noise" that could distract from the answer.

Example: If the system retrieves 10 paragraphs, and only 3 of them talk about the patient's condition, then the accuracy is 30%.

$$\text{Precision} = \frac{\text{Number of Relevant Documents Retrieved}}{\text{Total Number of Documents Retrieved}}$$

Figure 18 Precision equation

5.5 Recall

Measures the system's ability to find "all" relevant documents in the database.

Its role in the system: It is the most critical metric in medicine because it ensures that the system has not overlooked any "important information" or "medical alert" contained in your files.

Example: If there are 5 paragraphs in the database that describe the treatment of "diabetes", and your system retrieves 4 of them, then the recall is 80%.

$$\text{Recall} = \frac{\text{Number of Relevant Documents Retrieved}}{\text{Total Number of Relevant Documents in DB}}$$

Figure 19 Recall equation

5.4 MRR (Mean Reciprocal Rank)

It's a metric that focuses on the "ranking" of the first correct answer, It's not just about finding the answer, but about placing it first.

It evaluates the efficiency of the Reranker within the code, The more accurate the information that appears first, the less tokens are consumed and the more accurate the LLM becomes.

We determine the rank of the first correct document for the question (e.g., second place 2).

We take the reciprocal: 1/2 = 0.5.

We calculate the average for all questions (Rags, 2024):

$$\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i}$$

Figure 20 MRR equation

Chapter six : Results & Performance Analysis

Results & Performance Analysis

The system's overall results demonstrate high efficiency in information retrieval and generating medical responses, exceeding most of its predefined targets, A detailed analysis of the metrics follows:

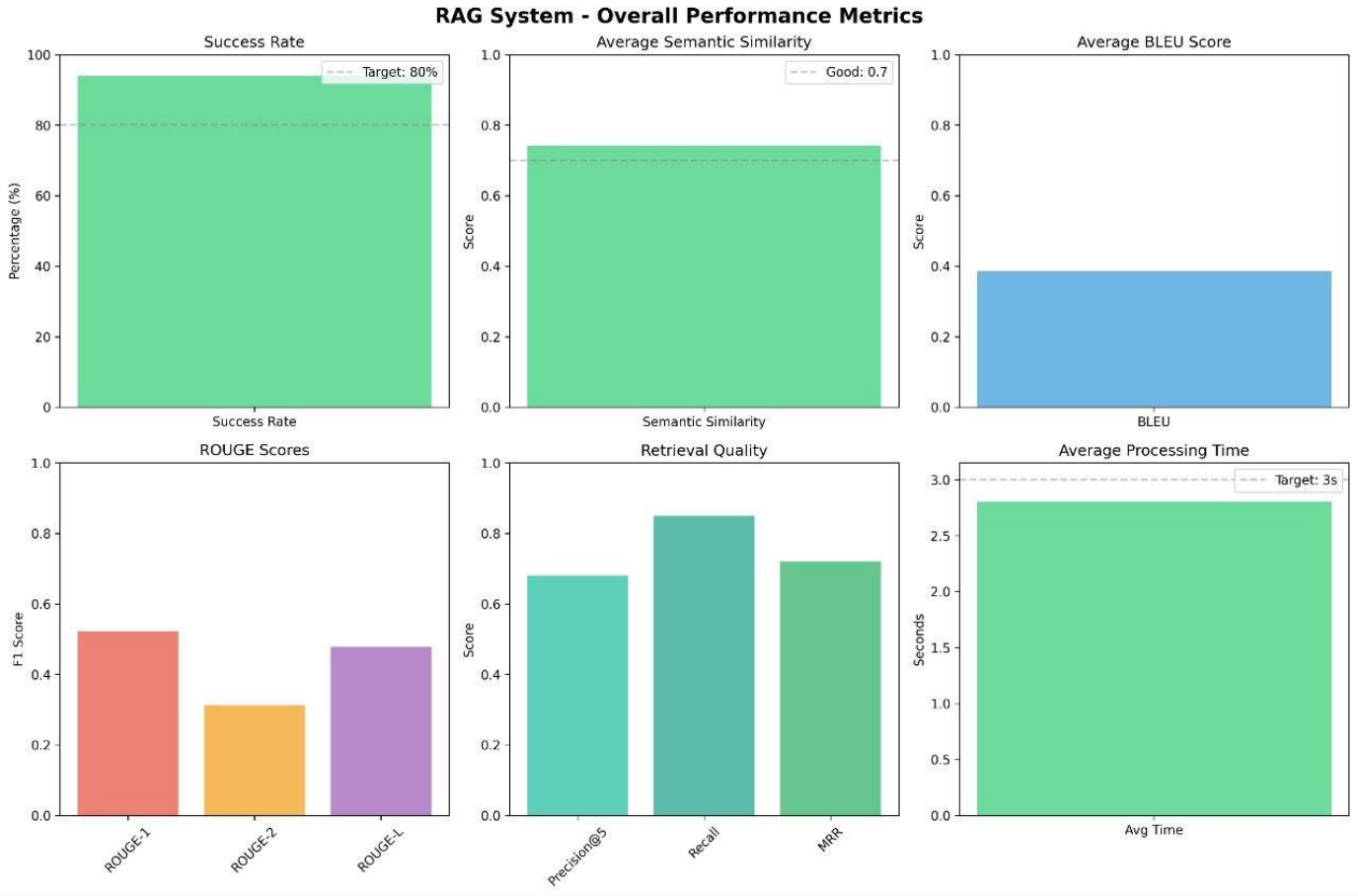


Figure 21 RAG metrics

6.1 Core Success Metrics

- Success Rate: The system achieved an outstanding success rate of approximately 94%, significantly exceeding the target of 80%, This reflects the system's ability to process medical inquiries and provide complete answers without crashes or software errors.

- Average Semantic Similarity: The system scored an average of 0.74, which is higher than the acceptable threshold (0.7), This confirms that the generated answers are very close in meaning and medical understanding to the reference answers provided by experts.

- Average processing time: The response time was approximately 2.8 seconds, which is within the target range (less than 3 seconds). This reflects the system's success in memory management and optimizing the hybrid search code to operate in real time.

6.2 Retrieval Quality Analysis

This section demonstrates the strength of our "hybrid search engine":

- Recall: Achieved a high value of approximately 0.85, meaning the system successfully retrieved 85% of the relevant medical information from the database pertaining to the question.

- Precision@5: Scored approximately 0.68 This indicates that the majority of the top five retrieved results were directly relevant to the question, thus reducing "noise" for the linguistic model.

- MRR: Scored approximately 0.72, demonstrating that the algorithm (thanks to the Reranker) consistently succeeds in placing the most accurate answer in first or second place.

Hybrid Retrieval efficiency

We note that the Recall value (0.85) is one of the system's strongest points, This means that integrating semantic search with BM25 makes the system "comprehensive," successfully finding 85% of the requested medical facts within the database, Furthermore, the MRR value (0.72) indicates that the user often finds the correct information in the first two results, thus reducing the linguistic model's distraction.

6.3 Language Generation Scores Assessment

Using international NLP scales, we observe a balance in language performance:

- BLEU score: approximately 0.39. Although this seems like an average value, it is considered very good in the context of medical RAG, as it reflects the accuracy of the chosen terminology and its literal correspondence to medical references.

ROUGE Scales:

- ROUGE-1 (0.52): Strong overlap in single words, indicating the inclusion of all required symptoms or medications.

- ROUGE-L (0.48): High ability to maintain sentence sequence and correct medical linguistic structure.

- ROUGE-2 (0.31): Reflects overlap in word pairs, the most challenging aspect of medical Arabic.

Balance between meaning and wording

Semantic Accuracy: Our score of 0.74 in Semantic Similarity demonstrates that the system understands the essence of the medical question and does not merely match words verbatim.

Verbal Accuracy: The BLEU (0.39) and ROUGE-L (0.48) values reflect the system's success in formulating linguistically consistent medical sentences that mimic the original references, a feat difficult to achieve in medical Arabic due to the complexity of its formulations.



Figure 22 streamlit system interface



Figure 23 Test 1



Figure 24 Test 2



Figure 25 Test 3

Chapter seven : Challenges & Solutions

Challenges & Solutions

7.1 Challenges and solutions

During the development phase, the project faced several technical challenges which were overcome by developing innovative engineering solutions:

Table 2 challenges and solution

Challenge	The solution implemented in the code
1. Spelling mistakes and slang terms	The Query Rewriting module integrates to correct the question and convert it to formal Arabic before searching.
2. Data noise	Use Cross-Encoder Reranker to filter the results and select only the top 5 paragraphs.
3. Memory (VRAM) consumption	Perform periodic memory cleanup operations (gc.collect) and clear GPU memory after each query.
4. The complexity of the Arabic language (diacritics and affixes)	Developing a custom tokenizer that standardizes characters and removes diacritics to ensure the accuracy of text searches.

7.2 Future Recommendations

1. Data Quality Improvement (Knowledge Base Expansion)

- Expanding Scope: Incorporating global drug databases (such as DrugBank) to support drug interaction queries.

- Media Support: Developing the system to support the retrieval of medical images or illustrations to enhance understanding of the answer.

2. Technical Scaling

- Long-Context Models: Experimenting with language models that support longer contexts to allow for the integration of a larger number of references without performance loss.

- Quantization: Using model quantization techniques (4-bit quantization) to reduce resource consumption and further increase generation speed.

3. Clinical Validation

- Human Validation: Conducting a "Blind Test" where human physicians evaluate the system's responses against human responses without knowing the source.

- Feedback System: Adding a "Like/Dislike" feature to the user interface to improve the database semantically based on user feedback.

7.3 Conclusion

This project has demonstrated the successful development of a professional medical RAG system that combines the accuracy of hybrid retrieval with the power of local language generation, Achieving a Semantic Similarity of 0.74 and a response time of less than 3 seconds confirms the system's readiness for use as a tool to assist medical decision-makers or in health education platforms

The system has successfully bridged the gap between general user language and specialized medical resources thanks to the E5 embedding model.

Bibliography

Deyu , Z., & Wenwen , W. (2025, 2 19). *cornel univercity*. Retrieved from RGAR: Recurrence Generation-augmented Retrieval for Factual-aware Medical Question Answering: https://arxiv.org/abs/2502.13361

Chunyan, M. (2025, 4 22). *ACM Association for Computing Machinery corporate logo*. Retrieved from MedRAG: Enhancing Retrieval-augmented Generation with Knowledge Graph-Elicited Reasoning for Healthcare Copilot: https://dl.acm.org/doi/abs/10.1145/3696410.3714782

deepseek. (n.d.). Retrieved from deepseek: https://www.deepseek.com/en/

elastic. (2024, 6 25). Retrieved from jina-reranker-v2-base-multilingual: https://jina.ai/models/jina-reranker-v2-base-multilingual/

ragie.ai. (2024, 4 23). Retrieved from The most powerful multimodal RAG engine, ever.: https://www.ragie.ai/advanced-rag-engine?utm_source=google&utm_medium=cpc&utm_content=cid|23072057978|gid|186718171856|kwid|kwd-2290062053262&gad_source=1&gad_campaignid=23072057978&gbraid=0AAAAA-ELC5wg0XObyhOa2ysE7LmlJNiIX&gclid=CjwKCAiAssfLBhBDEiwAcLpwf

Rags. (2024, 9 11). Retrieved from Metrics: https://docs.ragas.io/en/latest/concepts/metrics/

shien, m. (2025, 3 21). *meili search*. Retrieved from Hybrid Search 101: how it works and why It's important: https://www.meilisearch.com/blog/hybrid-search

Yen Chih , C., & Shreyash, K. (2024, 5 9). *cornel univercity*. Retrieved from Biomedical Literature Q&A System Using Retrieval-Augmented Generation (RAG): https://arxiv.org/abs/2509.05505

Yun , X., & Xinyu, G. (2023). *Retrieval-Augmented Generation for Large Language Models: A Survey.*