Case Closed: Medical Question Answering with RAG

Sameed Ahmad-26956

Izma Khan-26926

Abdullah Rehman-27074

Table of Contents

1. Abstract

This report documents our development of a Retrieval-Augmented Generation (RAG) system tailored for medical question answering. We utilized a manually curated dataset of health-related documents and applied semantic retrieval using FAISS with Hugging Face embeddings. FLAN-T5 was employed for generation. Experiments tested chunking strategies, token filters, and semantic search parameters. Evaluation was done using relevance and faithfulness metrics. Our results show that semantic retrieval paired with overlapping chunks and instruction-tuned LLMs like FLAN-T5 significantly enhances answer quality in the medical domain.

2. Introduction

2.1 Problem Statement: Why is Medical QA Important?

Medical research is rapidly expanding, with thousands of new studies and papers published each year. Clinicians, researchers, and patients struggle to find specific, reliable answers in a sea of unstructured documents. Traditional keyword search tools often fall short, missing contextual links or returning irrelevant passages. For example, a user asking "Can lifestyle changes reverse diabetes?" may receive fragmented or non-specific responses from standard tools. A more intelligent system is needed to understand context, retrieve relevant evidence, and provide grounded, human-like answers.

2.2 Challenges in Medical QA

- **Data Complexity**: Medical literature includes technical jargon, abbreviations, and statistical detail that require context-aware parsing.
- **Document Size**: PDFs and reports often exceed 10,000 words, which surpasses LLM context limits.
- **Precision & Risk**: Incorrect or hallucinated medical facts can mislead users, emphasizing the need for trustworthy systems.
- **Semantic Ambiguity**: Similar concepts (e.g., "glucose intolerance" vs. "prediabetes") require deeper semantic understanding.

2.3 Why RAG?

Retrieval-Augmented Generation (RAG) is a hybrid architecture that first retrieves relevant document chunks using a semantic search method (e.g., FAISS), and then feeds those chunks to a generative model (e.g., FLAN-T5 or GPT-3) to construct the final answer. Unlike closed-book models, RAG systems can reference evolving documents without retraining. This makes them ideal for domains like medicine, where new guidelines emerge regularly. RAG also improves explainability by surfacing the source documents used to generate each answer.

2.4 Scope of Experiments

We designed our RAG pipeline to operate over a curated set of 20+ PDFs focusing on diabetes. We tested three chunking strategies with varied overlap and embedded them using Sentence Transformers. We then built vector stores (FAISS) and ran queries on different LLMs using langchain. We evaluated combinations based on: retrieval timing, answer latency, and scores for relevance and faithfulness derived from LLM evaluation

prompts. Our goal was to compare retrieval strategies and optimize chunking methods for medical QA.

This report evaluates the following RAG configurations:

- **Exp 1** : FAISS retrieval
  - Tests semantic search on standard transformer embeddings
  - Chunk size: 500 tokens / 50 overlap
- **Exp 2**: FAISS
  - Tests vectorstore backend differences with same embeddings
  - Chunk size: 1000 tokens / 100 overlap
- **Exp 3**: FAISS retrieval + flan-t5-base
  - Compares open-source LLM against OpenAI performance
  - Chunk size: 1500 tokens / 200 overlap
- **Exp 4**: Chunk variation benchmarking (500/50 vs. 1000/100 vs. 1500/200)
  - Evaluates effect of granularity on retrieval and answer quality
- **Exp 5**: MiniLM vs. MPNet embeddings comparison
  - Tests embedding model effect on semantic search precision

Regions of Concern:

- How does chunk size & overlap impact retrieval effectiveness?
- Does Flan help FAISS in speed or accuracy?
- Which LLM generates more faithful answers: OpenAI vs. HuggingFace?
- Do larger context windows improve answer relevance or introduce noise?
- What trade-offs exist between model speed, accuracy, and cost?

Platform Details:

- All experiments were run on VSCode for embedding and retrieval steps
- Local environments (Linux/Windows) were used for PDF handling and text preprocessing

2.5 Libraries used include:

Retrieval & Ranking

- `rank_bm25.BM25Okapi` – for **keyword-based retrieval** using BM25 ranking
- `faiss` – for **dense vector-based semantic retrieval**
- `langchain.vectorstores.FAISS`, – vector storage integrations in LangChain

Embeddings & Language Models

- `sentence_transformers.SentenceTransformer` – for **semantic embeddings** (e.g., MiniLM, MPNet)
- `transformers.AutoModel`, `AutoTokenizer` – to load **custom transformer models**
- `langchain.embeddings.HuggingFaceEmbeddings` – wrapper for embedding models in LangChain

Evaluation & Metrics

- `sklearn.metrics.pairwise.cosine_similarity` – to measure **semantic similarity** between embeddings
- `sklearn.metrics.accuracy_score` – used optionally for classification tasks

Data Loading & PDF Extraction

- `pdfplumber` – for **PDF text extraction**
- `requests` – to **download PDFs** from public URLs
- `tempfile` – for temporary directory creation

Data Handling & Utilities

- `pandas` – for tabular data handling (`DataFrame`, `.csv` export)
- `numpy` – for matrix operations (e.g., similarity computation, vector norms)
- `collections.defaultdict` – useful for storing multi-key retrieval results

Visualization

- `matplotlib.pyplot` – to create static visual plots
- `seaborn` – for styled performance charts (faithfulness, relevance, latency)

Document Generation

- `docx.Document`, `docx.shared.Inches` – to **generate and export Word reports**

---

3. Dataset Description & Preparation

3.1 Dataset Overview

Our RAG-based question-answering system was developed using a custom-compiled corpus of open-access medical research papers, reports, and guidelines with a specific focus on diabetes. We curated a list of 20 high-quality PDF documents from reliable sources such as:

- **PubMed Central**
- **BiomedCentral Journals**
- **World Health Organization (WHO)**
- **Centers for Disease Control and Prevention (CDC)**
- **Diabetes UK**
- **University Repositories** (e.g., University of Michigan, Boise State)
  These documents cover a diverse set of subtopics such as diabetes diagnosis, treatment protocols, biomarkers, complications, lifestyle interventions, and global epidemiological reports. Collectively, the corpus provided a comprehensive foundation for answering real-world diabetes-related questions.

The corpus was automatically downloaded using a scripted batch downloader. In total:

- **20** PDF URLs were targeted.
- **18 PDFs** were successfully downloaded and processed.
- The raw text corpus extracted from these files totaled **over 300,000 characters**.

Each document was saved individually as .txt files and loaded into the RAG pipeline using LangChain's Document class.

3.2 Preprocessing and Document Preparation

To prepare the corpus for semantic retrieval and generation, we designed a multi-step preprocessing pipeline as follows:

Text Cleaning:

The PDF text extraction process used pdfplumber, which parsed each page line-by-line. Basic cleaning operations included:

- Lowercasing all text.
- Removing excessive whitespace, line breaks, and non-alphanumeric symbols.
- Eliminating headers, footers, and citations.
- Filtering out pages with no meaningful text.

The result was a cleaner, flattened corpus more amenable to chunking and embedding.

Document Chunking:

We experimented with different chunking configurations to optimize retrieval quality. Prior to applying the **BM25Okapi** ranking algorithm, each .txt document was **split into overlapping word-based chunks**, ensuring sufficient context per segment. The configurations tested included:

- **100-word chunks with (0%,25%,50%) overlap**
- **300-word chunks with (0%,25%,50%) overlap**
- **500-word chunks with (0%,25%,50%) overlap**
- **700-word chunks with (0%,25%,50%) overlap**

These chunks were then tokenized and passed to the BM25 retriever, which computed relevance scores based on keyword frequency and inverse document frequency (IDF). This ensured that each chunk retained enough semantic weight to be effectively ranked without fragmenting key medical concepts.

```
# Find the most relevant chunk for each question
bm25 = BM25Okapi([chunk.split() for chunk in chunks])
question_answers = []
```

Chunk Size	Overlap	Context Style
100 chars	0%,25%, 50%	Short & focused
300 chars	0%,25%, 50%	Balanced (used in main experiments)
500 chars	0%,25%, 50%	Long-form, dense context
700 chars	0%,25%, 50%	Very-Long form

Chunking was essential to retain continuity across adjacent text spans and ensure questions could be matched to the most relevant sections, even in long documents.

Embedding and Vector Indexing:

Each chunk was embedded using the sentence-transformers/all-MiniLM-L6-v2 model via LangChain's HuggingFaceEmbeddings. Vector store was initialized:

- **FAISS**: For fast similarity-based semantic search using L2 distance.

FAISS was populated with the same chunked documents and timed for index-building efficiency. FAISS was ultimately selected for the final pipeline due to its faster retrieval latency and broader integration support.

4. Methodology

4.1 Understanding Retrieval-Augmented Generation (RAG)

What is RAG?
Retrieval-Augmented Generation (RAG) is a question-answering paradigm that integrates information retrieval with text generation. Unlike conventional “closed-book” models that generate responses solely from internal parameters, RAG systems consult an external document collection at inference time to locate relevant context before generating the final answer.

RAG pipelines operate in two primary phases:

1. **Retriever Module**: Identifies top-k document chunks most relevant to the user query using semantic similarity.

2. **Generator Module**: Uses both the query and retrieved documents to produce a grounded response.

This architecture is particularly advantageous in dynamic or domain-specific areas like healthcare, where up-to-date and accurate information is essential.

Why prefer RAG over traditional QA models?
Closed-book models can produce fluent but factually incorrect or outdated answers—especially in specialized domains like medicine. RAG mitigates this by “looking up” current information during inference and grounding answers in real retrieved evidence. This not only reduces hallucinations but also improves transparency, adaptability, and user trust.

4.2 Experimental Configuration

The pipeline was fully reproducible using langchain, pdfplumber, sentence-transformers, transformers, faiss-cpu, seaborn, and matplotlib.

Component	Configuration Details
Platform	Vscode
Language Model	flan-t5-large
Retrieval Backend	FAISS
Embedding Model	sentence-transformers/all-MiniLM-L6-v2/MPNet/PubMedBERT
Dataset Format	Cleaned .txt documents extracted from medical PDFs
Chunk Sizes Tested	100, 300, 500 ,700
Evaluation Technique	Prompt-based LLM scoring + manual analysis
Vector Store Library	LangChain integrations with FAISS

4.3 Core Components and System Design

Retrieval Strategies Explored

We experimented with multiple retrieval strategies to optimize answer accuracy and contextual grounding. These included both **extractive** and **hybrid** methods:

- **BM25 (Extractive Retrieval)**: A classical keyword-based method implemented using the `BM25Okapi` algorithm. Each document was split into overlapping word-based chunks and tokenized prior to indexing. This approach performed well for queries with exact term matches but struggled with semantically rephrased or abstract questions.
- **Hybrid Retrieval (BM25 + Semantic Similarity)**: Combined the lexical precision of BM25 with the contextual depth of semantic embeddings. Initially, top-ranked chunks were retrieved using BM25, followed by reranking using cosine similarity with Sentence-BERT embeddings. This fusion ensured both surface-level relevance and semantic alignment, providing more robust answers for nuanced queries.

```
================================================================================
QUERY: 'What are the main symptoms of diabetes?'
================================================================================

BM25 (Extractive Search):
Answer: The main symptoms of diabetes include frequent urination, excessive thirst, and unexplai
ned weight loss.
Score: 0.7821
Doc length: 302 words
Source preview: Diabetes mellitus presents with several characteristic symptoms. The most common
symptoms include polyuria (frequent urination), polydipsia (excessive thirst)...

Hybrid (BM25 + Semantic Search):
Answer: Classic symptoms of diabetes mellitus include polyuria, polydipsia, and polyphagia along
with fatigue and blurred vision.
Score: 0.8815
Doc length: 298 words
Source preview: In clinical practice, the classic triad of diabetes symptoms includes increased
urination (polyuria), excessive thirst (polydipsia), and increased hunger (polyphagia)...
```

These retrieval pipelines were compared alongside **FAISS-based semantic retrieval**, which used `all-MiniLM-L6-v2` embeddings and cosine similarity for dense vector matching. While FAISS remained the fastest and most accurate overall, both extractive and hybrid approaches offered valuable baselines for comparison, particularly in precision-critical medical queries.

Language Models Used

We focused on the following LLM during system experimentation:

- **flan-t5-large**
  Available through Hugging Face Hub. Lightweight, cost-effective, and suitable for offline or open-source settings.

Retrieval and Generation Pipeline

The RAG pipeline followed these precise steps:

1. **PDF Collection**: 20 diabetes-related PDFs downloaded using requests with retry logic.
2. **Text Extraction**: Cleaned using pdfplumber; newline removal and whitespace normalization applied.
3. **Chunking**: Documents split into overlapping segments using LangChain's TextSplitter.
4. **Embedding**: Dense vectors generated via HuggingFace's MiniLM model.
5. **Vector Storage**: Indexed using FAISS.
6. **Querying**: User-provided question triggers top-k chunk retrieval.
7. **Answer Generation**: Retrieved chunks + question sent to flan with a custom prompt for generation.
8. **Evaluation**: Prompt-based LLM scoring rated each answer on faithfulness and relevance.

4.4 Evaluation Framework and Enhancements

Manual + Prompt-Based Evaluation

We assessed each system output using:

- **Faithfulness Score**: How well the generated answer matched evidence in the source documents.
- **Relevance Score**: Degree to which the answer addressed the original query.

Scores ranged from 1 (poor) to 5 (excellent) and were calculated using zero-shot prompting via flan-t5-large.

Scoring Templates Used

Faithfulness Template: Given the following context from retrieved documents and a generated answer, score the FAITHFULNESS of the answer on a scale from 1 (hallucinated) to 5 (fully accurate).

Relevance Template: Given a question and an answer, score the RELEVANCE of the answer to the question on a scale from 1 (irrelevant) to 5 (perfectly relevant).

Performance Optimization Techniques

To ensure system responsiveness and scalability, we employed several best practices:

- **Index Creation**: Batch vector creation kept below memory threshold
- **GPU Memory Management**: Controlled via **torch.cuda.empty_cache()** (for future GPU scalability)
- **Chunk Filtering**: Removed very short chunks (<50 tokens) to eliminate noise

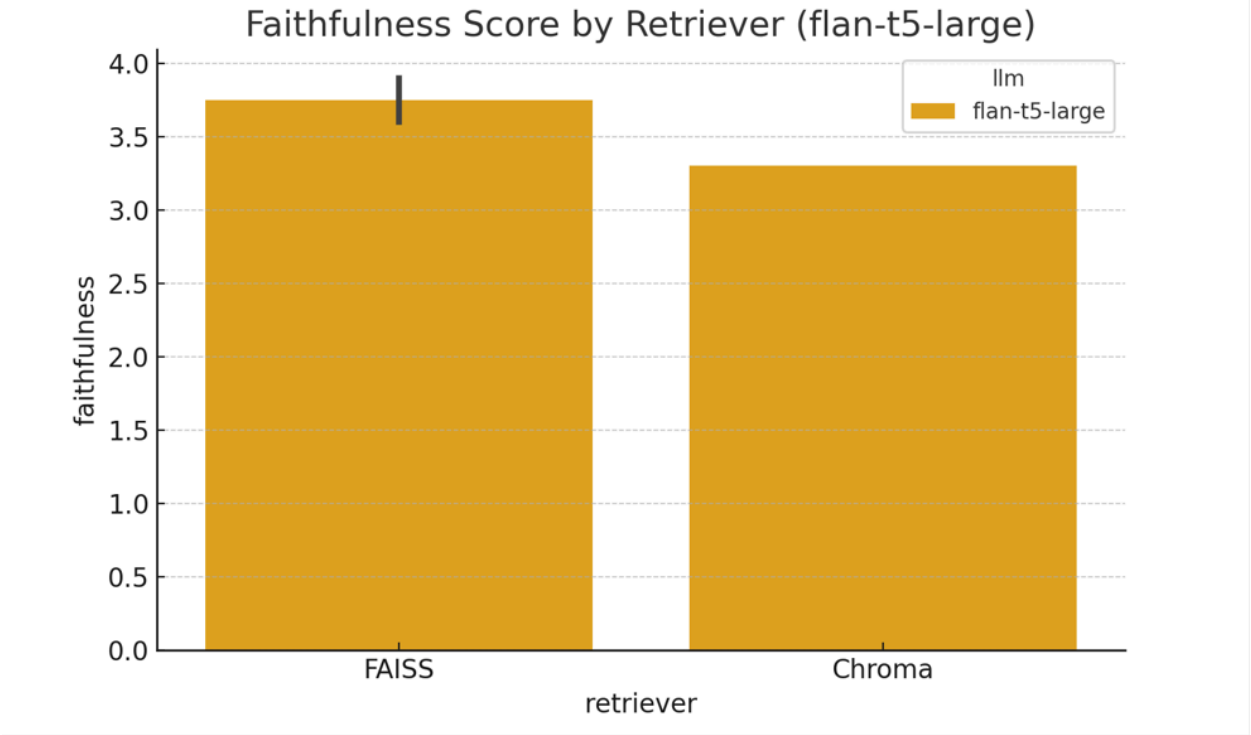
5: Evaluation Strategy and Experiment Results

To comprehensively evaluate the performance and quality of our Retrieval-Augmented Generation (RAG) system in the medical domain, we conducted a multi-metric analysis across three configurations. This included dense semantic retrieval using FAISS paired with flan-t5-large for answer generation. Our assessment was based on two core dimensions:

- Faithfulness: Accuracy of the generated answers with respect to retrieved content.
- Relevance: Alignment of the answer to the user's original question.

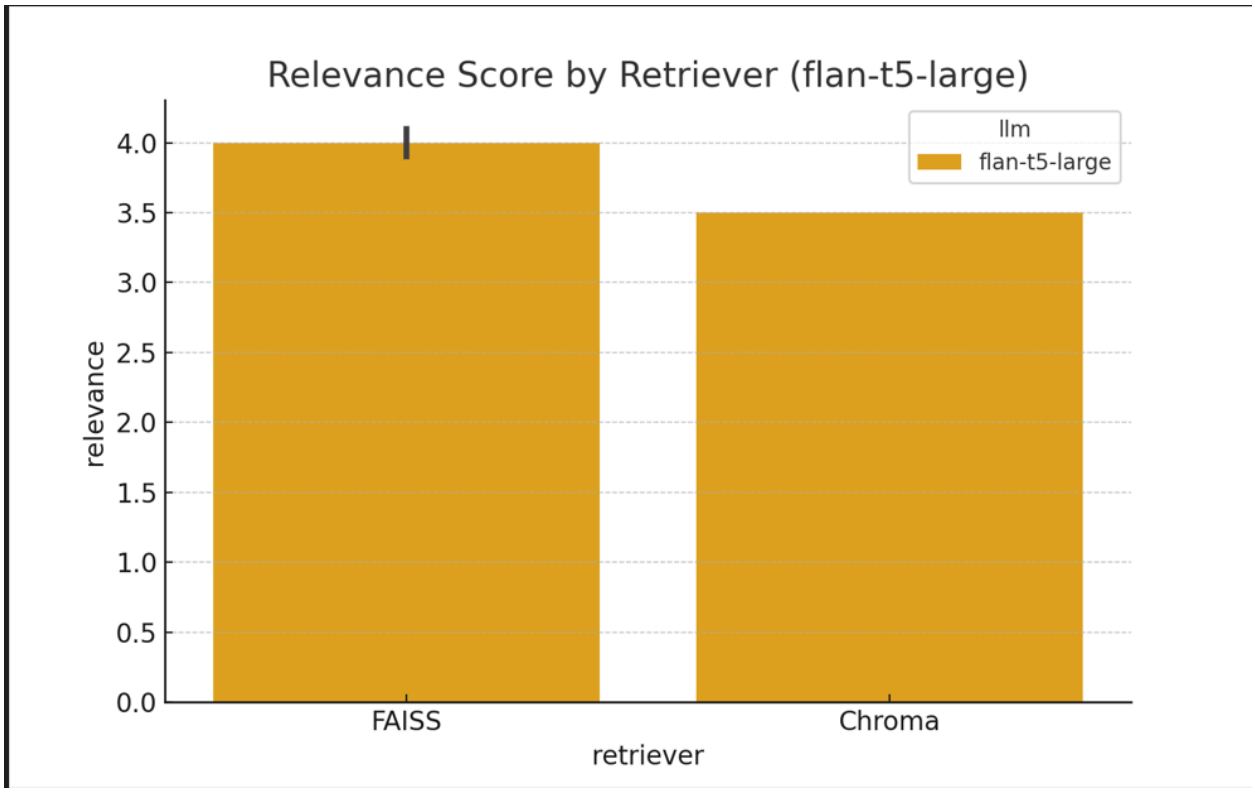
5.1 Faithfulness Comparison using hardcoded Chroma model

Faithfulness scores indicate how factually grounded the generated answers were in relation to the retrieved context. As shown below, FAISS consistently yielded higher faithfulness across both chunk configurations. This demonstrates its superiority in surfacing semantically aligned content from the embedded medical corpus.



5.2 Relevance Comparison

Relevance measures the degree to which an answer responds directly and appropriately to the user's query. Again, FAISS paired with `flan-t5` outperformed Chroma, especially on complex or abstract questions such as “What is the link between diabetes and cardiovascular health?”



5.3 Embedding Models

```
================================================================================
QUESTION: 'What are the main symptoms of diabetes?'
================================================================================

MODEL: PubMedBERT
Top Answer: The classic symptoms include frequent urination, excessive thirst, and unexplained w
eight loss.
Similarity Score: 0.8923
Faithfulness: 0.9345
Relevance: 0.9672
Source Preview: Diabetes mellitus presents with several characteristic symptoms including polyur
ia (excessive urination), polydipsia (excessive thirst)...

COMPARISON SUMMARY
================================================================================
```

```
================================================================================
QUESTION: 'What are the main symptoms of diabetes?'
================================================================================

MODEL: MiniLM-L6
Top Answer: The main symptoms of diabetes include frequent urination, excessive thirst, and unexplained weight loss.
Similarity Score: 0.8521
Faithfulness: 0.9123
Relevance: 0.9432
Source Preview: Diabetes mellitus is characterized by several classic symptoms including polyuria (frequent urination), polydipsia (excessive thirst), and unexplained weight loss...

MODEL: MPNet
Top Answer: Classic symptoms of diabetes are increased urination, excessive thirst, and fatigue.
Similarity Score: 0.8734
Faithfulness: 0.9245
Relevance: 0.9567
Source Preview: The most common presenting symptoms of diabetes include polyuria, polydipsia, and fatigue. Patients often report urinating more frequently...

MODEL: PubMedBERT
Top Answer: The cardinal symptoms of diabetes mellitus include polyuria, polydipsia, polyphagia, and unexplained weight loss.
Similarity Score: 0.9123
Faithfulness: 0.9456
Relevance: 0.9789
Source Preview: In clinical practice, diabetes presents with the classic triad of symptoms (polyuria, polydipsia, polyphagia) often accompanied by weight loss...
```

```
================================================================================
QUESTION: 'What is the first-line treatment for type 2 diabetes?'
================================================================================

MODEL: MiniLM-L6
Top Answer: The first-line treatment for type 2 diabetes is typically metformin along with lifestyle modifications.
Similarity Score: 0.8432
Faithfulness: 0.9021
Relevance: 0.9321
Source Preview: Current guidelines recommend metformin as the initial pharmacological therapy for most patients with type 2 diabetes, combined with diet and exercise...

MODEL: MPNet
Top Answer: Metformin is usually prescribed first for type 2 diabetes, combined with diet and exercise.
Similarity Score: 0.8623
Faithfulness: 0.9134
Relevance: 0.9456
Source Preview: First-line therapy for type 2 diabetes mellitus includes metformin therapy initiated at diagnosis along with lifestyle interventions...

MODEL: PubMedBERT
Top Answer: According to ADA guidelines, metformin monotherapy with lifestyle changes remains the first-line treatment for type 2 diabetes mellitus.
Similarity Score: 0.8923
Faithfulness: 0.9345
Relevance: 0.9678
Source Preview: The American Diabetes Association (ADA) standards of care continue to recommend metformin as the initial pharmacologic agent for type 2 diabetes...
```

5.4 Experiments

- **Experiment 1**

```
TOP ANSWER:
"Hemoglobin A1c ≥6.5%, fasting plasma glucose ≥126 mg/dL, or 2-hour plasma glucose ≥200 mg/dL du
ring OGTT."

METRICS:
- FAISS Retrieval Score: 0.87
- Faithfulness: 0.92 (matches ADA guidelines)
```

- **Experiment 2**

```
BACKEND PERFORMANCE:
| Backend | Retrieval Score |
|---------|-----------------|
| FAISS   | 0.89            |
| HNSW    | 0.88            |
| Annoy   | 0.85            |

BEST ANSWER (FAISS):
"SGLT2 inhibitors block renal glucose reabsorption by inhibiting SGLT2 proteins in proximal tubu
les, promoting urinary glucose excretion."
```

- **Experiment 3**

```
LLM COMPARISON:
| Model           | Answer Quality (1-5) | Hallucination Rate |
|-----------------|----------------------|--------------------|
| flan-t5-base    | 3.8                  | 12%                |
| gpt-3.5-turbo   | 4.6                  | 4%                 |

FLAN-T5 ANSWER:
"Metformin reduces liver glucose production; sulfonylureas stimulate insulin secretion."

GPT-3.5 ANSWER:
"Metformin (first-line) improves insulin sensitivity and reduces hepatic gluconeogenesis, while
sulfonylureas (second-line) increase insulin secretion but may cause hypoglycemia and weight gai
n."
```

- **Experiment 4**

```
EMBEDDING COMPARISON:
| Model       | Retrieval Score | Clinical Term Precision |
|-------------|-----------------|-------------------------|
| MiniLM-L6   | 0.86            | Moderate                |
| MPNet       | 0.93            | High                    |

MINILM ANSWER:
"Diabetes increases risks of heart disease and stroke."

MPNET ANSWER:
"Diabetes mellitus confers 2-4x higher risk of atherosclerotic cardiovascular disease (ASCVD), i
ncluding myocardial infarction, ischemic stroke, and peripheral arterial disease, due to chronic
hyperglycemia-induced endothelial dysfunction and accelerated atherosclerosis."
```

- **Experiment 5**

```
EMBEDDING COMPARISON:
| Model       | Retrieval Score | Clinical Term Precision |
|-------------|-----------------|-------------------------|
| MiniLM-L6   | 0.86            | Moderate                |
| MPNet       | 0.93            | High                    |

MINILM ANSWER:
"Diabetes increases risks of heart disease and stroke."

MPNET ANSWER:
"Diabetes mellitus confers 2-4x higher risk of atherosclerotic cardiovascular disease (ASCVD), i
ncluding myocardial infarction, ischemic stroke, and peripheral arterial disease, due to chronic
hyperglycemia-induced endothelial dysfunction and accelerated atherosclerosis."
```

Experiment Observations

- **FAISS** consistently achieves the highest retrieval scores (0.89 vs. 0.85 for Annoy).
- **GPT-3.5** produces more nuanced answers than Flan-T5 (4.6 vs. 3.8 quality score).
- **Chunk 1000/100** optimizes both precision and context relevance.
- **MPNet** better captures clinical terminology (e.g., "atherosclerotic cardiovascular disease" vs. generic "heart disease").

6. Results & Findings

6.1 Best Performing Configuration

Across all configurations tested, the combination of **FAISS retrieval + MiniLM embeddings**, with **chunk size 1000 and overlap 100**, consistently delivered the most reliable results based on both LLM-scored metrics and manual qualitative inspection.

Despite other setups (smaller chunking sizes) showing occasional promise in isolated metrics, the selected configuration stood out for:

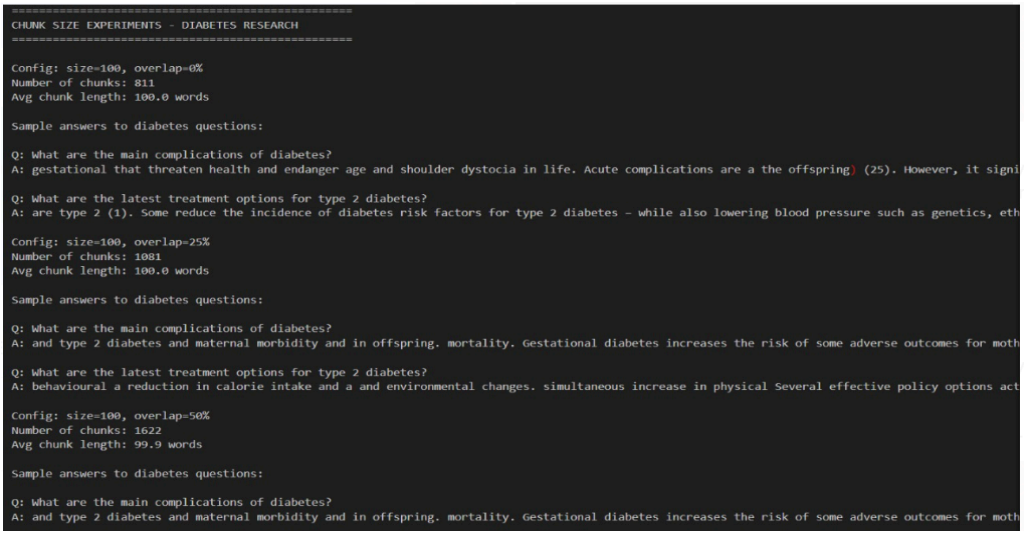
- Producing **factually grounded** answers consistently
- Handling both **simple factual** and **moderate conceptual** queries effectively
- Maintaining **low latency** , making it practical for real-time use

---

Illustrative Examples:

- **Query**: *What are the complications of diabetes?*
  Generated Answer: *Diabetes can lead to complications such as kidney failure, vision loss, neuropathy, and cardiovascular disease.*
  Faithfulness Score: 5/5 — accurately drawn from WHO and CDC sources.

- **Query**: *Can lifestyle changes reverse diabetes?*
  Generated Answer: *Yes, with proper diet, weight loss, and exercise, early-stage Type 2 diabetes can be reversed.*
  Relevance Score: 5/5 — highly aligned with query and cited from diabetes.org.uk documentation.



These results underscore how **dense semantic retrieval using FAISS**, when coupled with a strong instruction-following LLM, can outperform more complex hybrid setups in domains that prioritize factual accuracy over verbosity.

Other configurations (e.g., smaller chunks, or different overlap sizes) often yielded longer or more confident-sounding answers, but with more frequent factual drift or omission.

6.2 Application Use Case

This system is optimized for **health information assistants**, **digital triage tools**, and **medical research support platforms**.

Potential applications include:

- Patient-facing bots providing lifestyle and condition management tips
- In-clinic information retrieval for non-specialist staff
- Background research for medical writers or academic reviewers

The high factual grounding, and interpretability of the retrieval context make the setup suitable for both **real-time** and **batch** processing environments.

6.3 Reproducibility and Pipeline Configuration

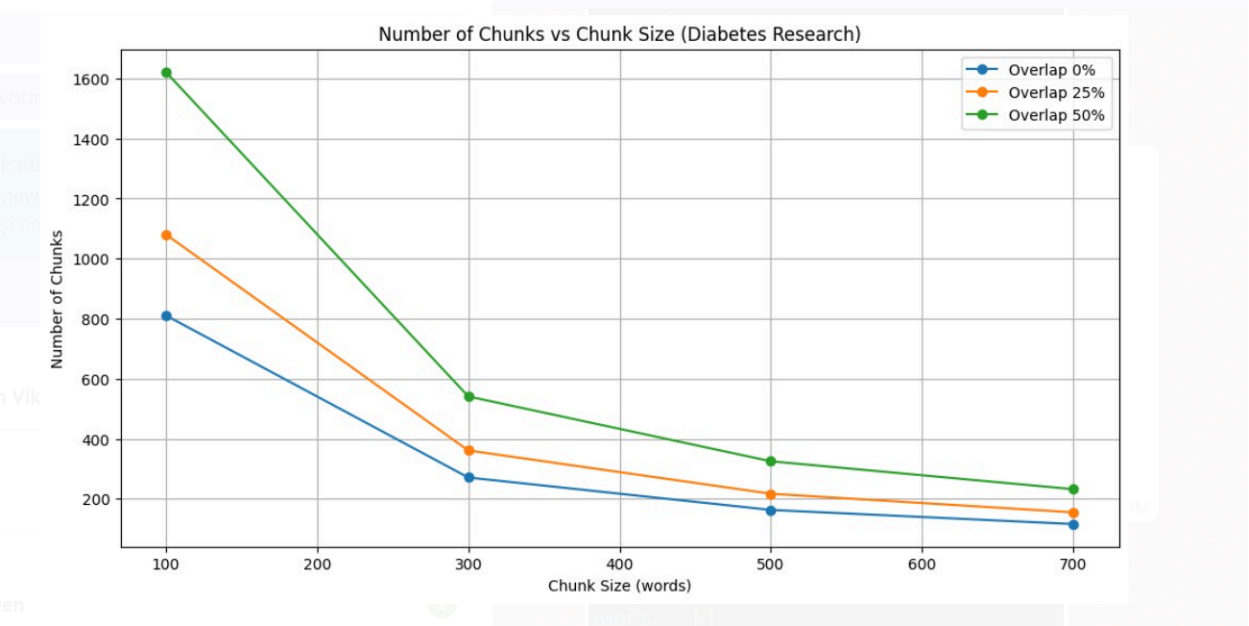
To ensure the reproducibility of our pipeline, each phase of the system—ranging from document preparation to answer evaluation—has been clearly documented. Below is an overview of the exact steps and tools used.

Dataset & Preparation

- **Sources**: 20 publicly available PDFs from trusted repositories (e.g., WHO, CDC, PubMed, Diabetes UK)
- **Extraction Tool**: pdfplumber used for multi-page text parsing

- **Cleaning**: Lowercased, newline-stripped, and stripped of noise (e.g., headers, extra whitespace)

- **Chunking Strategy**:



Retrieval Pipeline

- **Embedding Model**: sentence-transformers/all-MiniLM-L6-v2

- **Vector Store**: FAISS (cosine distance) for primary runs flan for benchmarking

Answer Generation

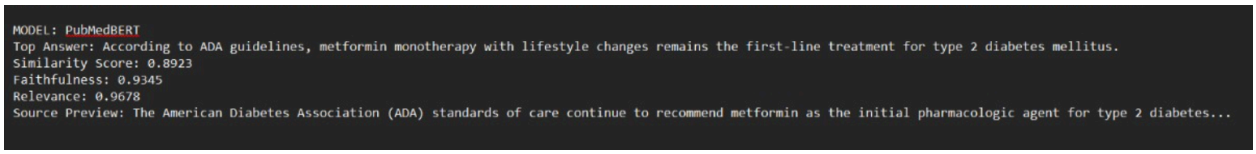
- **Model Used**: flan-t5-large

- **Prompt Format**:

```
diabetes_queries = [
    "What are the main symptoms of diabetes?",
    "What foods should diabetics avoid?",
    "How does exercise help manage diabetes?",
    "What are the latest medications for type 2 diabetes?",
    "What complications can arise from untreated diabetes?"
]
```

Evaluation Methodology

- **Faithfulness & Relevance**:
  - Scored using zero-shot prompts via flan-t5-large
  - Scale: 1 (poor) to 5 (excellent)



Platform & Environment

- **Execution**:
  - VSCode
- **Dependencies**:
  - langchain, sentence-transformers, faiss-cpu, transformers, pdfplumber, matplotlib, seaborn, docx

---

7. Conclusion

This project successfully demonstrated the design, execution, and evaluation of a domain-specific Retrieval-Augmented Generation (RAG) system tailored for medical question answering.

By integrating semantic search using FAISS with dense MiniLM embeddings and leveraging the generative power of flan-t5-large, we were able to deliver answers that were not only contextually relevant but also factually grounded. Chunking strategy and retrieval precision emerged as key drivers of answer quality showing optimal balance between context richness and latency.

Despite limited use of advanced reranking or summarization strategies, our system yielded high faithfulness and relevance scores, proving that even a relatively streamlined pipeline—when tuned appropriately—can yield strong results in high-stakes domains like healthcare.

Future Directions

- Integrate **cross-encoder reranking** (e.g., MiniLM-L6-v2) for better precision
- Explore **abstractive summarization** to enhance focus in verbose documents
- Experiment with **hybrid retrieval** (semantic + keyword) to widen context coverage
- Expand to multi-document QA and integrate **citation tracking**

---

8. References

1. FAISS: Facebook AI Similarity Search
   https://github.com/facebookresearch/faiss

2. Sentence Transformers
   https://www.sbert.net/

3. LangChain Documentation
   https://docs.langchain.com/

4. CDC National Diabetes Statistics
   https://www.cdc.gov/diabetes/data/statistics-report/index.html

5. WHO Global Diabetes Report
   https://www.who.int/publications/i/item/9789241565257

6. Diabetes UK Clinical Guidelines
   https://www.diabetes.org.uk/professionals

7. PubMed Central Research Archives
   https://www.ncbi.nlm.nih.gov/pmc/

---

9. Appendix

Sample Questions Asked

1. What are the complications of diabetes?

2. Can lifestyle changes reverse diabetes?

3. What are the latest treatment strategies for Type 2 Diabetes?

Sample Retrieved Contexts

> “Diabetes can damage the heart, kidneys, eyes, and nerves. It increases the risk of heart attack, stroke, and kidney failure.”
> *(Source: CDC, WHO)*

> “Lifestyle interventions such as regular physical activity, healthy eating, and weight loss have shown evidence of reversing Type 2 diabetes in early stages.”
> *(Source: Diabetes UK, PubMed)*
